ISSN 2005-3053

2017년도

제29회 한글 및 한국어 정보처리 학술대회

한글 및 한국어 정보처리

일시: 2017년 10월 13일(금) ~ 14일(토)

장소: 대구가톨릭대학교 취창업관

경북대학교 공대12호관

■ 주최: 한국정보과학회, 한국인지과학회

■ 주관: 한국정보과학회 언어공학연구회

■ 후원: ㈜SK텔레콤, ㈜엔씨소프트, KISTI, 대신정보통신(주), 네이버, 삼성 한국전자통신연구원, ㈜다음소프트, 아이티센, 카카오, KT, 솔트룩스

2017년도
제29회 한글 및 한국어 정보처리 학술대회

➽위원회

- **대회장** : 김영길 (ETRI)
- **조직위원장** : 김병창(대구가톨릭대학교), 박성배(경북대학교)
- **조직위원** :

강승식(국민대)	강현규(건국대)	권혁철(부산대)	김경선(다이퀘스트)
김덕봉(성공회대)	김영섬(Konantech)	김재훈(한국해양대)	김판구(조선대)
김학수(강원대)	김현기(ETRI)	나동열(연세대)	맹성현(KAIST)
박상규(ETRI)	박세영(경북대)	박종철(KAIST)	박혁로(전남대)
서정연(서강대)	심광섭(성신여대)	안동언(전북대)	옥철영(울산대)
유홍진(ITCEN)	윤준태(다음소프트)	이경일(Saltlux)	이근배(POSTECH)
이재성(충북대)	임해창(고려대)	임희석(고려대)	장두성(KT)
장명길(ETRI)	최기선(KAIST)	최재웅(고려대)	황도삼(영남대)

- **학술위원장** : 이공주(충남대학교), 고언숙(조선대학교)
- **학술위원** :

강상우(가천대)	강신재(대구대)	고영중(동아대)	김상범(NHN)
김성동(한성대)	김유섭(한림대)	나승훈(전북대)	류법모(부산외대)
민혜진(NHN)	박소영(상명대)	선충녕(KISTI)	양단희(평택대)
여상화(경인여대)	오효정(전북대)	윤성희(상명대)	이경순(전북대)
이도길(고려대)	이상곤(전주대)	이성욱(한국교통대)	이승우(KISTI)
이창기(강원대)	이충희(ETRI)	이현아(금오공대)	이호준(유원대)
임수종(ETRI)	임준호(ETRI)	전문기(건국대)	정민화(서울대)
정한민(KISTI)	차정원(창원대)	최호섭(한국이디에스)	한경수(성결대)
황영숙(SK플래닛)			

- **국어 정보 처리 시스템 경진대회** : 김유섭(한림대학교)

➽ 후원기관:

㈜SK텔레콤, ㈜엔씨소프트, KISTI, 대신정보통신(주), 네이버, 삼성 한국전자통신연구원, ㈜다음소프트, 아이티센, 카카오, KT, 솔트룩스

Platinum

Gold

Silver

Bronze

➽ 초대의 말씀

한글 및 한국어 정보처리 학술대회는 1989년 10월에 제1회 학술대회를 시작으로 매년 한글날 전후에 개최하고 있는데 지난 29년 동안 계속되고 있으며 올해로 29번째 학술대회를 개최하게 되었습니다. 언어처리 기술을 주요 내용으로 하고 있는 우리 학술대회는 자연언어처리 기술을 기반으로 기계번역과 정보검색, 말뭉치 구축, 시맨틱웹, 온톨로지, 대화체 질의응답 시스템, 텍스트 마이닝, 빅 데이터 분석, SNS 분석 등 다양한 분야로 융합되어 발전되고 있습니다.

빅데이터가 화두인 요즘 방대한 텍스트로부터 의미 있는 정보를 끄집어내기 위한 언어처리 연구가 전 세계적으로 활발히 진행되고 있습니다. 지난 29년간 산학관연을 아우르는 언어처리 학술교류의 장으로써 맡은바 소임을 다해오고 있는 "한글 및 한국어 정보처리 학술대회"에 오셔서 그간의 연구 결과를 발표하고, 학문 발전을 위한 활발한 토론을 하며 자리를 빛내 주시길 희망합니다.

이번 학술대회 논문은 소정의 심사 과정을 통하여 구두 발표와 포스터 발표가 준비되어 있습니다. 이를 위하여 바쁘신 와중에서도 논문 모집에서부터 심사에 이르기까지 많은 수고를 해주신 조직위원 및 학술위원님들께 깊이 감사드립니다.

이번에도 언어처리 기술과 관련된 많은 학자들이 모여서 연구 및 학술의 교류와 활발한 토론이 이루어지는 학술대회가 되기를 바랍니다. 마지막으로 이번 행사를 위하여 후원해 주신 여러 후원 기관들에게 진심으로 감사의 말씀을 전합니다. 또한, 학술대회를 준비하고 진행을 맡아주신 조직위원 및 행사 진행 요원들께도 이 자리를 빌어서 고마움을 전하고 싶습니다.

2017년 9월 22일

제29회 한글 및 한국어 정보처리 학술대회 조직위원장 김병창(대구가톨릭대학교), 박성배(경북대학교)
학술위원장 이공주(충남대학교)
국어 정보 처리 시스템 경진대회 운영위원장 김유섭(한림대학교)
한국정보과학회 언어공학연구회 위원장 김영길(한국전자통신연구원)
한국인지과학회 회장 장병탁(서울대학교)
한국인지과학회 학술위원장 고언숙(조선대학교)

프로그램 전체 일정표

■ 10월 13일(금요일)

장소: 대구가톨릭대학교 취창업관, 성바오로문화관 강당

<table>
<tr><th>시 간</th><th>일 정</th><th>시간</th><th>일 정</th></tr>
<tr><td></td><td>신진연구자 발표 및 초청강연
(취창업관 208호)</td><td></td><td>국어정보처리 경진대회
(성바오로문화관 강당)</td></tr>
<tr><td>12:00~13:00</td><td>등록</td><td>13:00~13:30</td><td>등록</td></tr>
<tr><td>13:00~13:10</td><td>개회식</td><td rowspan="2">13:30~13:40</td><td rowspan="2">개회식</td></tr>
<tr><td rowspan="2">13:10~14:10</td><td rowspan="2">신진연구자 발표
(송현제 박사, 김세종 박사)</td></tr>
<tr><td rowspan="4">13:40~15:20</td><td rowspan="4">발표 1부
(지정 분야:
개체명 인식 시스템)</td></tr>
<tr><td>14:10~14:30</td><td>휴식</td></tr>
<tr><td>14:30~15:00</td><td>초청강연-1
The Paradigm Shift from Retrieval to Question Answering
(네이버 강인호 박사)</td></tr>
<tr><td rowspan="2">15:00~15:30</td><td rowspan="2">초청강연-2
딥러닝 기반 자연어 처리: 요약과 챗봇
(경북대 이민호 교수)</td></tr>
<tr><td rowspan="2">15:20~16:00</td><td rowspan="2">10개 팀 시스템 시연 및
휴식</td></tr>
<tr><td>15:30~16:00</td><td>초청강연-3
빅데이터 분석의 미래 - 로봇 리포트 작성 (다음소프트 윤준태 박사)</td></tr>
<tr><td>16:00~16:30</td><td>초청강연-4
우리나라에서 기계번역의 태동
(김길창 교수)</td><td>16:00~16:25</td><td>발표 2부
(일반 분야)</td></tr>
<tr><td>16:30~17:30</td><td>기업체 세션</td><td>16:25~17:30</td><td>심사 회의</td></tr>
<tr><td>17:30~18:00</td><td colspan="3">경과보고, 공로패 및 감사패 증정, 경진대회 시상식</td></tr>
<tr><td>18:00~19:20</td><td colspan="3">저녁식사</td></tr>
<tr><td>18:00~18:30</td><td colspan="3">운영위원회</td></tr>
</table>

■ 10월 14일(토요일)

장소: 경북대학교 공대 12호관

시 간	일 정			
	학술정보실2 109호	세미나실4 111호	세미나실3 113호	세미나실 114호
09:00 ~ 10:20	구두발표 1: 언어자원 좌장: 김한샘 교수 (연세대)	구두발표 2: 대화/질의응답 1 좌장: 온병원 교수 (군산대)	구두발표 3: 정보검색 좌장: 나승훈 교수 (전북대)	구두발표 4: 언어처리 활용 좌장: 김학수 교수 (강원대)
10:20 ~ 11:30	포스터 발표(공대 12호관 로비) 좌장: 이공주 교수(충남대)			차세대정보컴퓨팅 사업 소개 (10:40~11:30)
11:30 ~ 12:50	구두발표 5: 형태소/구문분석 좌장: 임수종 박사 (ETRI)	구두발표 6: 대화/질의응답 2 좌장: 이창기 교수 (강원대)	구두발표 7: 정보추출 좌장: 이현아 교수 (금오공대)	구두발표 8: 의미분석 좌장: 류법모 교수 (부산외대)
12:50	폐회			

프로그램

10월 13일(금)

◉ 신진연구자 발표 및 초청강연 (대구가톨릭대학교 취창업관 208호)

13:00 ~ 16:30

13:00 ~ 13:10 개회식

13:10 ~ 13:40 분류 문제의 Covariate shift 해소와 Encoder-Decoder 기반의 한국어 자연어처리
(송현제 박사, 네이버)

13:40 ~ 14:10 서브토픽 마이닝 소개 및 활용 (김세종 박사, 네이버)

14:10 ~ 14:30 휴식

14:30 ~ 15:00 강인호 박사 (네이버)
The Paradigm Shift from Retrieval to Question Answering

15:00 ~ 15:30 이민호 교수 (경북대)
딥러닝 기반 자연어 처리: 요약과 챗봇 (Nature language processing based on deep learning: Abstractive summarization and Chatbot)

15:30 ~ 16:00 윤준태 박사 (다음소프트)
빅데이터 분석의 미래 - 로봇 리포트 작성

16:00 ~ 16:30 김길창 교수
우리나라에서 기계번역의 태동 (Beginning of machine translation in Korea)

◉ 기업체 세션 (대구가톨릭대학교 취창업관 208호)

16:30 ~ 17:30

16:30 ~ 17:30 기업체 세션

◉ 시상식 및 폐회식 (대구가톨릭대학교 취창업관 208호)

17:30 ~ 18:30

17:30 ~ 18:30 HCLT 2017 경과 보고

언어공학회 공로패 및 감사패 증정

경진대회 시상식(사회: 김유섭(한림대))

10월 14일(토)

◉ 구두발표 1: 언어자원 (경북대학교 공대12호관 109호 학술정보실2)

09:00 ~ 10:20 좌장: 김한샘 교수 (연세대)

09:00 ~ 09:20 지식베이스 확장을 위한 행렬 분해 모델 / 김지호, 남상하, 최기선 (KAIST)

09:20 ~ 09:40 딥러닝을 이용한 대규모 한글 폰트 인식 / 양진혁, 곽효빈, 김인중 (한동대)

09:40 ~ 10:00 한국어에서 Attention 모델과 Naïve Bayes 모델 기반의 어휘 말뭉치 구축 및 응용에 관한 연구 / 윤주성, 김현철 (고려대)

10:00 ~ 10:20 한국어 대화문 화행 자동분류를 위한 언어학적 기반연구 / 구영은, 김지연, 홍문표, (성균관대), 김영길(ETRI)

◉ 구두발표 2: 대화/질의응답 1 (경북대학교 공대12호관 111호 세미나실4)

09:00 ~ 10:20 좌장: 온병원 교수 (군산대)

09:00 ~ 09:20 복사 방법 및 검색 방법을 이용한 종단형 생성 기반 질의응답 채팅 시스템 / 김시형, 김학수 (강원대), 권오욱, 김영길(ETRI)

09:20 ~ 09:40 한국어 대화 모델 학습을 위한 디노이징 응답 생성 / 김태형, 노윤석, 박성배, 박세영 (경북대)

09:40 ~ 10:00 S2-Net: SRU 기반 Self-matching Network를 이용한 한국어 기계 독해 / 박천음, 이창기(강원대), 홍수린, 황이규, 유태준 (마인즈랩), 김현기(ETRI)

10:00 ~ 10:20 Dual Bi-Directional Attention Flow를 이용한 한국어 기계이해 시스템 / 이현구, 김학수(강원대), 최정규, 김이른(LG전자)

◉ 구두발표 3: 정보검색 (경북대학교 공대12호관 113호 세미나실3)

09:00 ~ 10:20 좌장: 나승훈 교수 (강원대)

09:00 ~ 09:20 TextRank 알고리즘과 주의 집중 순환 신경망을 이용한 하이브리드 문서 요약 / 정석원, 이현구, 김학수 (강원대)

09:20 ~ 09:40 대규모 분류 체계에서 계층적 샘플링을 활용한 문서의 분류 / 홍성모, 장헌석, 강인호 (네이버)

09:40 ~ 10:00 CNN을 이용한 발화 주제 다중 분류 / 최경호, 김경덕, 김용희, 강인호 (네이버)

10:00 ~ 10:20 단어의 위치정보를 이용한 Word Embedding / 황현선, 이창기(강원대), 장현기, 강동호 (SK C&C)

◉ 구두발표 4: 언어처리 활용 (경북대학교 공대12호관 114호 세미나실)

09:00 ~ 10:20 좌장: 김학수 교수 (강원대)

09:00 ~ 09:20 Sequence-to-sequence 모델을 이용한 로마자-한글 상호(商號) 표기 변환 시스템 / 김태현, 정현근, 김재화, 김정길 (사람인HR)

09:20 ~ 09:40 심층학습을 이용한 음절태깅 기반의 외래어 인식 시스템 / 박호민(한국해양대), 김창현(ETRI), 천민아, 노경목, 김재훈(한국해양대)

09:40 ~ 10:00 Distance LSTM-CNN with Layer Normalization을 이용한 음차 표기 대역 쌍 판별 / 이창수, 천주룡, 김주근, 김태일, 강인호 (네이버)

10:00 ~ 10:20 LSTM을 이용한 한국어 이미지 캡션 생성 / 박성재, 차정원 (창원대)

◉ 구두발표 5: 형태소/구문분석 (경북대학교 공대12호관 109호 학술정보실2)

11:30 ~ 12:50 좌장: 임수종 박사 (ETRI)

11:30 ~ 11:50 동적 오라클을 이용한 한국어 의존 구문분석 / 이경호, 이공주 (충남대)

11:50 ~ 12:10 멀티 레이어 포인터 네트워크를 이용한 한국어 의존 구문 분석 / 박천음, 황현선, 이창기 (강원대), 김현기(ETRI)

12:10 ~ 12:30 딥러닝을 이용한 전이 기반 한국어 품사 태깅 & 의존 파싱 통합 모델 / 민진우, 나승훈 (전북대), 신종훈 (ETRI)

12:30 ~ 12:50 Multi-task sequence-to-sequence learning을 이용한 한국어 형태소 분석과 구구조 구문 분석 / 황현선, 이창기 (강원대)

◉ 구두발표 6: 대화/질의응답 2 (경북대학교 공대12호관 111호 세미나실4)

11:30 ~ 12:50 좌장: 이창기 교수 (강원대)

11:30 ~ 11:50 도메인 특정 지식을 결합한 End-to-End Learning 방식의 한국어 식당 예약 대화 시스템 모델 개발 / 이동엽, 김경민, 임희석 (고려대)

11:50 ~ 12:10 심층적 의미 매칭을 이용한 cQA 시스템 질문 검색 / 김선훈, 장헌석, 강인호 (네이버)

12:10 ~ 12:30 CNN-LSTM 신경망을 이용한 발화 분석 모델 / 김민경, 김학수 (강원대)

12:30 ~ 12:50 색인어 인코딩과 음절 디코딩에 기반한 생성 채팅 모델 / 김진태, 김시형, 김학수 (강원대), 이연수, 최맹식(엔씨소프트)

◉ 구두발표 7: 정보추출 (경북대학교 공대12호관 113호 세미나실3)

11:30 ~ 12:50 좌장: 이현아 교수 (금오공대)

11:30 ~ 11:50 Bidirectional LSTM-CRF 앙상블을 이용한 공간 개체 추출 / 민태홍, 이재성(충북대)

11:50 ~ 12:10 다중-어의 단어 임베딩을 적용한 CNN 기반 원격 지도 학습 관계 추출 모델 / 남상하, 한기종, 김은경, 권성구, 정유성, 최기선 (KAIST)

12:10 ~ 12:30 제한된 언어 자원 환경에서의 다국어 개체명 인식 / 천민아(한국해양대), 김창현(ETRI), 박호민, 노경묵, 김재훈 (한국해양대)

12:30 ~ 12:50 한국어 특질을 고려한 단어 벡터의 Bi-LSTM 기반 개체명 모델 적용 / 남석현, 함영균, 최기선 (KAIST)

◉ 구두발표 8: 의미분석 (경북대학교 공대12호관 114호 세미나실)

11:30 ~ 12:50 좌장: 류법모 교수 (부산외대)

11:30 ~ 11:50 코어넷을 활용한 비지도 한국어 어의 중의성 해소 / 한기종, 남상하, 김지성, 함영균, 최기선 (KAIST)

11:50 ~ 12:10 Highway BiLSTM-CRFs 모델을 이용한 한국어 의미역 결정 / 배장성, 이창기 (강원대), 김현기(ETRI)

12:10 ~ 12:30 Layer Normalized LSTM CRF를 이용한 한국어 의미역 결정 / 박광현, 나승훈 (전북대)

12:30 ~ 12:50 SC-GRU encoder-decoder 모델을 이용한 자연어생성 / 김건영, 이창기 (강원대)

◉ 포스터발표 (경북대학교 공대12호관 로비)

10:20~11:30 좌장: 이공주 교수(충남대)

P01 구텐베르그 프로젝트 텍스트 데이터를 활용한 시각화 및 용례 검색 / 김동성, 신연수, 이지안, 유지민 (이화여대)

P02 형식형태소가 한국어 단어 벡터 생성에 미치는 영향 / 윤준영, 김도원, 민태홍, 이재성 (충북대)

P03 한-베 기계번역에서 한국어 분석기 (UTagger)의 영향 / 원광복, 옥철영 (울산대)

P04 단어 임베딩과 음성적 유사도를 이용한 트위터 '서치 방지 단어'의 자동 예측 / 이상아 (서울대)

P05 문서 임베딩을 이용한 소셜 미디어 문장의 개체 연결 / 박영민(서강대), 정소윤(LG전자), 이정엄, 신동수, 김선아(현대자동차), 서정연(서강대)

경진대회

10월 13일(금)

◉ 경진대회 논문 (대구가톨릭대학교 성바오로문화관 강당)

13:40~15:20

지정분야	Bidirectional Dynamic LSTM 을 이용한 음절 단위 개체명 추출 및 자동화된 말뭉치 구축 / 오성식, 임창대, 안기호, 박외진 (㈜아크릴)
지정분야	Bidirectional LSTM CRFs를 이용한 한국어 개체명 인식기 / 강상우, 송치윤, 양성민 (가천대)
지정분야	KACTEIL-NER: 딥러닝과 앙상블 기법을 이용한 개체명 인식기 / 박건우, 박성식, 장영진, 최기현, 김학수 (강원대)
지정분야	Bi-directional LSTM-CNN-CRF를 이용한 한국어 개체명 인식 시스템 / 이동엽, 임희석 (고려대)
지정분야	음절 기반의 CNN를 이용한 개체명 인식 / 박혜웅(아이리마인즈), 송영숙(경희대)
지정분야	언어 모델 다중 학습을 이용한 한국어 개체명 인식 / 김병재, 박찬민, 최윤영, 권명준, 서정연 (서강대)
지정분야	상대적 가중치 자질을 반영한 CRF 기반의 개체명 인식 / 정진욱 (봄랩스)

➽ 목차

● 구두발표 1: 언어자원 (경북대학교 공대12호관 109호 학술정보실2)

● 구두발표 2: 대화/질의응답 1 (경북대학교 공대12호관 111호 세미나실4)

● 구두발표 3: 정보검색 (경북대학교 공대12호관 113호 세미나실3)

● 구두발표 4: 언어처리 활용 (경북대학교 공대12호관 114호 세미나실)

● 구두발표 5: 형태소/구문분석 (경북대학교 공대12호관 109호 학술정보실2)

● 구두발표 6: 대화/질의응답 2 (경북대학교 공대12호관 111호 세미나실4)

● 구두발표 7: 정보추출 (경북대학교 공대12호관 113호 세미나실3)

● 구두발표 8: 의미분석 (경북대학교 공대12호관 114호 세미나실)

● 포스터발표 (경북대학교 공대12호관 로비)

● 경진대회 (대구가톨릭대학교 성바오로문화관 강당)

● 구두발표 1: 언어자원

지식베이스 확장을 위한 행렬 분해 모델

김지호○, 남상하, 최기선

한국과학기술원
{hogajiho, nam.sangha, kschoi}@kaist.ac.kr

Matrix Factorization Models for Knowledge Base Population

Jiho Kim○, Sangha Nam, Key-Sun Choi
KAIST

요 약

지식베이스의 목표는 세상의 모든 지식을 데이터베이스화 하는 것이지만 지식 획득 능력의 부족으로 항상 지식 부족 문제에 시달린다. 지식 획득은 주로 웹 상에 있는 자연언어문장을 지식화 하는 외부적인 지식 획득을 통해 이루어지지만, 지식베이스 내부에서 지식을 확장해 나가는 방법에 대해서는 연구가 소홀히 이루어지고 있다. 따라서 본 논문에서는 내부적인 지식 획득을 위한 지식베이스 행렬 분해 모델을 소개한다. 본 논문에서 소개하는 방법은 지식베이스를 행렬로 변환한 뒤 행렬 분해 모델을 통해 새로운 지식에 대한 신뢰도를 점수화하는 방법이다. 본 논문에서 소개한 방법의 우수성과 실효성을 입증하기 위해 한국어 지식베이스인 한국어 디비피디아(2016-10)를 대상으로 본 모델의 정확도 측정 실험 결과를 소개한다.

주제어: 지식베이스, 지식베이스 확장, 행렬 분해, 지식 획득

1. 서론

웹과 인터넷의 등장과 함께 세상에는 방대한 양의 정보가 공유되기 시작했다. 방대한 정보를 효율적으로 관리하고 사용하려면 기계가 해석할 수 있도록 정보를 정제하는 과정이 필요하다. 이를 위해 정보를 구조화하고 데이터베이스화 시킨 지식베이스(knowledge base)에 대한 연구가 활발히 진행되고 있다. 지식베이스는 정보를 트리플(triple) 구조로 저장한다. 트리플이란 두 개체(entity)간의 관계(relation) 정보를 <주어 개체 - 관계 - 목적어 개체>와 같이 표현하는 구조이다. 예를 들면 "대한민국의 수도는 서울이다." 라는 정보를 트리플 형태로 표현하면 <대한민국 - 수도 - 서울>과 같이 표현할 수 있다.

지식베이스가 유용하게 쓰이기 위한 조건은 크게 두 가지가 있다. 첫 번째는 포함하고 있는 정보의 범위가 넓어야 하는 것이고 두 번째는 포함하고 있는 정보의 정확성이 높아야 하는 것이다. 즉 지식베이스의 유용성을 높이기 위해서는 지속적으로 지식베이스에 새롭고 정확한 지식을 추가하는 지식베이스 확장(knowledge base population)이 필요하다. 지식베이스 확장을 위해서는 새롭게 추가할 트리플을 습득해야 하고, 이에 선행하여 새로운 지식을 알아내는 지식 획득(knowledge acquisition)이 이루어져야 한다.

지식 획득은 새로운 지식의 출처에 따라 지식베이스 내부에서 새로운 지식을 획득하는 내부적인 지식 획득(interior knowledge acquisition)과 지식베이스 외부에서 새로운 지식을 획득하는 외부적인 지식 획득(exterior knowledge acquisition)로 나눌 수 있다. 지식 획득에 관한 연구는 주로 웹에 존재하는 자연언어문장을 읽고 지식을 추출하는 관계추출(relation extraction)에 집중되어 있다. 관계추출에 관한 연구에는 TAC Knowledge Base Population[1][2][3], OpenIE[4], OLLIE[5] 등이 있다. 이들은 영어 도메인에서 자연언어문장을 읽어들여 개체명 간의 관계를 나타내는 트리플을 추출하는 문제에 집중하였다.

지식베이스 확장의 효율성을 높이기 위해서는 내부적인 지식 획득이 동반되어야 한다. 내부적인 지식 획득에 관한 연구로는 귀납적 논리 프로그래밍(inductive logic programming)에 기반한 AMIE[10] 등이 있다. 이들은 지식베이스로부터 논리적 법칙(logical rules)들을 추출하여 새로운 지식을 알아내는 문제에 집중하였다. 하지만 아직 내부적인 지식 획득에 있어 행렬 분해 모델을 이용한 연구는 없다. 따라서 본 논문에서는 내부적인 지식 획득 방법인 KBMF(Knowledge Base Matrix Factorization)을 소개한다. 본 방법은 지식베이스에 저장되어 있는 정보만을 이용하여 새로운 지식에 대한 신뢰도를 점수화하는 방법이다. KBMF를 통해 지식베이스에 새로운 트리플을 추가할 수 있으며, 추가하려는 트리플의 신뢰성까지 측정이 가능하다. 또한 본 방법은 지식베이스의 행렬화와 행렬 분해(matrix factorization) 모델을 분리해서 설계하였기 때문에 추후 관계추출 결과와 지식베이스를 통합하여 분석하는 방향으로 시스템을 확장해 나갈 수 있다.

본 논문은 내부적인 지식 획득에 있어서 본 논문에서 소개한 KBMF가 효과적임을 보이는 것에 중점을 둔다. 한국어 디비피디아(Korean DBpedia)[6]에 본 논문의 행렬 분해 방법을 적용하여 실험해 봄으로써 실효성을 증명하고자 한다.

2. 지식베이스 행렬 분해 모델(KBMF)

KBMF는 행렬화 단계와 행렬 분해 단계, 그리고 상위 트리플 추출 단계로 구성된다. 모델의 흐름도는 그림 1과 같다. KBMF는 지식베이스에 저장된 트리플들을 행렬화하고, 행렬 분해 모델을 통해 개체쌍(entity tuple)과

관계 각각에 대한 특징 벡터를 학습한 뒤 새로운 트리플들이 참일 확률을 계산하여 지식베이스 행렬을 최적화한다. 최적화된 지식베이스 행렬 로부터 학습 데이터를 제외한 나머지 중 높은 확률을 가진 트리플들을 추출하여 새로운 지식을 추출하게 된다.

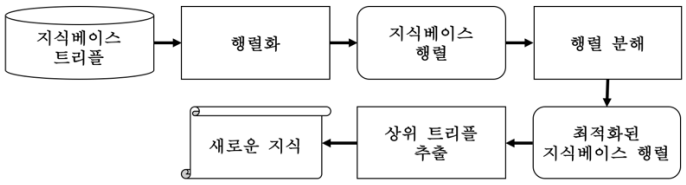


그림 1 KBMF 흐름도

2.1. 지식베이스 행렬화

지식베이스에는 <주어 개체 - 관계 - 목적어 개체>로 구성된 트리플 형태로 지식이 저장되어 있다. 지식베이스 행렬화란 <주어 개체, 목적어 개체> 로 이루어진 개체쌍을 행으로, 관계를 열로 가진 행렬로 트리플들을 표현하는 작업이다. 그림 2는 두 개의 트리플을 행렬화 하는 예시를 나타내고 있다. 관계 "장군(X,Y)"는 "X의 장군 Y"라는 관계이고, "수도(X,Y)"는 "X의 수도는 Y"라는 관계를 나타낸다.

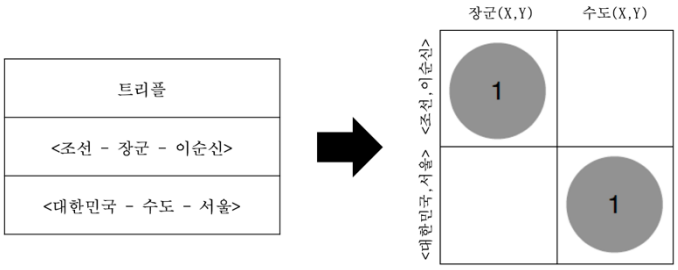


그림 2 트리플 행렬화의 예시

행렬화를 통해 지식베이스에 등록된 트리플은 1이고, 나머지 칸(slot)은 모두 0인 행렬 K를 얻을 수 있다.

2.2. 행렬 분해

행렬 분해 모델의 목표는 2.1에서 얻은 지식베이스 행렬 K의 근사 행렬 K'을 구하는 것이다. 2.2.1에서는 근사에 사용할 목적 함수(objective function)에 대해 소개한다. 2.2.2에서는 행렬 분해에 사용할 자연 매개 변수(natural parameter)들에 대해 설명한다. 2.2.3에서는 행렬 분해 알고리즘에 대해 설명한다.

2.2.1. 목적 함수

본 논문의 행렬 분해 모델에서 사용하는 목적 함수는 BPR[7]에서 사용한 목적 함수와 동일하다. 상품 추천 시스템에서 행과 열을 이루는 고객과 상품은 본 논문에서 각각 개체쌍과 관계에 대응된다. 어떤 개체쌍 $t = < sbj, obj >$가 관계 r_i을 관계 r_j보다 선호할 확률(r_i이 성립할 확률이 r_j가 성립할 확률보다 높을 확률을 의미한다)은 식(1)과 같이 정의한다.

$$p(r_i >_t r_j|\Theta) := \sigma(\hat{x}_{tij}(\Theta)) \quad \text{식(1)}$$

σ 는 로지스틱 시그모이드 함수(logistic sigmoid function)이며, Θ는 모델의 자연 매개 변수이다. $\hat{x}_{tij}(\Theta)$는 결과값이 실수인 함수이며, 행렬 분해 모델에 따라 달라지는 함수가 된다. 우리의 목적은 학습 데이터에 속한 트리플 (t, r_i)와 속해 있지 않은 트리플 (t, r_j)에 대해 $p(r_i >_t r_j|\Theta)$를 최대화하는 것이다. $\hat{x}_{tij}(\Theta)$는 식(2)와 같이 정의된다. $\hat{x}_{tij}(\Theta)$에서 변수 Θ는 앞으로 편의상 생략한다.

$$\hat{x}_{tij} := \hat{x}_{ti} - \hat{x}_{tj} \quad \text{식(2)}$$

여기서 $\hat{x}_{ti}$는 행렬 분해 모델에서 최종적으로 구하고자 하는 K'의 t번째 행, i번째 열의 값이다. 이는 2.2.2에서 중점적으로 다루게 될 것이다. 우리가 행렬 분해 모델에서 최대화하고자 하는 목적 함수는 식(3)과 같다. 이 목적 함수가 의미하는 바는 '학습 데이터에 부여하는 확률 값과 나머지 데이터에 부여하는 확률 값의 차이'다.

$$Obj = \sum_{(t,i,j) \in D} ln\, \sigma(\hat{x}_{tij}) - \lambda_\Theta \|\Theta\|^2 \quad \text{식(3)}$$

λ_Θ 는 모델에 따른 정규화 변수(regularization parameter)다. D의 정의는 식(4)와 같다.

$$D := \{(t,i,j) | (t,r_i) \in KB, (t,r_j) \notin KB\} \quad \text{식(4)}$$

2.2.2. 자연 매개 변수

본 논문에서는 $\hat{x}_{ti}$에 대한 총 5가지의 모델을 사용한다. $\hat{x}_{ti}$가 의미하는 것은 최적화된 지식베이스 행렬의 slot값이며, 개체쌍 t와 관계 i로 이루어진 트리플에 모델이 부여하는 확률 값이다. 각 $\hat{x}_{ti}$ 모델은 행렬 K를 분해하는 서로 다른 방법들이기도 하다. 모델들은 Riedel et al.(2013)[8]을 참고했음을 밝힌다. 각 모델의 이름, $\hat{x}_{ti}$의 정의, 자연매개변수 목록을 표 1에 정리하였다. 편의상 관계 r_i는 i로 표현하였다. f-model(latent feature model)은 개체쌍의 특성 벡터 $\mathbf{w}$와 관계의 특성 벡터 $\mathbf{h}$의 선형 결합으로 $\hat{x}_{ti}$ 를 정의한다. n-model(neighborhood model)은 관계 간의 weight matrix R을 정의한 뒤 어떤 트리플 (t,i)에 대해 $(t,j) \in KB$인 모든 관계 j를 찾고 관계 i와 j사이의 가중치 값 $r_{i,j}$의 합으로 $\hat{x}_{ti}$를 정의한다. nf-model은 n-model과 f-model의 합으로 정의한다. e-model(entity model)은 개체쌍 $t = (e_1, e_2)$ 에 대해 각 개체별 특성 벡터 e_{e_1}, e_{e_2}를 정의하여 이를 관계의 특성 벡터 d_1, d_2와의 선형 결합으로 $\hat{x}_{ti}$를 정의한다. d_1은 관계의 주어 개체 자리의 특성 벡터를, d_2는 목적어 개체 자리의 특성 벡터를 의미한다. nfe-model은 (n,f,e)-model들의 합으로 정의한다.

모델	$\hat{x}_{ti}$	Θ
f-model	$\hat{x}_{ti} = \sum_{k}^{dim} w_{t,k} h_{i,k}$	W, H
n-model	$\hat{x}_{ti} = \sum_{(t,j)\in KB} r_{i,j}$	R
nf-model	$\hat{x}_{ti} = \sum_{k}^{dim} w_{t,i} h_{t,i} + \sum_{(t,j)\in KB} r_{i,j}$	W, H, R
e-model	$\hat{x}_{ti} = \sum_{a=1}^{2} \sum_{k}^{dim} d_{a,k} e_{e_a,k}$	D, E
nfe-model	$\hat{x}_{ti} = \sum_{k}^{f\_dim} w_{t,i} h_{t,i} + \sum_{(t,j)\in KB} r_{i,j} + \sum_{a=1}^{2} \sum_{k}^{e\_dim} d_{a,k} e_{e_a,k}$	W, H, R, D, E

표 1 행렬 분해 모델

2.2.3. 행렬 분해 알고리즘

본 절에서 설명할 행렬 분해 알고리즘은 1과 0으로 이루어진 행렬 K를 입력 받아 목적 함수를 최대화하도록 확률적 경사 하강법(Stochastic Gradient Descent)을 통해 모델 별로 2.2.2에서 정의한 자연매개변수들을 최적화하여 반환한다. 최적화한 자연매개변수들을 다시 조합하여 K의 근사 행렬 K'를 출력할 수 있다. 그림 3은 행렬 분해 알고리즘의 의사코드(pseudocode)이다.

알고리즘 1 행렬 분해 알고리즘

```
1:  procedure OptimizeKBMF(Triples,Θ,model)
2:    initialize Θ
3:    repeat for batch_size
4:      randomly draw (t,i,j)from Triples
5:      Θ ← Θ + α(∂Objective Function/∂Θ)
6:    return Θ
```

그림 3 행렬 분해 알고리즘

본 알고리즘은 지식베이스에서 뽑아낸 트리플들의 리스트(Triples)와 자연매개변수(Θ), 그리고 모델 이름(model)을 입력으로 받아 학습된 자연매개변수를 반환한다.알고리즘의 핵심은 트리플 리스트로부터 트리플 하나 (t,i) 를 무작위로 추출하고, 리스트에 속하지 않은 트리플 (t,j)를 무작위로 추출하여(Line 4) 이를 (t,i,j)로 합쳐 확률적 경사 하강법을 진행하는 것이다(Line 5). α는 학습 정도(learning rate)이며, $\frac{\partial Objective\ Function}{\partial \Theta}$은 모델에 따라 달라진다. 정해진 반복 횟수(batch_size)만큼 샘플 추출 과 확률적 경사 하강법을 반복하여 최적화된 Θ를 반환하게 된다.

3. 실험

본 논문에서 소개한 KBMF의 평가를 위해 한국어 디비피디아를 대상으로 두 가지 실험을 진행하였다. 2.2에서 소개한 다섯 가지 모델들에 대한 성능평가 실험을 3.1에서, 추출한 지식에 대한 정밀도(precision) 평가 실험을 3.2에서 소개한다. 두 실험에서 사용한 데이터셋은 표 2에 정리되어 있다.

항목	개수
대상 관계 수	59
대상 관계를 3개 이상 가진 개체쌍	1217
트리플 수	3702

표 2 실험 데이터셋

3.1 모델 별 성능 평가

본 실험의 목적은 2.2에서 소개한 다섯 가지 행렬 분해 모델의 성능을 평가하는 것이다. 평가 방법으로는 one out evaluation scheme을 사용하였다. 각 개체쌍 별로 가지고 있는 관계 하나씩을 무작위로 골라 실험 데이터로 분리하고, 원래 데이터셋에서 실험 데이터를 제외한 나머지를 학습 데이터로 사용하였다. 학습 데이터를 입력으로 넣어 지식베이스 행렬에 행렬 분해 모델을 적용한 뒤 식(5)와 같이 AUC(area under the ROC curve)를 계산하였다.

$$AUC = \frac{1}{|T|}\sum_{t}\frac{1}{|E(t)|}\sum_{(i,j)\in E(t)}\delta(\hat{x}_{ti} > \hat{x}_{tj}) \quad \text{식(5)}$$

$E(t)$는 식(6)과 같이 정의한다.

$$E(t) := \{(i,j) | (t,i) \in S_{test} \wedge (t,j) \notin (S_{train} \cup S_{test})\} \quad \text{식(6)}$$

AUC가 높을수록 모델의 신뢰도가 높다고 할 수 있다. 각 트리플에 대해 무작위로 점수를 매기는 모델의 AUC값은 0.5에 근접할 것이며, AUC의 최대값은 1이다.

실험에서 사용한 hyperparameter들은 nfe-model을 기준으로 가장 좋은 성능을 보이는 값들이며, 표 3에 정리되어 있다. 기준을 nfe-model로 삼은 이유는 hyperparameter에 관계 없이 다른 모델들보다 항상 높은 성능을 보여주었기 때문이다. f_dim 은 W, H 의 dimension을 의미하고, e_dim은 D, E 의 dimension을 의미한다.

Hyperparameter	Value
f_dim	30
e_dim	3
batch_size	len(train_set) * 30
learning rate	0.03

표 3 Hyperparameters

각 모델의 AUC를 측정한 결과는 표 4에 정리되어 있다. 모델 별로 각각 10번씩 실험하여 얻은 AUC값의 평균을

냄으로써 측정하였다.

모델	AUC
f-model	0.6022
n-model	0.8825
nf-model	0.8770
e-model	0.5551
nfe-model	0.9530

표 4 모델별 AUC 측정 결과

다섯 모델 중에서 nfe-model이 0.9530의 가장 높은 성능을 보여주었다.

3.2 추출한 지식의 정밀도 평가

본 실험의 목적은 KBMF가 실제로 새로운 지식을 잘 추출해낼 수 있는지 알아보기 위함이다. 가장 높은 성능을 보였던 nfe-model을 적용한 행렬 분해를 통해 얻은 최적화된 지식베이스 행렬에서 학습 데이터를 제외한 나머지 중 점수가 가장 높은 100개의 트리플을 뽑아 실제로 맞는 트리플인지 사람이 직접 평가하였다. 표 5에 평가 결과를 관계 별로 정리하였다.

관계	트리플 수	참인 트리플	거짓인 트리플	정밀도
nationality	22	20	2	91%
deathPlace	19	4	15	21%
birthPlace	14	5	9	36%
city	14	14	0	100%
writer	9	1	8	11%
club	7	7	0	100%
director	6	1	5	17%
team	3	2	1	67%
producer	2	0	2	0%
routeEnd	2	1	1	50%
routeStart	1	0	1	0%
managerClub	1	1	0	100%
total	100	57	43	57%

표 5 상위 100개 트리플 정밀도 측정 결과

추출한 트리플 100개 중 57개가 참인 트리플로 평가되었다. 관계의 종류에 따라 정밀도가 큰 차이로 변화하는 것을 확인할 수 있다. 추출된 트리플 수가 5개 이상인 관계 중 nationality, city, club은 90%가 넘는 정밀도를 보인 반면, deathPlace, birthPlace, writer, director의 경우 40% 미만의 낮은 정밀도를 보였다.

높은 정밀도를 보였던 city가 추출된 개체쌍의 경우 해당 개체쌍의 학습 데이터에 routeStart, routeEnd, LocatedIn이 포함되어 있었다. 예를 들어 city가 추출된 개체쌍 <대구_도시철도_1호선 - 대구광역시>의 경우 나머지 세 관계가 학습 데이터에 포함되어 있었다. club의 경우 managerClub, team, position이 학습 데이터에 포함되어 있었고, nationality의 경우 birthPlace, deathPlace, position이 학습 데이터에 포함되어 있었다. 학습 데이터의 세 관계로부터 추출된 관계를 추론할 수 있는 경우에 높은 정밀도를 보여주었다고 판단할 수 있다. nationality에서 추출된 거짓인 트리플 <윌리엄_카를로스_윌리엄스 - nationality - 뉴저지_주>의 경우 '뉴저지_주'가 국가가 아니어서 nationality 관계가 성립할 수 없었기 때문이다.

낮은 정밀도를 보였던 deathPlace가 추출된 개체쌍의 경우 해당 개체쌍의 학습 데이터에 birthPlace, country, nationality가 포함되어 있었다. 이는 대부분의 인물 개체에서 birthPlace, country, deathPlace, nationality가 모두 동일하기 때문이다. 개체쌍이 birthPlace, country, nationality 관계를 가지는데 deathPlace를 가지지 않는 경우는 대부분 해당 인물 개체가 다른 곳에서 사망하였기 때문인데, 이 패턴은 학습되지 않았기 때문에 deathPlace 관계가 추출되었다고 판단된다. deathPlace 관계를 맞게 추출한 트리플 <원균 - deathPlace - 조선>의 경우 지식베이스에 deathPlace가 누락되어 있었다. birthPlace의 경우도 deathPlace와 유사하게 country, nationality, deathPlace가 학습 데이터에 있는 경우에 추출되었다. 낮은 정밀도를 보이는 이유 또한 동일하다. birthPlace, country, nationality, deathPlace 관계를 모두 가지는 개체쌍이 많지만, 출생지와 사망지가 다른 인물들에 대한 트리플을 주로 추출하였기 때문에 낮은 정밀도를 보인다고 해석된다.

4. 관련 연구

행렬 분해란 행렬 K를 행렬 T와 R의 곱으로 근사하는 것을 말한다. 행렬 분해는 이미지 처리, 문서 분류, 상품 추천 시스템 등에서 쓰이고 있다. 특히 상품 추천 시스템에서 행렬 분해를 이용한 연구가 많이 수행되었다.

상품 추천 시스템이란 특정 상품 목록에 대해 개인화된 순위를 매기는 문제를 푸는 작업을 의미한다. BPR(Bayesian Personalized Ranking from Implicit Feedback)[7]은 베이지안 분석 기법(Bayesian Analysis)에 의거한 순위 매김에 최적화 되어 있는 목적 함수를 처음 제안하고 확률적 경사 하강법을 이용한 행렬 분해 알고리즘을 선보였다. 제안된 알고리즘은 10,000명의 고객과 4,000개의 상품에 대한 426,612회의 구매가 기록된 로스만 데이터(Rossmann Dataset)에 대해 AUC 값이 최대 0.92까지 올라감을 보였다.

최근에는 관계추출 문제에서 행렬 분해를 적용하려는 연구들이 활발히 이루어지고 있다. 관계추출 문제에서 행렬 분해를 통해 얻고자 하는 것은 개체쌍과 관계 각각의 저차원 벡터 임베딩(low-dimensional vector embedding)이다. 특정 트리플에 부여되는 점수는 해당 트리플의 개체쌍 특성 벡터와 관계 특성 벡터의 선형결합으로 나타내어지게 된다. BPR[7]에서 선보인 목적 함수를 자연언어처리 분야에 적합하게 수정하고, 추가적인 모델들을 설계하여 행렬 분해를 영어 관계추출에 적용한 연구가 있다[8]. 이 연구에서 처음으로 제안한 유니버셜 스키마(universal schema)는 관계 추출 시에 특정한 지식베이스 스키마에 국한되지 않고 여러 지식베이스 스키마와 더불어 OpenIE[4]를 통해 자연언어문장에서 추출한

스키마까지 모두 통합된 스키마이다. 관계 추출 시에 유니버셜 스키마를 구축하고 행렬화 한 뒤 자신들이 제안한 행렬 분해 모델들(n-model, e-model, nf-model, nfe-model)로 전체 스키마를 근사하는 방법을 제안했다. Freebase와 NYTimes corpus를 데이터셋으로 한 실험에서 기존의 관계추출 방법들에 비한 성능 향상을 증명하였다. 이에 추가로 1차 논리(first-order logic)를 통합하여 고려한 행렬 분해 모델로 성능을 향상시킨 연구도 이루어졌다[9].

5. 결론

본 논문에서는 지식베이스 확장을 위한 내부적인 지식 획득 방법으로써 행렬 분해 모델 KBMF를 소개하였다. KBMF는 지식베이스에 등록된 개체쌍과 관계 각각에 대한 특징 벡터를 학습하여 새로운 지식의 순위를 매긴다. 한국어 디비피디아에 이를 적용하는 실험을 수행한 결과 본 새로운 지식의 순위를 매기는 데에 있어 높은 성능을 보인다는 사실을 확인하였다. 새로운 지식 추출에 있어서는 해당 지식의 관계 종류에 따라 정밀도에 큰 차이가 있음을 확인하였다. 낮은 정밀도를 보인 관계들의 경우 향후 자연언어문장으로부터 어휘적 관계를 추출하여 지식베이스와 통합하는 한국어 유니버셜 스키마에 본 논문에서 소개한 모델을 적용하는 연구를 통해 성능을 향상시키고자 한다. 또한 본 논문에서 소개한 방법을 적용한 지식베이스 확장 시스템 설계에 대한 연구, 그리고 다양한 언어 지식베이스에 대한 실험을 통해 본 모델이 언어의 종류에 구애 받지 않음을 증명하고자 한다.

사사

이 논문은 2017년도 정부(미래창조과학부)의 재원으로 정보통신기술진흥센터의 지원을 받아 수행된 연구임. (2013-0-00109, WiseKB: 빅데이터 이해 기반 자가학습형 지식베이스 및 추론 기술 개발)

참고문헌

[1] https://tac.nist.gov/2017/KBP/

[2] Ji, Heng, and Ralph Grishman. "Knowledge base population: Successful approaches and challenges." *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1.* Association for Computational Linguistics, 2011.

[3] Surdeanu, Mihai, and Heng Ji. "Overview of the english slot filling track at the tac2014 knowledge base population evaluation." Proc. Text Analysis Conference (TAC2014). 2014.

[4] Oren Etzioni, Michele Banko, Stephen Soderland, and Daniel S. Weld. 2008. Open information extraction from the web. Commun. ACM, 51(12):68-74

[5] Mausam, Mausam. "Open information extraction systems and downstream applications." Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence. AAAI Press, 2016.

[6] http://ko.dbpedia.org

[7] Rendle, Steffen, et al. "BPR: Bayesian personalized ranking from implicit feedback." *Proceedings of the twenty-fifth conference on uncertainty in artificial intelligence. AUAI Press*, 2009.

[8] Riedel, Sebastian, et al. "Relation Extraction with Matrix Factorization and Universal Schemas." *HLT-NAACL*. 2013.

[9] Rocktäschel, Tim, Sameer Singh, and Sebastian Riedel. "Injecting Logical Background Knowledge into Embeddings for Relation Extraction." *HLT-NAACL*. 2015.

[10] MLA Galárraga, Luis Antonio, et al. "AMIE: association rule mining under incomplete evidence in ontological knowledge bases." *Proceedings of the 22nd international conference on World Wide Web*. ACM, 2013.

딥러닝을 이용한 대규모 한글 폰트 인식

양진혁O, 곽효빈, 김인중

한동대학교, 전산전자공학부
yjh2067@gamil.com, gyqls1494@gmail.com, ijkim@handong.edu

Large-Scale Hangul Font Recognition Using Deep Learning

Jin-Hyeok YangO, Hyo-Bin Kwak, In-Jung Kim
School of Computer Science and Electrical Engineering, Handong Global University

요 약

본 연구에서는 딥러닝을 이용해 3300종에 이르는 다양한 한글 폰트를 인식하였다. 폰트는 디자인 분야에 있어서 필수적인 요소이며 문화적으로도 중요하다. 한글은 영어권 언어에 비해 훨씬 많은 문자를 포함하고 있기 때문에 한글 폰트 인식은 영어권 폰트 인식보다 어렵다. 본 연구에서는 최근 다양한 영상 인식 분야에서 좋은 성능을 보이고 있는 CNN을 이용해 한글 폰트 인식을 수행하였다. 과거에 이루어진 대부분의 폰트 인식 연구에서는 불과 수 십 종의 폰트 만을 대상으로 하였다. 최근에 이르러서야 2000종 이상의 대용량 폰트 인식에 대한 연구결과가 발표되었으나, 이들은 주로 문자의 수가 적은 영어권 문자들을 대상으로 하고 있다. 본 연구에서는 CNN을 이용해 3300종에 이르는 다양한 한글 폰트를 인식하였다. 많은 수의 폰트를 인식하기 위해 두 가지 구조의 CNN을 이용해 폰트인식기를 구성하고, 실험을 통해 이들을 비교 평가하였다. 특히, 본 연구에서는 3300종의 한글 폰트를 효과적으로 인식하면서도 학습 시간과 파라미터의 수를 줄이고 구조를 단순화하는 방향으로 모델을 개선하였다. 제안하는 모델은 3300종의 한글 폰트에 대하여 상위 1위 인식률 94.55%, 상위 5위 인식률 99.91%의 성능을 보였다.

주제어: 한글폰트인식, 딥러닝, CNN, ResNet

1. 서론

폰트는 디자인 분야에 있어서 필수적인 요소이며 문화적으로도 중요하다. 한글 폰트 인식은 우리의 문자인 한글의 아름다움과 중요성을 보존하고 홍보하기 위해 유용한 기술이다. 폰트를 구분하기 위해서는 문자 영상에 존재하는 지역적인 세부 형태를 효과적으로 구분해야 한다. 폰트 인식 연구는 해외에서 영어권 언어나 중국어를 중심으로 진행되었다. 최근 Google, Adobe, Snapchat 등의 글로벌 IT 기업들은 딥러닝을 폰트 인식에 적용해 수 천 종의 폰트에 대해서도 좋은 성과를 얻었다[1]. 그러나, 한글은 영어권 언어보다 문자 수가 훨씬 많기 때문에 한글 폰트 인식은 영어권 폰트 인식보다 어렵다. 과거에 이루어진 대부분의 한글 폰트 인식 연구들은 불과 수 십 종의 폰트 만을 인식 대상으로 하고 있다.

본 연구에서는 최근 영상인식 분야에서 좋은 성능을 보이고 있는 CNN을 이용해 3300종에 이르는 다양한 한글 폰트를 인식하였다. 두 가지 다른 구조의 CNN을 이용해 폰트인식기를 구성하고, 실험을 통해 이들을 비교 평가하였다. 특히, 본 연구에서는 폰트 인식에 필요한 지역적 세부 특징을 효과적으로 추출하면서도 학습 시간과 파라미터의 수를 줄이는 방향으로 모델을 개선하였다. 제안하는 모델은 3300종의 한글 폰트에 대하여 상위 1위 인식률 94.55%, 상위 5위 인식률 99.91%의 성능을 보였다.

2. 관련 연구

최근 해외에서는 딥러닝을 이용한 폰트 인식 방법들이 제안되었다. [1]에서는 영상인식에 성능이 우수한 딥러닝 모델인 CNN(Convolutional Neural Network)을 이용해 2383종의 폰트를 인식하였다. 학습 데이터로는 기본 폰트 영상에 다양한 변형을 적용함으로써 더 많은 영상을 추가해 사용하였다. 또한, 원활한 학습을 위해 SCAE (Stacked Convolutional Auto-Encoder)를 이용해 사전 학습을 수행한 후, 폰트 인식을 위한 교사 학습을 진행하였다.

해외에서는 폰트 인식 연구가 활발히 이루어졌으며, 딥러닝을 비롯한 첨단 기술들도 많이 적용되었다. 그러나, 국내에서는 폰트 인식 연구가 많이 이루어지지 않았을 뿐 아니라 주로 전통적인 특징 추출 알고리즘에 의존하고 있다. 한글은 2350개의 문자를 포함하고 있으며 현재 한국에서는 3000종이 넘는 한글 폰트들이 사용되고 있다. 문자 및 폰트의 종류가 많을수록 폰트를 정확히 인식하는 것은 어렵다. 따라서, 한글 폰트를 인식하는 것은 영어권 폰트 인식에 비해 난도가 높다. 지금까지 수행된 대부분의 한글 폰트 인식 연구들은 불과 수 종의 폰트 만을 인식 대상으로 하였다. 1997년도에 진행된 연구에서는 한글 문서의 폰트를 MLP(Multi-layer Perceptron)를 이용해 학습하였다[2]. MLP를 학습시키기 위해서 몇 가지 특징을 사용했는데, 문서에서 일정한 크기의 블록을 추출해서 수직방향과 수평방향으로 FFT(Fast Fourier Transform)를 수행한 후, 각 방향에 대해서 평균을 취하고, 그 결과 중에서 64개의 특징 값을 추출했다. 이와 같이 추출한 특징을 이용해 명조체, 신명조체, 견명조체, 고딕체, 중고딕체, 견고딕체, 궁서체, 샘물체, 필기체, 그래픽체라는 한글 문서의 기본이

되는 10가지 폰트를 인식하여 평균 95.19%의 인식률을 얻었다.

한글 문자 인식 연구는 폰트 인식보다는 많이 수행되었다. [3]에서는 CNN을 이용해 필기 한글을 인식하였는데, 4개의 컨볼루션계층과 4개의 맥스풀링(max-pooling)계층, 그리고 2개의 완전연결계층(fully-connected)으로 구성된 CNN을 이용하였다. 이 연구에서는 520자의 조합에 대한 성능은 97.67%, 2350자의 조합에 대한 성능은 96.34%의 정확도를 보였다.

3. 시스템 구성

3.1. 데이터 구성

본 연구는 한글 폰트 3300종을 인식 대상으로 한다. 각 폰트는 한글 2350자를 포함하고 있으므로 총 7,755,000(3300x2350)가지 폰트-문자 조합이 존재한다. CNN의 학습과 평가에는 48x48 크기의 문자 영상들을 사용하였다. 총 7,755,000개의 문자 영상들을 학습 데이터, 검증 데이터, 평가 데이터로 나누었으며, 각각의 데이터 셋은 전체 데이터의 80%, 10%, 10%의 비율로 랜덤 분할하였다.

3.2. 폰트 인식 CNN의 구성

본 연구에서는 두 가지 모델을 사용하였는데, 각 모델의 구성은 그림 1과 같다. 폰트를 효과적으로 인식하기 위해서는 문자 영상의 지역적인 세부 특징을 추출해야 한다. 이를 위해 본 연구에서 사용한 CNN은 다음과 같은 특징을 갖는다.

3.2.1. CNN 기본 모델

CNN의 상위 계층에서는 고수준 특징(high-level feature)을 추출하고 하위 계층에서는 저수준 특징(low-level feature)을 추출한다[4]. 고수준 특징은 영상의 전역적인 특성을 잘 표현할 뿐 아니라, 변이에 강한 장점이 있다. 반면 저수준 특징은 지역적인 세부 형태를 잘 반영하는 장점이 있다. ImageNet 데이터 등 복잡한 영상에 사용되는 VGG, GoogLeNet 등의 CNN들이 매우 많은 수의 계층으로 구성된다[5][6]. 그러나, 본 연구에서는 지역적 세부 형태를 잘 추출하기 위해 비교적 적은 수의 계층으로 이루어진 [3]의 모델을 기본 모델로 택하였다. 적은 수의 계층을 사용할 경우 폰트 인식에 필요한 저수준 특징을 잘 반영할 수 있을 뿐 아니라 학습 시간과 파라미터의 수를 줄이는 데에도 바람직하다. 그러나, [3]의 모델은 한글 인식을 위해 설계되었으므로 폰트 인식에 좀 더 적합하도록 다음과 같은 변형을 적용하였다.

3.2.2. 모두 컨볼루션으로 구성된 네트워크

과거의 한글 인식 연구에서는 맥스풀링(max-pooling)계층을 통해 컨볼루션 계층에서 추출한 특징 벡터를 추상화하고 특징맵의 크기를 축소하였다[3]. 맥스풀링계층은 차원 축소 및 추상화 과정에서 특징의 위치 변이를 흡수하는데, 그 결과 폰트 인식에 필요한 지역적 세부 형태 정보가 소실된다. 본 연구에서는 이러한 문제점을 극복하기 위해 [7]과 같이 맥스풀링계층을 모두 동일한 크기의 커널과 보폭을 갖는 컨볼루션계층으로 대체했다.

3.2.3. 잔류 연결(Residual Connection)

폰트 인식에는 획에서의 미세한 차이로도 폰트의 종류가 달라지기도 한다. 따라서, 폰트를 효과적으로 인식하기 위해서는 추상화 수준이 높은 고수준 특징뿐 아니라 세부 형태를 반영하는 저수준 특징들도 요구된다. 이를 위해 본 연구에서는 저수준 특징들이 정보를 보존한 상태로 상위 계층까지 전달되기 위해 잔류 연결(residual connection)을 적용했다. 심층신경망이 잔류 연결을 포함할 경우 얕은 네트워크를 병렬적으로 연결한 것과 유사한 효과를 얻을 수 있는데[8], 그로 인해 하위 계층들이 추출한 저수준 특징들을 상위 계층까지 잘 전달할 수 있다.

3.2.4. 전역 평균 풀링(Global Average Pooling)

CNN의 완전연결계층은 파라미터의 수가 매우 많아 과적합(over-fitting)이 많이 발생하는 것으로 알려져 있다. 이를 완화하기 위해 본 연구에서는 [9]과 같이 완전연결계층 대신 CCCP(Cascaded Cross Channel Pooling) 계층과 전역평균풀링(global average pooling)을 사용하였다.

3.2.5. 폰트 인식 CNN의 학습

CNN의 학습 알고리즘으로는 RMSProp(Root Mean Square Propagation) 최적화 알고리즘과 모멘텀(momentum) 최적화 방법을 결합한 ADAM 최적화(ADAptive Momentum estimation optimizer) 알고리즘 [12]을 사용하였다. ADAM 최적화는 최근 많은 연구에서 좋은 성능을 보이고 있다.

또한, Xavier 초기화와 배치정규화를 함께 사용함으로써 학습 속도를 개선하였다. 최근에는 많은 수의 계층으로 구성된 CNN에서는 He 초기화를 Xavier 초기화 알고리즘보다 더 많이 사용하는 추세이다[10][11]. 그러나, 본 연구에서 두 알고리즘을 적용해 본 결과 Xavier 초기화 알고리즘이 근소하게 좋은 성능을 보였다. 이는 본 연구에서 사용한 CNN이 비교적 적은 수의 계층으로 구성되었기 때문으로 추정된다.

다수의 계층으로 구성된 CNN의 학습에는 내부 공변량 이동(internal covariate shift)문제가 발생한다. 이를 해결하기 위한 방법으로 배치정규화가 널리 사용된다[13]. 3300가지 폰트를 분류해야 하기 때문에, 초기값과 학습률을 세부적으로 고려하여 설정해주지 않으면 경사도(gradient)가 폭발적으로 증가하거나 소실될 수 있다. 배치정규화는 각 배치(batch)와 계층마다 정규화를 수행함으로써 이러한 문제들을 해결해주며 학습 시간도 크게 단축시킨다. 동일한 모델을 학습 할 경우에도 배치정규화를 사용할 경우 3배 이상의 수렴속도를 보였다.

동일한 폰트 내의 문자 영상들은 지역적 형태가 매우 유사하기 때문에 폰트인식기의 학습에서는 학습 데이터들의 배열에 섬세한 주의가 요구된다. 또한, 학습에

사용되는 배치정규화(batch normalization) 알고리즘은 각 배치별로 특징들의 분포를 추정하기 때문에, 모든 폰트 3300종에 대해 한글 조합 2350자가 고르게 섞이도록 하는 것이 중요하다. 본 연구에서는 많은 수의 문자 영상들을 고르게 분포하도록 하기 위해 모든 폰트-한글 조합에 대한 인덱스를 만들고, 매 반복마다 고르게 섞은 후 각 인덱스가 가리키는 폰트-문자 영상을 읽어와 학습에 사용하였다.

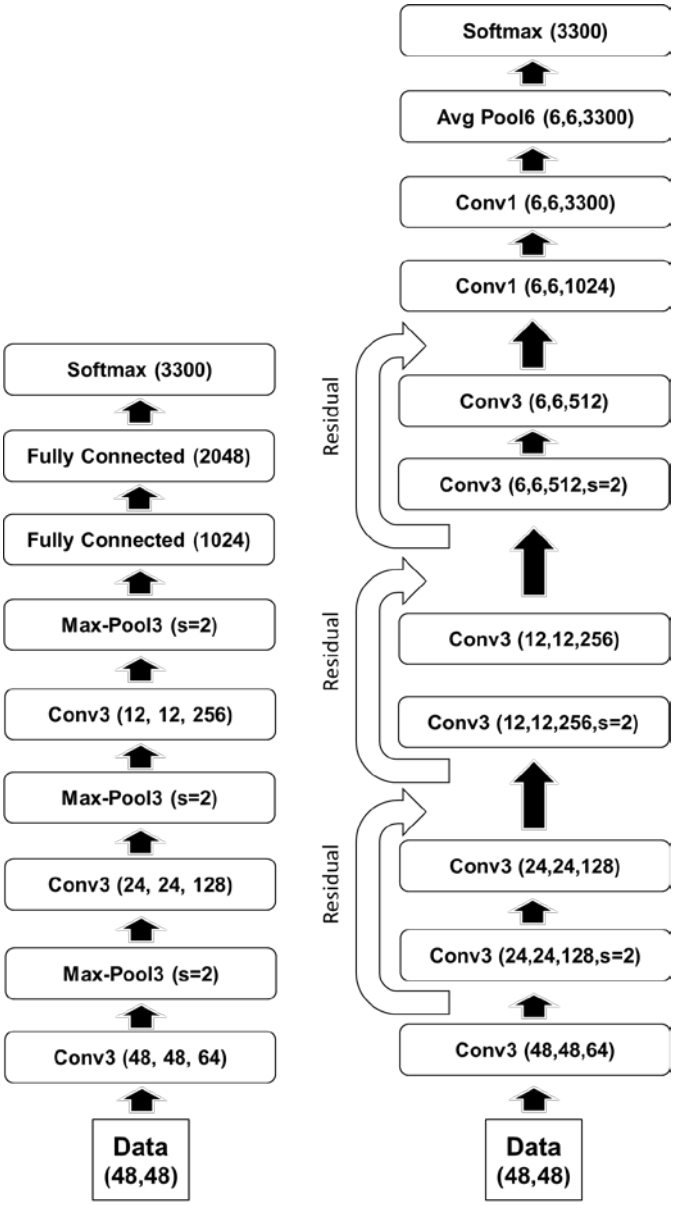


그림 1 모델 구성. 기본 모델 A(좌), 제안하는 모델 B(우).

각 상자 안의 이름은 계층의 종류, 그 옆의 숫자는 커널의 크기, 괄호안의 숫자는 차례대로 이미지의 높이, 넓이, 채널(혹은 노드)의 개수이다. 모든 은닉계층의 활성화 함수는 ReLU를 사용했다.

4. 실험 결과

표 1 두 모델의 폰트 인식률

	기본 모델 (A)	제안하는 모델 (B)	상대적 오차감소율
상위 1위 인식률	88.29%	94.55%	53.46%
상위 5위 인식률	99.03%	99.91%	90.72%

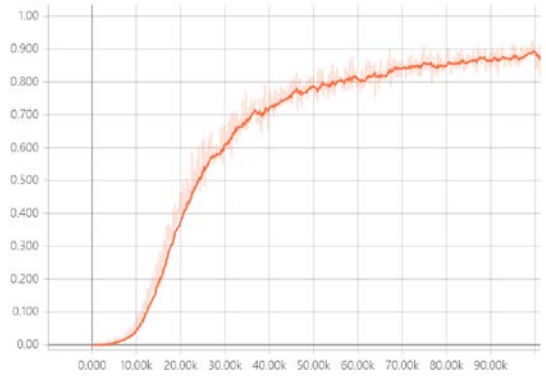


그림 2 모델 A 학습 과정의 인식률 변화

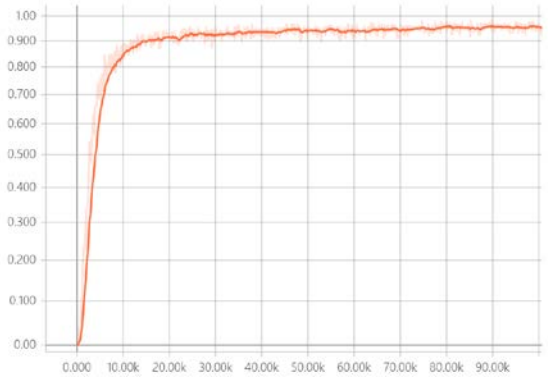


그림 3 모델 B 학습 과정의 인식률 변화

4.1. 실험 환경

학습을 위한 실험 환경으로 Intel i7-6700K 4.00GHz CPU, GeForce GTX-1080 GPU 2개, SSD, 32GB Ram를 사용하여 실험을 진행했다. 두 가지 모델에 대해 학습을 위한 배치의 크기는 256, 학습의 횟수는 10만 번으로 동일하게 실험했다. 모델 A는 학습률을 0.001로 설정하고 학습하였고, 모델 B는 배치정규화를 이용했기 때문에 높은 학습률인 0.1로 설정하고 학습했다. 배치정규화를 적용하지 않은 모델 A에 높은 학습률을 적용할 경우 학습이 진행되지 않았다.

4.2. 모델 별 실험 결과

그림 2와 그림 3은 모델 A와 모델 B의 학습 횟수에 따른 인식률의 변화를 나타낸 그래프이다. x축은 학습 횟수이며, y축은 인식률(%)이다. 모델 B의 수렴속도가

모델 A의 수렴속도보다 훨씬 빨랐다. 특히, 배치정규화를 적용하고 학습률을 높게 설정한 것이 수렴 속도에 큰 영향을 미쳤다.

표 1은 두 모델에 대해 상위 1위 인식률과 상위 5위 인식률에 대한 실험 결과이다. 모델 A보다는 모델 B가 더 우수한 성능을 보였다. 1위 인식률과 5위 인식률의 절대적 오류 감소는 각각 6.26%와 0.88%였다. 그러나, 상대적 오차 감소율은 1위 인식률이 53.46%와 5위 인식률이 90.72%로 나타나 3장에서 기술한 방법들이 폰트 인식 성능 개선에 효과적이었음을 확인할 수 있었다.

그림 4 폰트 인식 결과 예. 인식률 상위 2위의 폰트가 정답 폰트와 얼마나 유사한지를 나타내는 예시. 폰트 별로 3개의 이미지가 있으며, 점선을 기준으로 왼쪽은 정답 폰트와 인식률, 오른쪽은 해당 폰트에 대한 예측률이 높은 상위 2개의 폰트와 예측률. 파란색 글자는 정답 폰트와 일치하는 예측 폰트. 3300가지의 폰트에 대한 예측 시각화 중 12가지의 샘플 예시.

오인식 분석 결과 거의 같은 모양을 가진 폰트가 많았는데, 이러한 폰트들은 육안으로도 구분하기 어려웠다. 그림 4는 폰트 인식 결과의 예이다. 일부 폰트들은 형태가 단순함에도 불구하고 인식률이 낮게 측정되었는데, 이러한 폰트들은 다른 폰트와 매우 유사하거나 아예 동일한 형태를 갖는 경우가 많았다. 이러한 폰트들은 사실상 구분이 매우 어려웠다. 반면, 형태가 복잡한 그림체 폰트들은 동일한 형태의 폰트가 없어서 높은 인식률을 보이는 경우도 있었다.

5. 결론

본 논문에서는 CNN을 이용해 3300종의 한글 폰트를 인식하였다. 지역적인 세부 특징을 효과적으로 추출하기 위해 비교적 적은 수의 계층을 사용하였으며, 최상단 계층 외에는 모두 컨볼루션 계층으로만 구성된 CNN을 사용하였다. 또한, 저수준 특징이 최상단까지 잘 전달되도록 하기 위해 잔류 연결을 적용하였다. 그 결과 3300종의 폰트 전체에 대하여 1위 인식률 94.55%, 5위 인식률 99.91%의 높은 정확도를 보였다.

감사의 글

- 본 연구는 문화체육관광부 및 한국콘텐츠진흥원의 2017년도 문화기술 연구개발 지원사업으로 수행되었음 (2017000130)

- 본 연구는 미래창조과학부 및 정보통신기술진흥센터의 SW중심대학지원사업의 연구결과로 수행되었음

참고문헌

[1] Z. Wang, et. al., "Deepfont: Identify your font from an image," Proceedings of the 23rd ACM international conference on Multimedia. ACM, 2015.

[2] 박문호, et. al., "인공지능: 인쇄된 한글 문서의 폰트 인식.", 정보처리학회논문지 제4권 8호, pp. 2017-2024., 1997.

[3] I. Kim, C. Choi, and S. Lee, "Improving discrimination ability of convolutional neural networks by hybrid learning," International Journal on Document Analysis and Recognition (IJDAR), vol. 19, no.1, pp. 1-19, 2016.

[4] Ranjan, Rajeev, Vishal M. Patel, and Rama Chellappa. "Hyperface: A deep multi-task learning framework for face detection, landmark localization, pose estimation, and gender recognition," arXiv preprint arXiv:1603.01249, 2016.

[5] K. Simonyan, and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," arXiv preprint arXiv:1409.1556, 2014.

[6] C. Szegedy, et. al., "Going deeper with convolutions," Proceedings of the IEEE conference on computer vision and pattern recognition, 2015.

[7] J. Springenberg, et al. "Striving for simplicity: The all convolutional net," arXiv preprint arXiv:1412.6806, 2014.

[8] A. Veit, M. Wilber, and S. Belongie, “Residual Networks Behave Like Ensembles of Relatively Shallow Networks,” Advances in Neural Information Processing Systems. 2016.

[9] M. Lin, Q. Chen, S. Yan, “Network In Network,”

arXiv preprint arXiv:1312.4400, 2014.
[10] X Glorot, Y Bengio, “Understanding the difficulty of training deep feedforward neural networks,” PMLR, pp. 249-256, 2010.
[11] K. He, X. Zhang, S. Ren and J. Sun, “Delving deep into rectifiers surpassing human-level performance on imagenet classification,” arXiv preprint arXiv:1502.01852, 2015.
[12] Kingma, Diederik, and Jimmy Ba. "Adam: A method for stochastic optimization," arXiv preprint arXiv:1412.6980, 2014.
[13] S. Ioffe, and C. Szegedy. "Batch normalization: Accelerating deep network training by reducing internal covariate shift," International Conference on Machine Learning, 2015.

한국어에서 Attention 모델과 Naïve Bayes 모델 기반의 어휘 말뭉치 구축 및 응용에 관한 연구

윤주성○, 김현철

고려대학교
{xelloss705, hkim64}@gmail.com

Attention and Naïve Bayes Models based Lexicon Corpus and Applications for Korean

Joosung Yoon○, Hyeoncheol Kim
Korea University

요 약

감성 분석에서 어휘 말뭉치는 기존의 전통적인 기계학습 방법에서 중요한 특징으로 사용되었다. 최근 딥러닝의 발달로 hand-craft feature를 사용하지 않아도 되는 End-to-End 방식의 학습이 등장했다. 하지만 모델의 성능을 높이기 위해서는 여전히 어휘말뭉치와 같은 특징이 모델의 성능을 개선하는데 중요한 역할을 하고 있다. 본 논문에서는 이러한 어휘 말뭉치를 Attention 모델과 Naïve bayes 모델을 기반으로 구축하는 방법에 대해 소개하며 구축된 어휘 말뭉치가 성능에 끼치는 영향에 대해서 Hierarchical Attention Network 모델을 통해 분석하였다

주제어: 한국어 어휘 말뭉치, Attention, Naïve bayes

1. 서론

감성 분석(sentiment analysis)는 자연어처리 분야에서 가장 기본적인 태스크 중에 하나로써, 해당 문서의 감성이 긍정, 부정, 중립 중 어디에 속했는지 구분하는 것을 의미한다. 감성 분석을 위한 전통적인 기계학습 방법으로는 SVM[1], Naïve bayes[3] 등이 사용되었고 어휘(lexicon)기반의 특징(feature)을 활용한 연구도 많이 진행되었다[2].

최근 딥러닝(deep learning)의 발달[4]로, 이러한 어휘 기반의 특징인 hand-craft feature를 이용하는 것이 아닌 End-to-End 방식으로 문제를 해결하는 것이 가능해졌다. 하지만 End-to-End 방식으로 모든 문제가 해결되는 것이 아니며, 학습 데이터가 부족하거나 노이즈가 많을 경우 딥러닝을 써도 성능이 높게 나오지 않을 수 있다. 이러한 경우 전통적인 기계학습 방법에 사용되었던 hand-craft feature를 추가적으로 사용하여 모델을 개선할 수 있다.최근 연구 결과에 따르면 어휘 특징(lexicon feature)은 문서 분류를 위한 딥러닝 모델에서도 모델의 성능을 높이기 위해 여전히 유용한 특징으로 알려져 있다[2]. 그러므로 이러한 어휘 말뭉치(lexicon corpus)를 구축하는 일은 모델의 성능을 향상시키기 위해 필요하다고 할 수 있다.

본 논문에서는 Attention 모델과 Naïve bayes 모델을 사용으로 어휘 말뭉치를 구축하고 그것의 효과에 대해서 연구하였다.

본 논문의 구성은 2장에서 관련 연구 및 모델에 대해서 설명하고, 3장에서 실험에 대해서 서술한다. 4장에서는 결론 및 향후 연구에 대해서 다룬다.

2. 관련 연구

2.1 Naïve Bayes (NB)

Naïve bayes 모델은 아래와 같이 문서 d가 어떤 클래스 $c \in C$에 속하는지 확률을 사용해서 분류하는 모델이다[3].

$$c^{predict} = argmax_c P_{NB}(c|d)$$

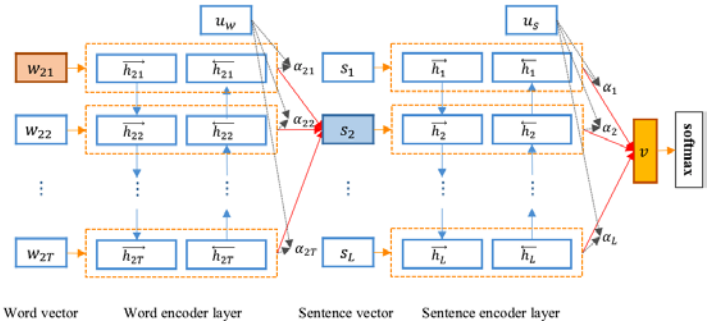
$$P_{NB}(c|d) := \frac{P(c)\sum_{i=1}^{m} P(f|c)^{n_i(d)}}{P(d)}$$

위 식에서 f는 특징(feature)을 의미하며, $n_i(d)$는 f_i 특징의 횟수를 의미한다. Naïve bayes 모델에서 f이 모델에 끼치는 영향을 분석해서 어휘 말뭉치를 구축하는데 사용했다.

2.2 Hierarchical Attention Networks (HAN)

HAN 모델[5]은 GRU[6]와 Attention이 계층적으로 이루어진 모델이며 문장 단위의 GRU와 문장을 이루는 단어 단위의 GRU로 이루어져있다. 각 GRU layer 위에 Attention layer가 있으며 단어에 대한 Attention과 문장에 대한 Attention을 계산한다. 단어에 대한 Attention은 모델이 학습될 때 같이 학습되는 단어의 컨텍스트 벡터 u_w가 어떻게 학습되는지에 따라 달라지며 본 논문에서는 단어에 대한 Attention을 어휘 말뭉치를

구축하는데 사용했다.



<그림 1: Hierarchical Attention Network 모델 구조>

2.2 형태소 분석

형태소 분석은 KoNLPy를 사용했다[8]. HAN에서 문서를 문장 단위로 나눠줄 때는 꼬꼬마 형태소 분석기[9]를 사용했으며 문장 내에서 형태소 분석을 할 때는 한국어 트위터 형태소 분석기[10]를 사용했다.

3. 실험

3.1 평가 방법

평가를 위한 기준은 Accuracy를 사용했으며, HAN 모델과 HAN 모델에 어휘 말뭉치 특징을 추가했을 때의 성능 변화를 비교했다.

3.2 말뭉치

말뭉치는 네이버로부터 얻은 영화 평점 데이터[1]를 사용했으며 중립 리뷰는 포함하지 않았다. 긍정적인 리뷰의 경우 9-10점 리뷰들로 구성되어있으며 부정적인 리뷰는 1-4점 리뷰로 구성되어 있으며 데이터의 크기는 <표 1>과 같다. 각 리뷰는 영화당 100개의 140자평을 초과하지 않는다. 모델에 사용한 단어 종류는 16,931개이며 등장 빈도가 4이하인 단어는 제외했다.

<표 1: 영화 리뷰 데이터>

	긍정	부정	문장당 평균 단어 수
Train	750,000	750,000	11.34
Test	25,000	25,000	11.37

3.3 어휘 말뭉치 구축

말뭉치로부터 어휘 말뭉치를 구축하기 위해 Naïve baye 모델과 HAN 모델을 학습 후 사용했다. Naïve bayes 어휘 말뭉치의 경우, Naïve bayes 모델에 가장 영향을 많이 끼친 단어 상위 2,200개를 다음과 같은 기준으로 선택했다.

$$w = argmax_w\left(\frac{max\ P_{NB}(w|c_i)}{min\ P_{NB}(w|c_i)}\right)$$

Naïve bayes 어휘 말뭉치에서 각 단어에 대한 감성 점수는 $max\ P_{NB}(w|c_i)\ /\ min\ P_{NB}(w|c_i)$의 값들에 대해서 0에서 1사이로 정규화 해서 나타냈다.

Attention 어휘 말뭉치의 경우, Attention모델이 문서를 분류 할 때 단어에 대한 Attention weight가 높은 단어에 대해서 상위 2,200개를 선택하고 weight기반으로 각 단어에 점수를 부여했다. 이때 형태소 분석기로부터 Josa, Verb, Punctuation이 태깅 된 단어들은 제외하였다. 어휘 말뭉치는 단어와 단어에 대응되는 감성 점수로 구성하였으며 각 점수는 0에서 1사이로 정규화 하였다.

<표 2: Attention과 NB의 어휘 말뭉치 상위 단어 비교>

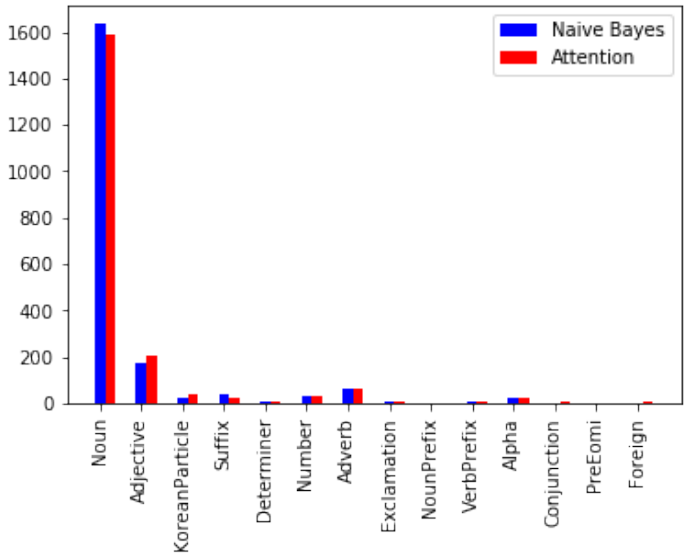
	Attention	Naïve Bayes
긍정	영화/Noun	울컥/Adverb
	재밌다/Adjective	♥/Foreign
	ㅋㅋ/KoreanParticle	Good/Alpha
	좋다/Adjective	펑펑/Noun
	너무/Noun	♥♥♥/Foreign
	정말/Noun	꿀잼/Noun
	최고/Noun	♡/Foreign
	재미있다/Adjective	아련하다/Adjective
	있다/Adjective	최고다/Noun
	ㅠㅠ/KoreanParticle	쵝오/Noun
부정	영화/Noun	최악/Noun
	없다/Adjective	낚였/Noun
	ㅋㅋ/KoreanParticle	낭비/Noun
	재미없다/Adjective	반개/Noun
	아깝다/Adjective	빵점/Noun
	점/Noun	노잼/Noun
	너무/Noun	하품/Noun
	이/Noun	기세/Noun
	쓰레기/Noun	이도/Noun
	진짜/Noun	개뿔/Noun

구축된 Attention 기반 어휘 말뭉치에서 긍정과 부정의 첫번째 단어가 영화/Noun 가 나온 것을 확인할 수 있었다. 이는 빈도수 기반의 Naïve bayes와는 달리 학습된 컨텍스트 벡터(context vector)를 통해 단어에

[1] http://github.com/e9t/nsmc/

Attention을 주는 모델의 특성이 반영된 것으로 보인다.

<그림 2>에 의하면, 전체적으로는 Noun이 태깅 된 단어는 Attention 어휘 말뭉치보단 Naïve bayes 어휘 말뭉치에서 더 많이 분포되어 있는 것으로 나타났다. 그리고 Naïve bayes 어휘 말뭉치에서는 Noun이 태깅 된 단어가 Attention 어휘 말뭉치에 비해 더 상위 순위에 있는 것을 확할 수 있었으며 Attention의 경우 Adjective, KoreanParticle등 다양한 품사를 가진 단어들이 상위에 분포하는 것을 확인 할 수 있었다.



<그림 2: 어휘 말뭉치 태깅 종류 분포>

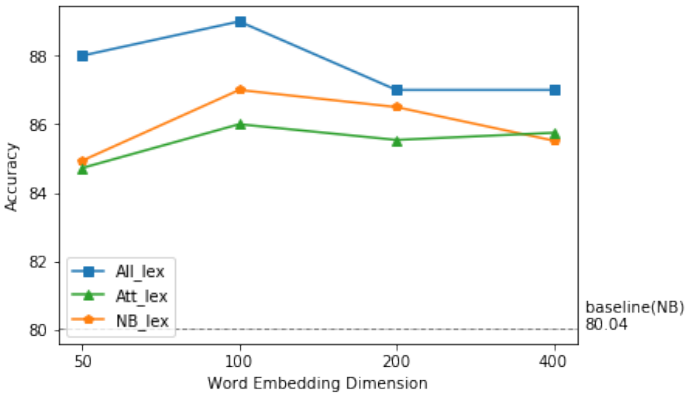
3.4 학습 파라미터

HAN 모델에 대한 학습 파라미터는 기본적으로 Zichao가 제안한 구성[5]과 같다. 모델 내의 Word embedding은 200 차원, Bidirectional GRU는 100차원, 단어와 문장의 컨텍스트 벡터는 100차원이다. 학습을 위한 파라미터는 다음과 같다. 모델 훈련을 위한 파라미터는 Stochastic gradient descent의 learning rate는 0.1, 모멘텀은 실험적으로 0.6의 파라미터로 설정했다. 모든 모델의 훈련을 위한 epoch은 20, batch size는 65로 설정했다.

Word embedding의 차원의 크기에 따른 성능 비교를 위해 차원의 크기를 50, 100, 200, 400으로 나눠서 실험하였다.

3.4 실험 모델

Attention과 Naïve bayes 모델을 통해 구축한 어휘 말뭉치가 감성 분석에 끼치는 영향을 알아보기 위해 HAN 모델에 각각의 어휘말뭉치를 단어에 대한 점수 벡터로 변환하여 Word embedding 벡터[11]에 붙여서 모델을 구성했다. <그림 3>에 의하면 Attention 기반 어휘 말뭉치와 Naïve bayes 기반 말뭉치를 모두 사용했을 때 가장 높은 성능을 나타냈으며 Word embedding이 100차원일 때 가장 성능이 높은 것으로 나타났다. 대체적으로 Attention 기반 어휘 말뭉치보다는 Naïve bayes 기반 어휘 말뭉치가 감성분석에서 성능을 높이는데 더 큰 영향을 주는 것으로 나타났다.



<그림 3: 어휘 말뭉치에 따른 모델 성능>

4. 결론 및 향후 연구

본 논문에서는 Attention과 Naïve bayes모델을 기반으로 어휘 말뭉치를 구축하고 그 효과에 대해서 감성 분석을 통해 평가하였다. 실험결과 Attention 어휘말뭉치는 Noun, Adjective, KoreanParticle등 다양한 형태소로 구성되었고 Naïve bayes 어휘 말뭉치는 주로 Noun 형태소로 구성된 것을 확인 할 수 있었다. 각 어휘말뭉치를 HAN 모델에 적용하여 감성 분석 실험결과 대체적으로 Attention과 Naïve bayes 어휘 말뭉치를 모두 사용한 모델이 가장 성능이 높았으며 각각을 따로 적용한 경우 Naïve bayes 어휘 말뭉치가 더 성능이 높음을 확인 할 수 있었다.

본 연구에서는 긍정과 부정을 분류하는 감성 분석을 통해 어휘말뭉치의 효과를 검증했지만 각 클래스당 어휘말뭉치를 생성해서 텍스트 분류(text classification)[12]에서는 이러한 연구가 진행된 바가 없다. 대부분의 분류 문제는 클래스가 2개 이상이므로 향후 연구에서는 텍스트 분류에 대해서도 적합한 어휘말뭉치를 구축하고 적용하는 연구가 필요할 것으로 보인다.

감사의 글

" 이 논문은 2017년도 정부(미래창조과학부)의 재원으로 한국연구재단의 지원을 받아 수행된 연구임 (NRF-2017R1A2B4003558)."

참고문헌

[1] Mullen, Tony, and Nigel Collier. "Sentiment Analysis using Support Vector Machines with Diverse Information Sources." EMNLP. Vol. 4. 2004.

[2] Shin, Bonggun, Timothy Lee, and Jinho D. Choi. "Lexicon integrated cnn models with attention for sentiment analysis." arXiv preprint arXiv:1610.06272, 2016.

[3] Go, Alec, Richa Bhayani, and Lei Huang. "Twitter

sentiment classification using distant supervision." CS224N Project Report, Stanford 1, 12, 2009.
[4] LeCun, Yann, Yoshua Bengio, and Geoffrey Hinton. "Deep learning." Nature 521.7553: 436-444, 2015.
[5] Yang, Z., Yang, D., Dyer, C., He, X., Smola, A. J., & Hovy, E. H. Hierarchical Attention Networks for Document Classification. In HLT-NAACL, pp. 1480-1489, 2016.
[6] Bahdanau, D., Cho, K., &Bengio, Y. Neural machine translation by jointly learning to align and translate. arXiv preprint arXiv:1409.0473, 2014.
[7] Chung, J., Gulcehre, C., Cho, K., & Bengio, Y. Empirical evaluation of gated recurrent neural networks on sequence modeling. arXiv preprint arXiv:1412.3555, 2014.
[8] 박은정, 조성준, “KoNLPy: 쉽고 간결한 한국어 정보처리 파이썬 패키지”, 제 26회 한글 및 한국어 정보처리 학술대회 논문집, 2014.
[9] 이동주, 연종흠, 황인범, and 이상구. “꼬꼬마: 관계형 데이터베이스를 활용한 세종 말뭉치 활용 도구.” 정보과학회논문지: 컴퓨팅의 실제 및 레터 16, no. 11, pp. 1046-1050, 2010.
[10] 트위터에서 만든 오픈소스 한국어 처리기, Github, twitter/twitter-korean-text, https://github.com/twitter/twitter-korean-text, 2016.
[11] Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., & Dean, J. Distributed representations of words and phrases and their compositionality. In Advances in neural information processing systems pp. 3111-3119, 2013.
[12] Kim, Yoon. "Convolutional neural networks for sentence classification." arXiv preprint arXiv:1408.5882, 2014.

한국어 대화문 화행 자동분류를 위한 언어학적 기반연구

구영은*○, 김지연*, 홍문표*, 김영길**
성균관대학교*, 한국전자통신연구원**
{sarah8835, kite92, skkhmp}@skku.edu, kimyk@etri.re.kr

A Linguistic Study of Automatic Speech Act Classification for Korean Dialog

Youngeun Koo*○, Jiyoun Kim*, Munpyo Hong*, Young-Kil Kim**
Sungkyunkwan University*, Electronics and Telecommunications Research Institute**

요 약

화행이란 의사소통 과정에서 발화자가 가지는 발화 의도를 말한다. 성공적인 의사소통을 위해서는 발화자의 화행을 정확하게 파악하는 것이 매우 중요하다. 본 논문에서는 한국어 대화체 문장의 화행 자동분류를 위해, 화행을 결정짓는 요인이 무엇인지 언어학적으로 분석하고자 하였다. 한국어 수업 대화를 분석하여 화행 분류 체계를 새롭게 자체 정립하였고, 언어학적 근거를 바탕으로 10개의 화행 분류 자질을 제안하였다. 또한 제안하는 화행 분류 자질을 검증하고자 웨카(Weka)를 이용하여 정확률 실험을 진행하였다.

주제어: 화행, 화행 자동분류, 화행 분류 자질, 기계 학습

1. 서론

화행은 “의사소통을 함으로써 화자가 드러내고자 하는 태도(the type of attitude being expressed)”를 의미한다 [2]. 화행은 발화자의 발화 의도를 뜻하며, 의사소통을 할 때 상대방의 발화 의도인 화행을 정확하게 파악하는 것은 매우 중요하다. 그러나 아래의 예문들과 같이 발화 문장의 표층 형태(surface form)만으로는 상대방의 발화 의도를 정확하게 분석할 수 없다.

(1) (집을 어지럽힌 아이에게) 도대체 왜 이렇게 집을 어지럽혔어?
(2) (창문 옆에 앉은 친구에게) 문이 열려있어서 그런지 되게 춥네.

예문 1은 의문문으로 된 문장으로 발화의 표층 형태로만 볼 때는 답변을 요구하는 질문인 듯 보인다. 그러나 예문 1은 아이의 대답을 듣기 위한 발화문이 아니고 집을 어지럽힌 아이를 꾸짖고자 하는 비난의 발화이다.
예문 2는 평서문으로 이루어진 문장이지만 단순히 정보를 제공하거나 상황을 묘사하기 위해 발화된 문장이 아니다. 실제로 예문 2는 창문을 닫아달라는 요청의 발화이다.
이처럼 발화의 화행은 표층 형태만으로는 정확하게 파악하기 어려운 경우가 많다. 여러 언어적·문맥적 정보를 활용해야 발화의 진짜 화행을 파악할 수 있고 성공적인 의사소통이 가능하다. 따라서 본 논문에서는 발화의 어떤 언어학적 특징으로 인해 그것의 화행이 결정되는지를 분석하고, 이를 이용하여 화행의 자동분류를 시도하고자 한다. 화행을 결정짓는 자질들에 대한 분석을 통해 화행의 정확한 분석이 가능하다면 인공지능 스피커 등 다양한 분야에 적용할 수 있을 것으로 기대된다.

본 논문은 다음과 같이 구성되어 있다. 먼저 2장에서는 화행이론이 무엇인지 간략히 살펴보고 화행분류에 관한 기존 연구들을 소개한다. 3장에서는 본 연구에서 제안하는 화행 분류 체계와 화행 분류를 위한 자질을 언어학적으로 분석한다. 4장에서는 3장에서 제안한 화행 분류 자질을 검증하기 위해 실험을 수행하고 그 결과를 분석한다. 끝으로 5장에서는 이 글을 정리하고 향후 연구방향을 제시한다.

2. 관련 연구

2.1. 화행이론

화행이론은 영국의 언어 철학자 오스틴(J. Austin)에 의해 창시되고 그의 제자 썰(J. Searl)에 의해 체계화된 이론이다. 기존의 언어학이 문장의 의미를 문장이 지니는 명제의 참과 거짓의 진리값(truth value)을 통해 언어 표현의 의미를 파악하려는 시도에서 그쳤다면, 화행이론에서는 발화에 담겨있는 의미와 더불어 결합되는 특정한 힘에 의해 이끌어지는 행위를 탐구한다 [1].
발화를 통해 이루어지는 행위는 크게 세 가지 종류가 있다. 첫 번째는 언표적 행위(locutionary act)이다. 언표적 행위는 발화자가 어떤 의미 있는 발화를 생성해내는 행위 자체를 의미한다. 이때 발화는 소리의 발화(phonetic), 어떤 문법적 형태의 소리들로 된 낱말들의 발화(phatic), 어떤 일정한 의미와 지시대상을 가진 발화(rhetic)로 다시 분류된다 [21]. 즉, 언표적 행위는 소리, 형태, 의미를 가지는 언어 표현의 ‘생성’을 의미한다고 할 수 있다. 두 번째 행위는 언표내적 행위(illocutionary act)이다. 언표내적 행위는 화자가 발화

를 통해 수행하고자 하는, 의도하는 행위이다. 다시 말해서 언표내적 행위는 언어 표현의 '전달'에 초점이 맞춰져있다. 보통 화행이론에서 말하는 좁은 의미의 화행은 이 언표내적 행위를 가리킨다 [7]. 세 번째 행위는 언표효과적 행위(perlocutionary act)이다. 언표효과적 행위는 어떤 발화의 수행을 통해 청자로부터 기대할 수 있는 효과로 언어 표현의 '인식'과 관련되어 있다.

2.2. 화행 분류

2.2.1. 오스틴과 썰 중심의 초기 화행 분류

Austin(1962)은 표 1과 같은 화행 분류를 제안한다. 발견한 내용을 전달하는 판정행위(Verdictives), 일련의 일에 대한 찬성 또는 반대의 영향력을 행사하는 행사행위(Exercitives), 일련의 일을 발화자가 감당할 것을 표현하는 언약행위(Commissives), 상대방의 행위, 운, 태도에 대한 발화자의 태도를 표현하는 행태행위(Behabitives), 견해의 해설, 논쟁, 명확한 설명을 하는 평서행위(Expositives)가 그것이다 [1].

표 1. Austin(1962)의 화행 분류 체계

화행 종류	화행 설명
Verdictives (판정 행위)	발견한 내용을 전달하는 행위
Exercitives (행사 행위)	일련의 일에 대한 찬성 또는 반대의 영향력 행사하는 행위
Commissives (언약 행위)	일련의 일을 발화자가 감당할 것을 표현하는 행위
Behabitives (행태 행위)	상대방의 행위, 운, 태도에 대한 발화자의 태도를 표현하는 행위
Expositives (평서 행위)	견해의 해설, 논쟁, 명확한 설명을 하는 행위

Searle(1976)은 표 2와 같이 화행을 화자가 사실이라고 믿거나 사실인 것으로 알고 있는 사태에 대해 말하는 단언 화행(Representative), 청자가 해주기를 원하는 행위를 화자가 언급하여 청자가 그 행위를 하게 하는 지시화행(Directives), 화자가 자신의 미래에 할 행위를 말하는 위임화행(Commissives), 화자의 심리적 태도를 표현하는 정표화행(Expressives), 행위를 통해 어떤 사태를 결정하거나 새로운 사태를 만드는 선언화행(Declaratives)으로 분류하였다 [15].

오스틴과 썰의 화행 분류는 이후 연구자들에게 많은 영향을 주었다. 그러나 화행간의 경계가 불분명하고, 화행 유형의 범위 설정에 오류가 존재하며, 복수 개의 화행으로 매칭될 가능성이 있다는 한계를 지적받는다. 그로 인해 이들의 화행 분류 체계에 대한 연구는 궁극적인 화행 분류이기 보다는 유사한 특징을 가지는 화행간의 결합일 뿐이라고 평가받기도 한다.

표 2. Searle(1976) 화행 분류 체계

화행 종류	화행 설명
Representatives (단언 화행)	화자가 사실이라고 믿거나 사실인 것으로 알고 있는 사태에 대해 말하는 것
Directives (지시 화행)	청자가 해주기를 원하는 행위를 화자가 언급하여 청자가 그 행위를 하게 하는 것
Commissives (위임 화행)	화자가 자신이 미래에 할 행위를 말하는 것
Expressives (정표 화행)	화자의 심리적 태도를 표현하는 것
Declaratives (선언 화행)	행위를 통해 어떤 사태를 결정 또는 새로운 사태를 만드는 것

이후 Austin(1962)과 Searle(1976)의 분류 체계의 한계를 보완하고자 다양한 접근 방식이 등장한다.

Searle(1976)의 화행 유형 용어를 차용한 Bach&Harnish(1979)는 "인간은 단순히 소리를 내기 위해 발화하는 것이 아니고, 반드시 어떠한 이유가 존재하기 때문에 발화한다"는 통보적 가정(Communicative presumption)을 바탕으로 화행을 분류하고자 하였다 [25]. 먼저 발화의 이유를 기준으로 통보적 측면의 발화와 그 외의 관습적인 상황과 연관된 발화로 분류하였다. 그 다음 통보적 측면의 발화, 즉 통보적 언표내적 행위를 약속, 제공 화행이 포함되는 언약화행(Commissives), 금지, 요구, 요청 등의 화행이 포함되는 지시화행(Directives), 가정, 귀속, 기술, 단언 등의 화행이 포함되는 진술화행(Constatives), 사과, 위로, 수락, 축하 등의 화행이 포함되는 인사화행(Acknowledgements)으로 분류하였다. 그리고 관습적 측면의 발화, 즉 관습적 언표내적 행위는 유효화행(Effectives)과 판정화행(Verdictives)으로 분류하였다 [25].

국내에서는 박영수(1981)가 수행동사를 법행위 동사, 종교 행위 동사, 사무 행위 동사가 포함되는 의식적 수행동사와 단정동사, 평가동사가 해당되는 통속적 수행동사로 분류하여 한국어 화행 분류를 시도하였다 [20].

이 외에도 장석진(1987)은 화행이 발생하는 상황을 고려하는 기존의 화행 분류 방식과 달리 화행이 발생하는 발화문의 통사적·의미적 구조를 분석하였다. 이를 바탕으로 문장을 13종의 통사 유형, 11종의 의미 유형으로 분류하였다. 이렇게 얻어진 통사·의미적 구조를 바탕으로 정립된 통사유형-의미유형 쌍과 Austin(1962), Searle(1976), Bach&Harnish(1979)의 화행 분류 체계를 활용하여 설정한 화행 유형을 연결시키는 방식으로 연구가 진행되었다 [26].

2.2.2. 기계학습을 이용한 화행 분류

최근에는 기계학습을 이용하여 자동으로 화행을 분류하는 연구들이 진행되고 있다. 기계학습을 위해서는 발화문이 가지는 화행과 화행 분류를 위한 자질 정보가 부착된 실험코퍼스가 필요하다. 화행 유형은 연구자들마다 다르게 설정하지만 크게 세 가지로 나눠 볼 수 있다.

먼저 썰의 화행 유형을 사용하는 연구 방법이 있다. Marineau et al.(2000) [9], Qadir et al.(2011) [12]과 같은 연구에서는 Searle(1976)이 사용한 5가지 화행 유형을 그대로 사용하여 화행 자동 분류를 시도하였다. 다음으로 DAMSL dialogue act tagset을 사용하는 경우이다. Grau et al.(2004) [5], 김민정 외(2006) [17] 등의 연구에서는 DAMSL tagset을 이용하여 실험을 진행하였다. 마지막으로 연구자들이 자체적으로 화행 분류 체계를 정립하는 경우가 있다. 자동 화행 분류의 정확도를 높이기 위하여 코퍼스를 분석한 뒤 이에 적합한 화행 분류 체계를 정립하여 연구를 진행하기도 한다.

화행이 부착된 실험 코퍼스로 화행을 자동 분류할 때 다양한 기계학습 알고리즘이 사용될 수 있다. 여러 가지 알고리즘이 존재하지만 특히 'Naïve Bayes Classifier (네이브 베이지언)'과 'Support Vector Machine(지지 벡터 기계)' 알고리즘이 가장 광범위하게 사용된다. 전자의 경우 Grau et al.(2004) [5], Moldovan et al.(2011) [10], Rasor et al.(2011) [13], Samei et al.(2014) [14] 등에서 사용되었고, 후자의 경우 은종민 외(2005) [22], 김세종 외(2008) [19], Qadir et al.(2011) [12] 등의 연구에서 활용되었다.

3. 화행 자동분류

3.1. 화행 분류 체계

본 연구에서는 한국어 화행 자동분류에 관한 기존 연구들에서 널리 사용하고 있는 이재원(1999) [24]을 분석함을 통해 화행 분류 체계를 자체 정립하고자 하였다. 호텔·항공·여행 예약 도메인을 분석한 이재원(1999) [24]에서 중복 혹은 누락된 화행 유형을 보완하고자 DAMSL tagset [4]의 유형을 활용하였다. 예를 들어, 이재원(1999)의 'Introducing oneself' 화행은 대화 중에 자신의 신분을 밝히는 호텔·항공·여행 예약 도메인에서만 특수적으로 발견된다. 일반 대화에서는 그 중요도가 떨어지므로 본 연구에서 제안하는 화행 분류 체계에서는 제외하고 'Greeting(인사)' 화행에 포함시켰다. 또한 이재원(1999) [24]에서 누락된 '대답회피'를 본 연구에서는 포함시켜 화행 분류 체계를 정립했다.

뿐만 아니라 한국어 대화 코퍼스를 분석하여 해당 코퍼스에서 발견된 화행들을 이용하여 아래와 같은 화행 분류 체계를 자체 정립하였다. 예를 들어, 실제 대화 코퍼스를 분석해보면 '주위 환기'와 같이 발화하는 이유는 존재하지만 문장 자체가 가지는 내용적 의미는 거의 없는 경우가 있다. 본 연구에서는 이를 위해 '감탄' 화행을 추가하였다.

표 3. 본 연구에서 제안하는 화행 분류 체계

번호	화행		번호	화행	
1	Accept	수락	14	Express	의지 표현
2	Acknowledge	호응	15	Greeting	인사
3	Answer	답변	16	Induce	유도
4	Apologize	사과	17	Inform	정보 제공
5	Ask-confirm	확인 요구	18	Maybe	추측
6	Ask-if	Y/N 질문	19	Praise	칭찬
7	Ask-ref	WH 질문	20	Promise	약속
8	Assert	주장	21	Reject	거절
9	Avoid	대답 회피	22	Request	요청
10	Call	부름	23	Response	반응
11	Correct	수정	24	Suggest	제안
12	Criticism	비난	25	Thanking	감사
13	Exclamation	감탄			

3.2. 화행 분류 자질

본 논문에서는 발화문의 화행을 결정짓는 자질을 언어학적으로 분석하고자 하였다. 제안하는 화행 분류 자질은 크게 문장 자질과 문맥 자질로 나뉜다. 문장 자질은 통사적 또는 어휘·의미적 특징과 같이 문장 자체에 나타나는 자질이며, 문맥 자질은 주변 발화 그리고 그것의 화행과 관련된 자질이다. 다음은 본 논문에서 제안하는 화행 분류 자질을 정리한 표이다.

표 4. 본 연구에서 제안하는 화행 분류 자질

번호	자질 이름		자질 값
1	문장 자질	문장의 유형	decl, yn_ques, wh_ques, ques, impe, excl
2		시제	1, 2, 3
3		주어의 인칭	1, 2, 3
4		부정형 포함 여부	0, 1
5		의문사 포함 여부	0, 1
6		본동사의 개수	0, 1, 2
7		문장의 길이	1, 2, 3
8	문맥 자질	바로 이전 화행	1~25
9		상대방의 바로 이전 화행	1~25
10		발화 차례의	y, n

		변화 여부	

a. 문장의 유형

문장의 유형에 의해 발화의 화행을 예측할 수 있다. 문장의 유형과 화행간의 관계에 관한 연구는 이성욱 외(1999) [23], 은종민 외(2005) [22]등에서 진행된 바 있다. 예를 들어 평서문은 특정한 정보를 전달하고, 의문문은 화자가 질문하는 것을 가능하게 한다. 문장의 유형은 화행을 결정하는 중요한 자질 중 하나이므로 본 연구에서는 문장의 유형을 평서문, 예/아니오-의문문, wh-의문문, 일반 의문문, 명령문, 감탄문으로 분류하여 문장의 유형과 화행의 상관관계를 살펴보았다.

b. 시제

발화문의 시제와 화행간의 관련성 또한 화행 분류를 위한 자질 중 하나이다. 이에 관한 연구는 이성욱 외(1999) [23], 은종민 외(2005) [22] 등에서 있었다. 제안, 약속, 요청 등과 같은 화행의 경우 현재나 미래 시제와 관련된다. 이런 화행은 대부분 현재나 미래에 발생하는 상황과 연관이 깊기 때문이다. 그러나 비난과 같은 화행은 이미 발생한 상황 또는 사건에 대한 언급이 필요하기 때문에 과거나 현재 시제와 관련성이 높다.

c. 주어의 인칭

주어의 인칭 역시 화행과 관련성이 있다. 주어가 1인칭일 경우 대다수의 문장이 발화자 자신에 대한 정보 또는 발화자를 포함한 모든 대화 참여자들에 관한 정보를 담고 있다. 따라서 이러한 발화의 경우 주장, 약속, 제안과 같은 화행이 빈번히 나타난다. 2인칭의 주어를 가진 발화는 대화에 참여하고 있는 청자에게 특정 행동을 요구하는 제안, 약속, 질문 등의 화행이 자주 발견된다. 3인칭 주어인 발화는 보통 객관적인 정보를 전달하는 경우가 많아서 정보제공의 화행이 등장할 가능성이 높다.

d. 부정형 포함 여부

부정형의 표현이 포함된 발화는 다수의 화행과 함께 등장한다. 그러나 감사, 감탄 또는 요청에 대한 수락을 나타내는 화행에서는 이러한 표현을 찾아보기 힘들다.

e. 의문사 포함 여부

의문사가 포함된 표현은 질문 화행에서 아주 자주 사용된다. 추측 화행 또한 마찬가지로 의문사와 함께 등장하는 경우가 빈번하게 발생한다. 또한 요청, 제안과 같은 화행도 문장 내에 의문사를 자주 포함하는데 청자에게 특정한 행동을 요청하는 경우 질문의 형태로 발화하기 때문이다. 그러나 정보제공, 거절, 주장과 같은 화행에서는 의문사가 포함된 표현을 찾기 힘들다.

f. 본동사의 개수

이성욱 외(1999) [23], 은종민 외(2005) [22], Qadir et al.(2011) [12]등의 연구에서 언급된 바와 같이 본용언은 상황을 표현하는데 가장 중요한 역할을 한다. 따라서 본동사의 개수 또한 화행을 결정하는데 영향을 끼치는 하나의 자질이 된다. 예를 들어, 감탄, 인사, 호응, 반응과 같은 화행에서는 본동사가 드물게 등장하는 반면, 제안, 약속, 요청과 같은 화행은 한 개 이상의 본동사가 필요하다. 또한 정보제공과 같은 화행에서는 한 번의 발화에 2개 이상의 본동사가 등장하기도 한다.

g. 문장의 길이

문장의 길이 또한 화행을 결정하는 자질 중 하나이다. 문장의 길이란 발화문에 등장하는 어절의 개수를 의미한다. 감탄, 인사, 확인요구와 같은 화행은 보통 짧은 문장에서 나타난다. 그러나 정보제공, 답변, 주장과 같은 화행의 경우 발화 과정에서 문장의 길이가 길어지는 경우가 빈번하게 발생한다. 본 연구에서는 문장의 길이를 어절의 개수가 5개 이하인 것, 6개 이상 15개 이하인 것, 16개 이상 인 것, 세 가지 종류로 분류하였다.

h. 바로 이전 화행

이전 발화의 화행에 대한 연구가 김민정 외(2008) [18], Samei et al.(2014) [14], Bayat et al.(2016) [3]와 같은 연구에서는 진행되었다. 대화는 분리되고 독립적인 문장들의 모음이 아니라 각 문장이 서로 연관된 연장선상에 놓여있다. 따라서 이전 화행 또한 화행을 분류하여 결정하는데 일정한 역할을 한다.

i. 상대방의 바로 이전 화행

그라이스(P. Grice)가 주장한 협력원리(cooperative principle)에 의하면 대화 참여자는 상대방과 긴밀한 관계를 가지는 내용을 발화한다 [6]. 서로 관계있는 대화를 한다는 것은 대화에 일정한 패턴이 자주 등장함을 의미한다. 예를 들어, 상대방이 질문하면 그 다음 발화자는 질문에 대한 답변 또는 반문한다. 요청의 발화가 등장하면 그 후에는 요청에 대한 수락 또는 거절 등이 나온다. 따라서 단순히 발화의 진행성을 고려한 '바로 이전 화행' 뿐만 아니라, 대화의 연결성을 고려한 '상대방의 바로 이전 화행'도 중요한 자질 중 하나이다.

j. 발화 차례의 변화 여부

화행을 결정하는 데는 발화자 정보 또한 중요한 요소이다. 발화자가 바뀌면서 이전 화행과 상대방의 이전 화행과 관련된 새로운 화행이 등장할 가능성이 높다.

4. 실험

4.1. 실험 코퍼스

본 연구에서 제안하는 화행 분류 자질을 검증하기 위해 한국어 대화 코퍼스를 이용하여 실험을 진행하였다. 본 실험에서는 국립국어원(NIKL, National Institute of the Korean Language)의 수업 대화 코퍼스를 사용하였다. 이는 총 1,835개의 발화로 구성되어 있다. 해당 코퍼스는 선생님과 학생 간의 일대일 대화인데, 수업 대화뿐만 아니라 일상 대화도 일부 포함되어 있다.

수집된 코퍼스에는 3장에서 제안한 화행 유형과 화행 분류 자질의 값을 직접 부착하여 최종 실험코퍼스를 구

축하였다. 아래는 화행 유형별 발화문 개수를 정리한 표이다.

표 5. 화행 유형별 코퍼스의 발화문 개수

번호	화행		발화	번호	화행		발화
1	Accept	수락	1	14	Express	의지 표현	17
2	Acknowledge	호응	187	15	Greeting	인사	0
3	Answer	답변	270	16	Induce	유도	50
4	Apologize	사과	1	17	Inform	정보 제공	423
5	Ask-confirm	확인 요구	91	18	Maybe	추측	47
6	Ask-if	Y/N 질문	115	19	Praise	칭찬	1
7	Ask-ref	WH 질문	158	20	Promise	약속	5
8	Assert	주장	139	21	Reject	거절	2
9	Avoid	대답 회피	12	22	Request	요청	78
10	Call	부름	4	23	Response	반응	138
11	Correct	수정	28	24	Suggest	제안	29
12	Criticism	비난	24	25	Thanking	감사	0
13	Exclamation	감탄	27				

실험은 자바(Java) 기반의 기계학습 알고리즘 툴인 웨카(Weka) 3.8.1 버전을 사용하였다. 그리고 기계학습 알고리즘으로는 'Support Vector Machine(SVM)'을 사용하였으며, 성능 평가는 '10-fold cross validation' 방식을 이용하여 실험 결과를 얻었다.

4.2. 분석

표 6은 각 자질들의 사용 여부에 따른 화행 자동분류 성능을 정리한 것이다.

표 6. 자질에 따른 정확률 비교

실험		정확률(%)
베이스라인	유니그램(Unigram)	40.27%
실험1	문장 자질	55.48%
실험2	문맥 자질	49.05%
실험3	문장 자질 + 문맥 자질	59.56%

본 실험의 목표는 발화문의 화행을 결정짓는 자질이 무엇인지 분석하는 것이기 때문에, 별도의 자질을 사용하지 않고 오직 코퍼스에 등장하는 유니그램만을 이용한 경우를 본 실험의 베이스라인으로 설정하였다. 이때 화행 분류 정확률은 40.27%이었다. 본 연구에서 제안하는 문장 자질과 문맥 자질을 모두 사용하여 화행을 분류한 결과 59.56%의 정확률을 보이며, 베이스라인 대비 약 20%의 성능 향상을 보였다.

실험1과 실험2의 결과를 비교해볼 때 문맥 자질만을 사용했을 때의 성능이 문장 자질만을 사용했을 때보다 약 7% 가량 더 낮은 것을 알 수 있다. 이 결과가 단순히 문맥 자질에 해당되는 자질의 개수가 문장 자질에 해당되는 자질의 개수보다 적기 때문이라고 보기는 어렵다. 실제로 문맥 자질은 화행 자동분류에 상당한 영향을 끼치는 자질임을 아래의 표를 통해 확인할 수 있다.

표 7. 영향력이 큰 상위 5개 자질

순위	자질	
1	문장 자질	문장의 유형
2	문맥 자질	상대방의 바로 이전 화행
3	문맥 자질	바로 이전 화행
4	문장 자질	시제
5	문장 자질	본동사의 개수

표 7은 웨카의 'InfoGainAttribute Evaluator' 기능을 통해 확인한 가장 높은 영향력을 지니는 다섯 개의 자질들을 정리한 것이다. 표 7을 통해 문맥 자질이 화행에 많은 영향을 끼치는 것을 알 수 있다. 특히 상대방의 바로 이전 화행은 문맥 자질 중에서 가장 중요한 화행 자동분류의 잣대임을 확인할 수 있다. 이는 본 실험에서 사용한 실험코퍼스가 일대일 대화이기 때문에 선행 발화와 후행 발화간의 관계가 매우 긴밀하기 때문이다.

대화에서는 빈번히 발생하는 인접 화행 쌍이 존재한다. 예를 들어, 선행 발화의 발화자가 질문을 하면 후행 발화의 발화자는 답변을 하거나 대답을 회피한다. 만일 선행 발화의 발화자가 무엇을 요청하면 후행 발화의 발화자는 그것을 수락, 거절 또는 반문한다. 이처럼 대화에는 인접 화행 쌍이 존재하며, 일대일 대화를 실험코퍼스로 갖는 본 실험과 같은 경우 그 중요도는 더 높다.

표 7에 의하면 본 연구에서 제안하는 자질 중에서 가장 영향력이 높은 것은 문장의 유형이다. 이는 많은 경우 의문문, 평서문, 감탄문 등의 문장 유형 자체가 이미 발화의 의도를 담고 있기 때문이다. 그러나 정확한 정보 전달을 위해 비교적 정형화된 표현을 발화하는 수업, 예약, 주문 등의 주제 대화와 달리 간접 화행이 빈번히 발생하는 일상 대화 코퍼스를 분석한다면 문장의 유형은 영향력이 떨어질 가능성이 있을 것으로 예측된다.

5. 결론

본 논문에서는 화행 분류 체계를 자체 정립하고 언어학적 분석을 통해 10개의 화행 분류 자질을 제안하였다. 또한 제안하는 화행 분류 자질을 기계학습을 통해 정확률을 검증하고 화행 자동분류를 시도하였다.

성공적인 의사소통을 위해서는 발화자의 발화 의도를 정확하게 파악하는 것이 매우 중요하다. 이는 일상생활뿐만 아니라 대화시스템과 같은 여러 다른 분야에서도 발화의 화행은 대화를 정확하게 분석하는 데에 중요한 요소이다. 본 연구에서는 한국어 코퍼스 분석을 통해 7개의 문장 자질과 3개의 문맥 자질을 제안하였다. 제안

하는 모든 자질을 사용한 경우 59.56%의 정확률을 보였으며 베이스라인 대비 20%의 성능 향상을 보였다.

화행 분류 자질에 대한 분석을 정교화하고 정확률을 향상시키기 위해서는 각 자질별 화행 분류 성능을 분석하고 보완해야 한다. 본 연구에서 제안한 7개의 문장 자질들과 3개의 문맥 자질들이 각각의 상위 자질의 성능 향상과 얼마나 밀접한 관계를 가지는지 통계적으로 분석할 필요가 있다. 예를 들어 주어의 인칭은 화행 분류에 미친 영향력이 다소 낮았다. 이는 한국어가 문장에서 대명사가 빈번히 생략되는 'pro-drop language' 이기 때문이다[11]. 주어를 복원한 후 주어의 인칭을 자질로 사용한다면 자질에 대한 정확한 분석이 가능해지고 성능 역시 향상될 수 있을 것이다.

향후 연구에서는 한국어 특유의 화행 분류 자질에 대한 연구를 진행할 계획이다. 한국어 코퍼스를 이용하여 수행동사(performative verb)나 수행부사(performative adverb) [8]에 대해 연구할 것이다. 또한 한국어 문법을 분석하여 한국어 화행 표지[15] 또는 용언의 어미와 같이 한국어에서 특수하게 나타나는 자질을 정립하여 화행 분류 자질을 정교화할 예정이다. 코퍼스 크기를 더 확장할 뿐만 아니라 다른 도메인에 본 연구의 화행 분류 자질을 적용하여 그 적용범위를 파악하고, 범도메인적 화행 분류 자질을 정립하기 위한 연구를 진행할 예정이다.

감사의 글

이 논문은 2017년도 정부(과학기술정보통신부)의 재원으로 정보통신기술진흥센터의 지원을 받아 수행된 연구임(R7119-16-1001, 지식증강형 실시간 동시통역 원천기술 개발)

참고문헌

[1] Austin, How to Do Things with Words, The William James Lectures Delivered at Harvard University in 1955, Clarendon Press, 1962.

[2] Bach, Conversational implicature. Mind & Language, 9(2), 124-162, 1994.

[3] Bayat et al., Supervised Speech Act Classification of Messages in German Online Discussions, In FLAIRS Conference, 204-209, 2016.

[4] Core et al., Coding dialogs with the DAMSL annotation scheme, In AAAI fall symposium on communicative action in humans and machines (Vol. 56), 1997.

[5] Grau et al., Dialogue act classification using a Bayesian approach, In 9th Conference Speech and Computer, 2004.

[6] Grice, Logic and Conversation, Syntax and semantics 3: Speech arts, Cole et al., 41-58, 1975.

[7] Huang, Pragmatics, Oxford: Oxford University Press, 2007.

[8] Levinson, Pragmatics, Cambridge University Press, 1983.

[9] Marineau et al., Classification of speech acts in tutorial dialog. In Proceedings of the Workshop on Modeling Human Teaching Tactics and Strategies of ITS 2000, 65-71, 2000.

[10] Moldovan et al., Automated Speech Act Classification For Online Chat. MAICS, 710, 23-29, 2011.

[11] Park et al., Zero Object Resolution in Korean. In PACLIC, 2015.

[12] Qadir et al., Classifying sentences as speech acts in message board posts, In Proceedings of the Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics, 748-758, 2011.

[13] Rasor et al., Student Speech Act Classification Using Machine Learning. In FLAIRS Conference, 2011.

[14] Samei et al., Context-based speech act classification in intelligent tutoring systems, In International Conference on Intelligent Tutoring Systems, Springer International Publishing, 236-241, 2014.

[15] Searle, A classification of illocutionary acts. Language in society, 5(01), 1-23, 1976.

[16] 김기찬, 한국어의 화행 표지. 언어과학연구, 10, 47-69, 1993.

[17] 김민정 외, 도메인에 비종속적인 대화에서의 화행 분류, 한국정보과학회 언어공학연구회, 한국정보과학회 언어공학연구회 학술발표 논문집, 10, 246-253, 2006.

[18] 김민정 외, 한국어 화행 분류를 위한 최적의 자질 인식 및 조합의 비교 연구, 정보과학회논문지: 소프트웨어 및 응용, 35(11), 681-691, 2008.

[19] 김세종 외, 이전 문장 자질과 다음 발화의 후보 화행을 이용한 한국어 화행 분석, 정보과학회논문지: 소프트웨어 및 응용, 35(6), 374-385, 2008.

[20] 박영수, 비표현 수행력 연구, 대구: 형설출판사, 1981.

[21] 박주현, 화행 이론에 대하여 - Austin, Searle, Grice를 중심으로, 한국영어영문학회, 영문영문학, 29(2), 515-536, 1983.

[22] 은종민 외, 지지벡터기계 (Support Vector Machines) 를 이용한 한국어 화행분석, 정보처리학회논문지 B, 12(3), 365-368, 2005.

[23] 이성욱 외, 결정트리를 이용한 한국어 화행 분석, 한국정보과학회 언어공학연구회, 한국정보과학회 언어공학연구회 학술발표 논문집, 10, 377-381, 1999.

[24] 이재원, 통계적 화행처리를 이용한 대화체 기계번역에서의 효율적인 대화분석, 박사학위논문, 한국과학기술원, 1999.

[25] 이준희, 언표내적 화행의 유형과 간접 화행. 우리어문연구, 24(단일호), 69-99, 2005.

[26] 장석진, 한국어 화행동사의 분석과 분류, 서울대학교 언어교육원, 어학연구, 23(3), 307-339, 1987.

● 구두발표 2: 대화/질의응답 1

복사 방법 및 검색 방법을 이용한 종단형 생성 기반 질의응답 채팅 시스템

김시형○, 김학수, 권오욱*, 김영길*
강원대학교 컴퓨터정보통신공학과, 한국전자통신연구원*
{sureear,nlpdrkim}@kangwon.ac.kr, {ohwoog*,kimyk*}@etri.re.kr

End-to-End Generative Question-Answering Chat System Using Copying and Retrieving Mechanisms

Sihyung Kim○, HarkSoo Kim, Oh-Woog Kwon*, Young-Gil Kim*
Kangwon National University Computer and Communication Engineering
Electronics and Telecommunications Research Institute*

요 약

채팅 시스템은 기계와 사람이 서로 의사소통 하는 시스템이다. 의사소통 과정에서 질문을 하고 질문에 대한 답변을 하는 질의응답 형태의 의사소통이 상당히 많다. 그러나 기존 생성 기반 채팅 시스템에서 자주 사용되는 Sequence-to-sequence모델은 질문에 대한 답변보다는 좀 더 일반적인 문장을 생성하는 경우가 대부분이다. 이러한 문제를 해결하기 위해 본 논문에서는 복사 방법과 검색 방법을 이용한 생성 기반 질의응답 채팅 시스템을 제안한다. 템플릿 기반으로 구축한 데이터를 통한 실험에서 제안 시스템은 복사 방법만 이용한 질의응답 시스템 보다 45.6% 높은 정확도를 보였다.

주제어: 복사 방법, 검색 방법, 생성 기반 질의응답 채팅 시스템

1. 서론

채팅 시스템(Chatting System)은 기계가 사람의 말을 적절하게 이해하여 답변을 하는 시스템이다. 사람의 대화에서는 질문을 하고 질문에 대한 답변을 하는 질의응답 형태의 대화가 상당히 많다. 예를 들어 “버락 오바마”에 대한 대화 중, “그런데 버락 오바마가 어디 출신이지?”라는 질문이 등장 할 수 있다. 이에 대한 정답은 “호놀룰루”로, 사람은 “버락 오바마는 호놀룰루 출신이야”와 같이 자연스러운 문장을 만들어 답변한다. 이 과정을 시스템이 수행하기 위해서는 먼저 질문에 해당하는 주어(Subject)와 술어(Predicate)를 찾아 목적어(Object)를 찾아야 한다. 그 이후 찾은 목적어를 정해진 형식의 템플릿(Template)에 맞추어 문장으로 변환하여 답변한다. 그러나 이 방법은 정해진 답변밖에 생성할 수 없는 문제점이 있다. 본 논문에서는 이러한 문제를 해결하기 위하여 질의에 대한 정답만 출력 하는 것이 아니라 답변에 주어의 다른 목적어 정보를 추가하여 보다 자연스러운 문장을 생성하는 종단형(End-to-end) 생성 기반 질의응답 시스템을 제안한다.

2. 관련 연구

두 개의 Recurrent Neural Network(RNN)을 사용하는 Sequence-to-sequence모델[1]은 입력 열을 인코딩 하여 출력 열을 디코딩 하는 모델로써 다양한 자연어 처리 연구에 활용 되고 있다. 번역 분야에서 주의 집중(attention)을 기반으로 한 Encoder-Decoder 모델은 디코더가 인코더에서 좀 더 중요한 정보를 선택할 수 있게 함으로써 디코더의 성능을 향상시켰으며[2], 생성 채팅 모델에서도 주로 사용되고 있다[3]. 복사 방법(Copying mechanism)은 디코더를 변형한 모델로써 기존의 주의 집중 방법에 더하여 입력을 복사하여 디코더의 출력으로 활용하는 연구가 진행 되었다[4]. 본 논문과 가장 연관이 깊은 [5]는 검색 방법(Retrieving mechanism)을 통한 생성 채팅 모델로써 복사 방법에 지식베이스를 검색하여 목적어를 찾아내고, 복사 방법과 유사하게 목적어를 디코더의 출력으로 활용하였다. 본 논문에서는 Decoder의 출력에 복사 방법과 검색 방법을 적용하는 종단형 생성 기반 질의응답 채팅 시스템을 제안한다.

3. 생성 채팅 모델

3.1 복사 방법을 활용한 생성 채팅 모델

그림 1은 복사 방법을 적용한 주의 집중 기반 Encoder-Decoder 모델이다. 복사 방법은 주의 집중 기반 Encoder-Decoder 모델에서 입력 어휘를 출력으로 복사되도록 하는 방법이다.

입력 열 $X=\{x_1,x_2,...,x_i\}$을 인코딩하기 위해 각각의 입력을 Bi-directional-LSTM[6]을 사용하여 다음과 같은 수식으로 인코딩 한다.

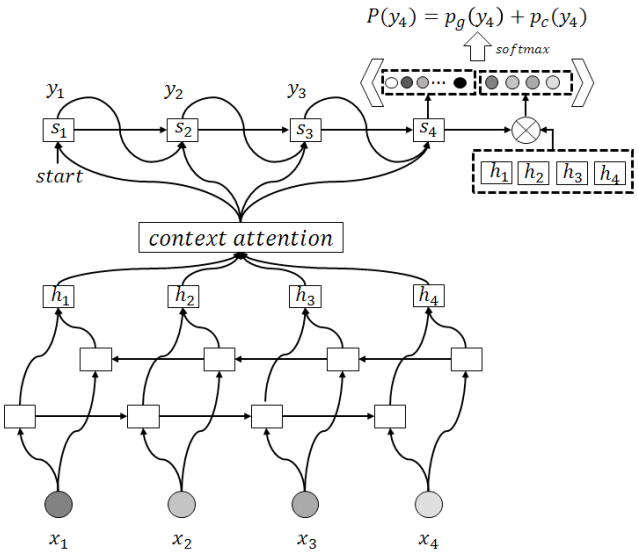


그림 1 복사 방법이 적용된 Encoder-Decoder 모델

$$
\begin{aligned}
h_{f,i} &= LSTM(x_i, h_{f,i-1}) \\
h_{b,i} &= LSTM(x_i, h_{b,i+1}) \\
h_i &= [h_{f,i}; h_{b,i}]
\end{aligned} \tag{1}
$$

식 1에서 $h_{f,i}$는 정방향의 은닉 계층이고, $b_{f,i}$는 역방향의 은닉 계층이며, [;]는 결합(concatenate)을 의미한다. 인코딩 한 은닉 계층들을 사용하여 다음 수식을 사용하여 고정된 차원의 문맥 벡터(context attention)를 생성한다.

$$
\begin{aligned}
e_{ij} &= f(s_{i-1}, h_j) \\
\alpha_{ij} &= \frac{\exp(e_{ij})}{\sum_{k=1}^{T_x} \exp(e_{ik})} \\
c_i &= \sum_{j=1}^{T_x} \alpha_{ij} h_j \\
s_i &= f(s_{i-1}, c_i)
\end{aligned} \tag{2}
$$

식 2에서 e_{ij}은 계산된 i번째 출력과 디코더 입력의 j번째 입력을 신경망에 적용한 값, α_{ij}는 e_{ij}를 확률로 표현한 주의 집중 가중치이다. T_x는 인코더의 길이, c_i는 주의 집중 가중치와 입력을 통해 생성된 문맥 벡터, s_i는 디코더의 은닉 계층, f는 비선형함수(Nonlinear function)이다. 복사 방법의 디코더의 출력은 다음과 같이 정의된다.

$$
\begin{aligned}
&P(y_i | s_i, c_i, y_{i-1}, h_{T_x}) = \\
&p_g(y_i | s_i, c_i, y_{i-1}, h_{T_x}) + p_c(y_i | s_i, c_i, y_{i-1}, h_{T_x})
\end{aligned} \tag{3}
$$

식 3에서 p_g는 생성 모드(Mode)의 확률을 의미하고, p_c는 복사 모드의 확률을 의미하고 h_{T_x}는 인코더의 은닉 계층을 의미한다. 두 확률은 각각 다음 수식과 같이 계산된다.

$$
\begin{aligned}
p_g(y_i | s_i, c_i, y_{i-1}, h_{T_x}) &= \frac{1}{Z} exp(\psi_g(y_i)), \\
& y_i \in V \\
p_c(y_i | s_i, c_i, y_{i-1}, h_{T_x}) &= \frac{1}{Z} \sum_{j: x_j = y_i} \exp(\psi_c(x_j)), \\
& y_i \in X, y_i \notin V
\end{aligned} \tag{4}
$$

식 4에서 V는 디코더에서 출력 될 수 있는 어휘 사전이고, Z는 생성 모드와 복사 모드가 공유하는 정규화 항(Normalization term) 이다. ψ_g는 생성 모드의 점수(Score)이고, ψ_c는 복사 모드의 점수이다. ψ_g와 ψ_c는 다음과 같이 계산된다.

$$
\begin{aligned}
\psi_g(y_i = v_k) &= \nu_k^T W_g * s_i + b_g \\
\psi_c(y_i = x_j) &= \sigma(h_j^T * W_c) s_i
\end{aligned} \tag{5}
$$

식 5에서 ν_k는 V에서 v_k의 벡터를 가리키는 One-hot이고, W_g는 생성 모드가 사용하는 가중치(Weight) 행렬이다. h_j는 입력 x_j의 은닉 계층이고, W_c는 복사 모드가 사용하는 가중치이고, σ는 비선형함수로 본 논문에서는 $Tanh$를 사용한다. 입력 어휘가 출력 어휘 사전에 같은 단어로 존재하면 생성 점수와 복사 점수를 합산하여 계산한다.

3.2 검색 방법을 활용한 생성 채팅 모델

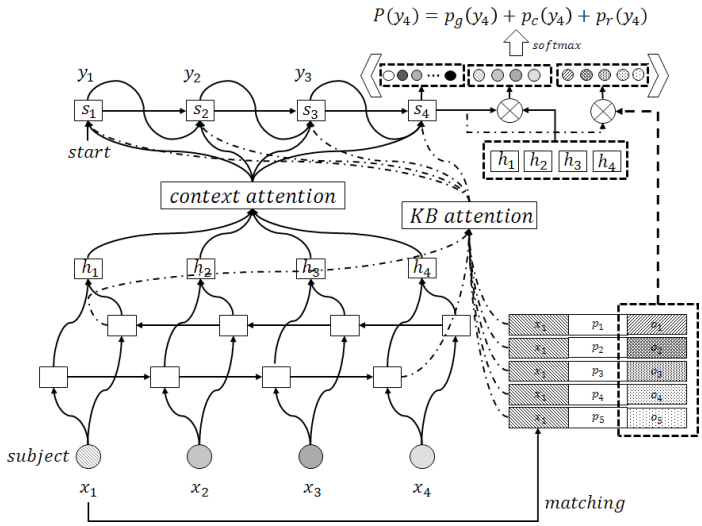


그림 2 복사 및 검색 방법이 적용된 Encoder-Decoder 모델

그림 2은 복사 방법 모델에 검색 방법을 적용한 모델이다. 검색 방법은 주어에 해당하는 술어와 목적어의 후보들을 검색하고 지식베이스 벡터(KB attention)를 생성한 이후 복사 방법과 비슷한 방법으로 목적어를 출력으

로 사용하는 방법이다.

검색 방법은 3.1절의 복사 방법의 입력 열 X에서 주어를 찾는다. 찾은 주어에 해당하는 술어 $P=\{p_1,p_2,...,p_l\}$와 목적어 $O=\{o_1,o_2,...,o_l\}$를 주어 x_s과 결합하여 트리플 임베딩 $k_s=[x_m;p_n;o_n]$를 생성한다. $k_s, h_{b,1}, h_{f,i}$를 사용하여 3.1절의 문맥 벡터 생성 방법과 같은 방법으로 지식베이스 벡터 kb_i를 생성한다. 검색 방법의 디코더의 출력은 다음과 같이 정의 된다.

$$\begin{aligned} &P(y_i|s_i,c_i,y_{i-1},h_{T_x},kb_i)= \\ &p_g(y_i|s_i,c_i,y_{i-1},h_{T_x},kb_i)+ \\ &p_c(y_i|s_i,c_i,y_{i-1},h_{T_x},kb_i)+ \\ &p_r(y_i|s_i,c_i,y_{i-1},h_{T_x},kb_i) \end{aligned} \quad (6)$$

식 6에서 p_r은 검색 모드의 확률을 의미한다. 세 확률은 각각 다음 수식과 같이 계산된다.

$$\begin{aligned} p_g(y_i|s_i,c_i,y_{i-1},h_{T_x},kb_i) &= \frac{1}{Z}exp(\psi_g(y_i)), \\ y_i &\in V \end{aligned}$$

$$\begin{aligned} p_c(y_i|s_i,c_i,y_{i-1},h_{T_x},kb_i) &= \frac{1}{Z}\sum_{j:x_j=y_i}\exp(\psi_c(x_j)), \\ y_i &\in X, y_i \notin V, y_i \notin O \end{aligned} \quad (7)$$

$$\begin{aligned} p_r(y_i|s_i,c_i,y_{i-1},h_{T_x},kb_i) &= \frac{1}{Z}\sum_{j:o_v=y_i}\exp(\psi_r(o_v)), \\ y_i &\notin X, y_i \notin V, y_i \in O \end{aligned}$$

식 7에서 Z는 생성 모드와 복사 모드와 검색 모드가 공유하는 정규화 항이고, ψ_r는 검색 모드의 점수이다. ψ_r는 다음과 같이 계산된다.

$$\psi_r(y_i=o_v)=\sigma(v_o^{T*}W_{kb})s_i \quad (8)$$

식 8에서 v_o는 V에서 o_v의 벡터를 가리키는 one-hot 벡터이고, W_{kb}는 검색 모드의 가중치이다. 목적어 어휘가 출력 어휘에 존재하면 검색 점수와 생성 점수를 합산하여 계산한다.

4. 실험 및 평가

4.1 실험 준비

본 논문에서는 복사 및 검색 방법을 실험하기 위해 [7]에서 사용한 템플릿 확장 방법을 사용하여 데이터를 구축하였다. 템플릿 확장 방법을 사용하기 위한 트리플 데이터는 DBPedia 2016-04[8]의 데이터 중 한국어 관계 트리플 정보(주어, 술어, 목적어)로 이루어진 데이터를 사용하였다. 표 1은 본 논문에서 사용한 Predicate의 종류와 개수이다. 확장 결과 총 1,050,575개의 데이터를 얻을 수 있었고 약 90%인 945,517개를 학습 데이터로 사용하고 나머지 데이터 105,058개를 실험 데이터로 사용하였다.

실험에 사용한 파라미터는 각 LSTM의 은닉 크기를 256, 단어의 차원 크기는 50, 배치 크기는 256으로 설정하였다. 검색 방법에서 검색을 하는 최대 길이는 5로 설정하고 주어는 문장에서 이미 찾았다는 가정하에 실험을 진행하였다. 실험의 입력과 출력 단위는 [7]에서 사용한 의사 형태소로 진행하였다. 주어와 목적어가 의사 형태소로 분리되는 것을 방지하기 위해 개체(Entity)를 하나의 단위로 묶어 사용 하였다.

표 1 Predicate의 종류와 개수

Predicate	데이터 개수
birthplace	25662
occupation	15508
nationality	12660
deathplace	4113
spouse	1674
developer	1448
place	1169
award	1003
parent	925
child	645
capital	533

4.2 실험 평가

실험 평가를 위해 예측한 문장에서 정답 목적어 존재 여부를 자동으로 체크하여 이에 대한 정확도(Accuracy)를 측정하였다.

표 2 복사 방법과 검색 방법의 성능

	Copying	Copying+Retrieving
정확도	37.8%	83.4%

표 2는 복사 방법 모델과 복사 방법에 검색 방법을 결합한 모델의 성능이다. 복사 방법 모델의 성능은 37.8%를 보여준 반면 복사 방법에 검색 방법을 결합한 모델의 성능은 83.4%로 복사 방법 모델보다 45.6% 높은 성능을 보여주었다. 이는 검색 방법이 검색된 목적어들의 후보를 바로 출력 디코더에 사용함으로써 예측 문장에 목적어가 등장할 확률을 상승시켰다고 판단된다.

4.3 결과 예시 및 분석

표 3은 복사 방법과 검색 방법을 결합한 모델의 실험 결과 예시이다. 주어는 굵은 글씨로, 목적어는 밑줄로 표현하였다. ID(1,4)는 정답 데이터와 일치된 예측 결과를 보여준다. ID(2)는 “생각”을 “만들”로 잘못 예측하였다. 이는 템플릿에 “생각 했어”와 “만들 었어”가 존재하는데, 가중치가 “만들”에 더 집중 된 것으로

표 3 복사 방법과 검색 방법을 결합한 모델의 결과 생성 예시

ID	Question	Gold	Predict
1	**요코미치 다카히로**가 태어난 곳은 ?	홋카이도에서 태어났습니다	홋카이도에서 태어났습니다
2	**모질라 애플리케이션 스위트**를 누가 만들었니 ?	모질라 재단이 생각 했어	모질라 재단이 만들 했어
3	**마르뱅 마르탱**은 어떤 나라 사람 인지 ?	**마르뱅 마르탱**은 France국적이지	**마르뱅 마르탱**는 France 국적이지
4	**어니스트 월턴**의 수상 내역 알려 줘	노벨 물리학상을 받았어~	노벨 물리학상을 받았어~
5	**왕다레이**의 국가는?	**왕다레이**는 차이나 국가를 가지고 있어	차이나는 차이나 국적를 가지고 있어
6	**교황 클레멘스 11세**가 죽은곳은?	이탈리아 입니다	이탈리아 에서
7	**오우라 가네다케**의 고향 은 어디인가요?	사쓰마 국에서 태어났습니다	사쓰마 국입니다 태어났습니다
8	**박병규**의 국가는?	**박병규**는 South Korea 국적이지	**박용택**는 South Korea 국적이지

보인다. 이와 비슷한 오류로 ID(7)이 있다. ID(3)은 자연스럽지 않은 조사가 예측되었는데, 이는 개체를 입력 단위로 분리하여 입력한 것이 아니라, 하나의 단위로 입력하였기 때문에 개체와 단어가 같은 공간에 매핑되지 않는 문제로 보인다. ID(5,8)은 복사 방법의 점수와 검색 방법의 점수가 상반되어 생긴 에러이다. ID(6)은 문장이 완전히 생성 되지 않은 문제로, 이는 디코더가 생성한 전체 문장을 고려하는 Search방법을 적용해야 할 것으로 보인다.

5. 결론

본 논문에서는 질의에 해당하는 목적어를 찾기 위한 방법으로 복사 방법과 검색 방법을 결합한 질의응답 채팅시스템을 제안하였다. 기존의 복사 방법에 검색 방법을 더하여 목적어가 출력에 더 잘 표현 될 수 있게 하였다. 향후 연구로 학습된 지식 임베딩을 직접 입력하는 연구를 진행할 예정이다.

감사의 글

이 논문은 2017년도 정부(과학기술정보통신부)의 재원으로 정보통신기술진흥센터의 지원을 받아 수행된 연구임. (R0126-15-1117, 언어학습을 위한 자유발화형 음성대화처리 원천기술 개발)

참고문헌

[1] O. Vinyals and Q. Le, A neural conversational model. *arXiv preprint arXiv:1506.05869*, 2015.

[2] D. Bahdanau, K. Cho and Y. Bengio, Neural machine translation by jointly learning to align and translate, *arXiv preprint arXiv:1409.0473*, 2014.

[3] J. Li, M. Galley, C. Brockett, J. Gao and B. Dolan, A diversity-promoting objective function for neural conversation models, *arXiv preprint arXiv:1510.03055*, 2015.

[4] J. Gu, Z. Lu, H. Li and V. O. Li, Incorporating copying mechanism in sequence-to-sequence learning, *arXiv preprint arXiv:1603.06393*, 2016.

[5] S. He, C. Liu, K. Liu and J. Zhao, Generating Natural Answers by Incorporating Copying and Retrieving Mechanisms in Sequence-to-Sequence Learning. *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics*, Vol. 1, pp. 199-208, 2017.

[6] M. Schuster and K. K. Paliwal, Bidirectional recurrent neural networks, *IEEE Transactions on Signal Processing,* 45.11, pp. 2673-2681, 1997.

[7] 김시형, 김학수, “의사 형태소 단위 채팅 시스템”, *제 28회 한글 및 한국어 정보처리 학술대회 논문집*, pp. 263-267, 2016.

[8] http://wiki.dbpedia.org/dbpedia-version-2016-04

한국어 대화 모델 학습을 위한 디노이징 응답 생성

김태형○, 노윤석, 박성배, 박세영
경북대학교
{thkim, ysnoh, sbpark}@sejong.knu.ac.kr, seyoung@knu.ac.kr

Denoising Response Generation for Learning Korean Conversational Model

Tae-Hyeong Kim○, Yunseok Noh, Seong-Bae Park, Se-Yeong Park
Kyungpook National University

요 약

챗봇 혹은 대화 시스템은 특정 질문이나 발화에 대해 적절한 응답을 해주는 시스템으로 자연어처리 분야에서 활발히 연구되고 있는 주제 중 하나이다. 최근에는 대화 모델 학습에 딥러닝 방식의 시퀀스-투-시퀀스 프레임워크가 많이 이용되고 있다. 하지만 해당 방식을 적용한 모델의 경우 학습 데이터에 나타나지 않은 다양한 형태의 질의문에 대해 응답을 잘 못해주는 문제가 있다. 이 논문에서는 이러한 문제점을 해결하기 위하여 디노이징 응답 생성 모델을 제안한다. 제안하는 방법은 다양한 형태의 노이즈가 임의로 가미된 질의문을 모델 학습 시에 경험시킴으로써 강건한 응답 생성이 가능한 모델을 얻을 수 있게 한다. 제안하는 방법의 우수성을 보이기 위해 9만 건의 질의-응답 쌍으로 구성된 한국어 대화 데이터에 대해 실험을 수행하였다. 실험 결과 제안하는 방법이 비교 모델에 비해 정량 평가인 ROUGE 점수와 사람이 직접 평가한 정성 평가 모두에서 더 우수한 결과를 보이는 것을 확인할 수 있었다.

주제어: 대화 모델, 시퀀스-투-시퀀스 모델, 자연어 생성

1. 서론

챗봇(chatbot) 혹은 대화 시스템은 자연어 질의에 대해 적절한 자연어 응답을 해주는 시스템이다. 이를 위해서는 두 가지 핵심 기술이 필요한데, 하나는 자연어로 된 질의를 정확히 이해할 수 있는 기술이며, 다른 하나는 입력받은 자연어 질의에 적합한 자연어 응답을 생성하는 기술이다. 이 두 기술을 하나의 모델로 구현하는 문제가 최근 널리 사용되는 딥러닝 아키텍처인 시퀀스-투-시퀀스(sequence-to-sequence) 모델 [1]과 구조적으로 잘 맞아 떨어지기 때문에, 대화 응답 모델 개발의 기본 모델로써 시퀀스-투-시퀀스 모델이 널리 사용된다 [2,3,4,5].

대화 시스템을 위한 시퀀스-투-시퀀스 모델은 입력받은 자연어 질의를 벡터 표현으로 변환하는 인코더와 인코딩된 질의 문장에 대한 벡터 표현을 입력받아 적절한 자연어 응답을 생성하는 디코더로 구성된다. 모델의 입력과 출력이 모두 자연어이므로 시퀀스 데이터를 다루기에 적합한 순환 신경망(recurrent neural network)이 인코더와 디코더에 사용된다. 대화 쌍 데이터가 충분한 경우 이러한 기본 형태의 시퀀스-투-시퀀스 모델로도 어느 정도 잘 동작하는 대화 모델을 학습할 수 있다는 것이 최근 학계에 보고되었으며 [2], 따라서 이를 기반으로 여러 추가 연구들이 발표되고 있다 [3,4,5].

본 연구에서는 이러한 시퀀스-투-시퀀스 모델을 통한 대화 응답 생성 모델이 잠재적으로 내포할 수 있는 문제를 다룬다. 표 1에서 볼 수 있는 것처럼 같은 의미를 나타내는 다른 자연어 질의에 대해 시퀀스-투-시퀀스 모델은 강건하지 못한 응답을 생성할 수 있다. 자연어는 그 특징상 같은 의미를 나타내는 수많은 다른 표현이 존재할 수 있다. 특히 한국어의 경우, 문장 내 표현이 동의어로 교체되는 변화 외에도 상대적으로 더 자유로운 어순 변화와 다양한 어미 변화가 가능하다. 이러한 자연어의 본질적인 특성으로 인해 기존 시퀀스-투-시퀀스 모델은 표 1과 같이 학습 시 접하지 못한 수많은 새로운 질의에 대해 제대로 된 응답을 생성할 수 없게 된다. 본 논문에서는 이런 현상을 (*말귀 못 알아듣는*) *사오정 문제* 로 명명한다. *사오정 문제*가 발생하는 근본적인 이유 중 하나는 시퀀스-투-시퀀스 모델의 인코더가 처음 보는 질의 시퀀스에 대해 핵심 의미를 잘 담고 있는 강건한 벡터 표현을 도출하지 못하기 때문이다. 이 문제는 비단 대화 데이터뿐만 아니라 모든 자연어 이해 문제에 공통적으로 나타나는 것이며, 상대적으로 학습 데이터가 작

예시 대화
Q: 공항 안에 커피를 마실 만한 곳이 *있나요?*
A: 네, 많이 있습니다.
어순의 변화
Q: 커피를 마실 만한 곳이 공항 안에 있나요?
A: 어떤 종류를 원하십니까?
단순 어미의 변화
Q: 공항 안에 커피를 마실 만한 곳이 *있습니까?*
A: 네, 바로 준비해 드리겠습니다.

표 1 어순, 어미 변화에 따른 시퀀스-투-시퀀스 모델 응답 생성 결과

고 문장 변화가 심한 한국어 학습 시에는 문제가 더욱 부각될 수 있다. 따라서 다양한 상황에서 나타날 수 있는 다양한 표현에 대해 적절한 응답을 생성하기 위해 *사오정 문제*는 대화 모델 학습 시 꼭 해결해야하는 문제이다.

본 논문에서는 *사오정 문제*를 완화하고 강건한 응답을 생성할 수 있는 디노이징 응답 생성(denoising response generation) 모델을 제안한다. 디노이징 응답 생성 모델은 기존 시퀀스-투-시퀀스 모델의 학습과정에 디노이징 메커니즘을 도입한 모델이다. 단어 순서 변경과 단어 삭제와 같은 노이즈가 가해진 질의 문장을 입력하고 그럼에도 불구하고 본래의 적절한 응답을 생성하도록 질의-응답 쌍을 학습한다. 즉, 같은 의미를 갖지만 다른 문장으로 표현되는 현상을 노이즈가 포함된 질의 문장으로 시뮬레이션하고, 서로 다른 노이즈가 가미된 여러 문장에 대해 모두 적절한 응답을 해내도록 학습하는 것이다. 이를 통해 모델의 인코더를 질의문의 세세한 표현보다는 핵심적인 의미를 포착하도록 학습함으로써 디코더가 보다 의미 적절한 응답을 생성할 수 있도록 한다.

한국어를 딥러닝 모델을 통해 학습할 때 가장 먼저 마주하는 문제는 데이터 부족이다. 대화 모델 학습의 경우 상대적으로 최근에 연구가 시작된 주제로 데이터 부족 문제가 더욱 두드러질 수 있다. 디노이징 응답 생성 모델은 다양한 노이즈가 추가된 질의-응답 쌍을 생성함으로써 데이터를 증강시키는 효과가 있어 데이터 부족 문제를 완화할 수 있다. 또한 한국어는 체언과 조사의 결합, 다양한 형태의 어미 변화 등으로 인해 어절 단위 학습 시 다루어야할 어휘양이 매우 커지며, 이는 시퀀스-투-시퀀스 모델 학습을 어렵게 만드는 요인이 된다. 이 문제를 해소하기 위해 본 논문에서는 형태소 단위와 음절 단위로 문장을 취급하여 디노이징 응답 생성 모델을 학습하는 방법을 소개한다.

제안하는 방법의 우수성을 보이기 위해 약 9만 건의 한국어 질의-응답 쌍 데이터에 대해 실험을 수행했다. 기본적인 시퀀스-투-시퀀스 모델과 제안하는 디노이징 응답 생성 모델을 정량적으로 비교하기 위해 각 모델이 생성한 응답에 대해 ROUGE score를 측정하였다. 또한 각 응답의 적절성 여부를 사람이 직접 {0, 1}로 평가하여 실험 모델들에 대해 정성 평가 역시 수행하였다. 그 결과 제안하는 모델이 비교 모델에 비해 ROUGE 점수에 대해서 최대 24%, 적절한 응답 비율에 대한 정성 평가에 대해 최대 35%, 다양한 질의에 대한 응답 능력에서 최대 34%의 성능 향상을 보임을 확인할 수 있었다.

2. 관련 연구

챗봇 관련 연구는 1966년의 ELIZA [6]로 거슬러 올라간다. ELIZA는 최초의 챗봇으로 알려져 있으며 응답을 생성하기 위해 미리 몇 가지 규칙을 정의하고 그 규칙에 따라 응답을 생성하는 방식으로 설계되었다. 이런 규칙 기반 응답 모델은 결국 제약된 성능을 보일 수밖에 없다. 따라서 최근에는 대용량 대화 코퍼스로부터 질의에 대한 응답 패턴을 학습하는 방법이 주로 연구되고 있다 [2,3,4,5]. 그 중 주목할 만한 연구로 Vinyal과 Le의 연구 [2]를 언급할 수 있다. 이 연구는 기계 번역 분야에서 큰 성과를 보인 시퀀스-투-시퀀스 모델을 사용하여 대화 데이터를 효과적으로 학습할 수 있음을 보였다. 이후 시퀀스-투-시퀀스 모델을 통해 대화의 응답을 생성할 때 발생하는 여러 문제를 해결하기 위한 연구들이 진행되고 있다. [3]와 [4]에서는 모델이 *I don't know*와 같은 학습 데이터에서 빈번히 나타나지만 의미 없는 응답을 자주 생성하는 문제를 해결하기 위해 각각 상호 정보(mutual information)와 데이터 증류(data distillation)를 이용한 방법을 제시하였다. Chen et al. 또한 의미 있는 응답을 생성하기 위해 확률 토픽 모델을 시퀀스-투-시퀀스 모델과 결합시킨 모델을 소개하였다 [5]. 그러나 이러한 연구들은 상대적으로 데이터가 풍부하고 문장 변화가 덜한 영어, 중국어를 대상으로 하여 본 연구에서 해결하고자하는 *사오정 문제*를 직접적으로 다루고 있지 않다.

본 논문에서 제안하는 디노이징 응답 생성 모델은 [7]에서 비지도 학습을 통해 강건한 문장 표현을 얻기 위해 제안한 시퀀셜 디노이징 오토인코더(SDAE)로부터 영감을 얻었다. SDAE는 문장 표현 학습을 위해 임의의 노이즈가 추가된 문장을 입력하여 원래의 문장으로 복원하는 오토인코더 학습 방법이며, SDAE를 통해 얻은 문장 표현이 여러 실험에서 좋은 성능을 보임으로써 그 우수성을 입증하였다. SDAE는 입력 문장에 대해 하나의 강건한 문장 표현을 얻어 다양한 문제에 활용하는 것이 주된 목적인 반면 본 연구에서는 강건한 문장 표현을 얻는 자체보다는 좋은 응답을 생성하는 것이 궁극적인 목적이다. 따라서 SDAE의 디노이징 모델 방식을 따르되 양방향 순환신경망과 어텐션 모델(attention model)을 도입하여 맥락에 따른 문장 표현을 얻을 수 있도록 디노이징 응답 생성 모델을 설계하였다.

조휘열 외는 한국어 대화 데이터에 대해 시퀀스-투-시퀀스 모델을 적용한 연구 [8]를 발표하여 그 가능성을 보였다. 이 연구에서는 아침, 아이돌보기 상황으로 대화의 시나리오를 제약하고 해당 시나리오에 해당하는 한국어 데이터를 활용하여 대화 모델을 학습하였다. 실험을 통해 시퀀스-투-시퀀스 모델이 한국어에 대해서도 제약된 시나리오 내에서 비교적 강건한 응답을 생성할 수 있음을 보였다.

3. 디노이징 응답 생성 모델

그림 1은 본 논문에서 제안하는 디노이징 응답 생성 모델을 도식화한 것이다. 제안하는 디노이징 응답 생성 모델은 시퀀스-투-시퀀스 모델에 디노이징 메커니즘을 도입한 모델이다. 디노이징 메커니즘은 입력으로 주어지는 시퀀스에 노이즈 함수를 이용하여 노이즈를 추가한 후 원래 목표 시퀀스를 생성하도록 학습하는 방법이다. 이를 통해 같은 뜻의 다양한 표현의 질의에도 적절한 응답을 생성하도록 한다.

3.1 시퀀스-투-시퀀스 모델

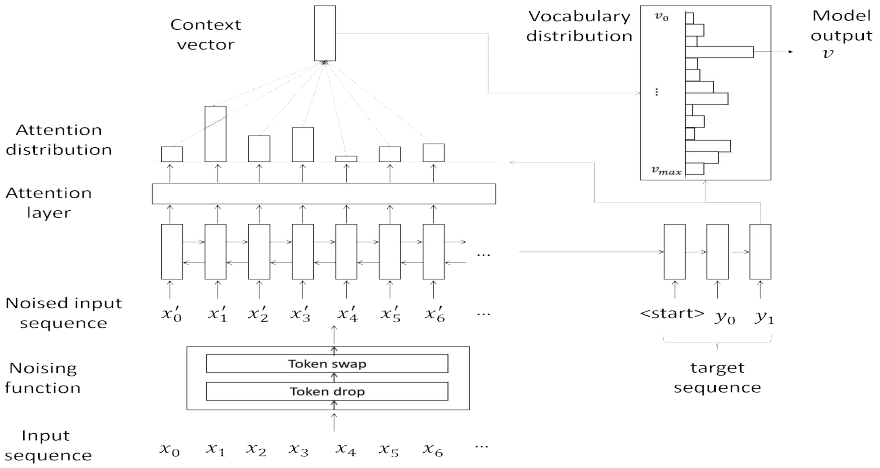


그림 1 디노이징 응답 생성 모델

```
function Noise:
  Input : input sequence S,
          token drop probability p_drop,
          token swap probability p_swap
  output : noised sequence S'

  COPY S to S'
  for i=0 to LENGTH(S):        # token drop
    if RANDOM() <= p_drop:
      REMOVE S'[i] from S'
  for i=0 to LENGTH(S)-1:      # token swap
    if RANDOM() <= p_drop:
      SWAP S'[i], S'[i+1]
  return S'
```

그림 2 노이즈 함수 의사코드

본 모델에서는 [9]에서 기계 번역을 위해 제안한 시퀀스-투-시퀀스 모델을 대화 데이터 학습을 위한 기본 모델로 차용한다. 이 모델은 인코더로 양방향(bi-directoinal) LSTM을 사용하며 디코더로는 LSTM을 사용한다. 양방향 LSTM은 시퀀스를 정방향과 역방향으로 동시 학습한 후, 양방향에서 도출된 두 개의 벡터 표현을 합쳐 입력 시퀀스에 대한 하나의 벡터 표현을 출력한다. 또한 양방향 LSTM에 더해, 어텐션(attention) 메커니즘을 적용하였다. 어텐션 메커니즘은 학습과정에서 중요하다 여겨지는 입력 시퀀스의 특정 부분을 다른 부분보다 집중적으로 반영하기 위한 방법이다. 이를 위해 중요하게 볼 시퀀스의 정보를 가진 맥락 벡터(context vector)를 생성하며, 맥락 벡터 생성을 위해 어텐션 층을 추가로 학습한다. 이렇게 생성된 맥락 벡터를 출력 시퀀스 생성과정에 반영한다.

3.2 디노이징 메커니즘

디노이징 메커니즘을 구현하기 위해서는 입력 시퀀스에 적절한 노이즈를 가할 노이즈 함수 N이 필요하다. 노이즈 함수 N은 노이즈 매개변수 p_{drop}와 p_{swap}를 기반으로 입력 시퀀스 S를 노이즈가 가미된 S'로 변환하는 함수이다.

$$S' = N(S; p_{drop}, p_{swap})$$

여기서 p_{drop}은 S 내에 각 토큰을 제거할 확률이며, p_{swap}은 S 내에 연속된 두 토큰의 위치를 교체할 확률이다. 그림 2는 노이즈 함수 N이 실제로 적용되는 알고리즘에 대한 의사 코드이다. 알고리즘에서 확인할 수 있는 것처럼 제안하는 방법에서는 토큰 제거를 먼저 적용 후 토큰 위치 교체를 적용한다.

한국어에 좀 더 적합한 디노이징 응답 생성 모델 학습을 위해 형태소를 노이즈 함수 N을 위한 토큰 단위로 사용할 것을 제안한다. 한국어 문장 데이터를 학습할 때 고려해야하는 주요 특징 중 하나는 어절의 다양함이다. 한국어는 조사, 어미 변화 등의 이유로 어절 종류가 매우 많아져 어절 단위 학습을 하기 어렵다. 형태소 단위 학습은 한국어의 어절 수 문제를 해결할 뿐만 아니라 한국어에서 빈번히 일어나는 어미, 조사 변화 등을 노이즈 함수를 통해 시뮬레이션하기에 적합하다. 즉, 노이즈 함수를 통해 조사, 어미 부분에도 확률적으로 노이즈를 부여함으로써 모델이 질의문의 본질적인 의미를 좀 더 잘 학습할 수 있게 될 것이다.

4. 실험 및 평가

4.1 데이터 셋

본 논문에서는 스마트 모바일 다국어 언어음성 데이터로부터 대화 쌍을 추출하여 실험 데이터를 구축하였다. 스마트 모바일 다국어 언어음성 데이터는 관광지, 호텔, 공항, 역, 길 등의 장소에서 두 명의 화자에 의해 이루어지는 대화를 가지고 있다. 해당 데이터 셋의 각 대화로부터 연속적인 두 개의 발화에 대해 각각을 선행 발화를 질의 문장, 후행 발화를 응답 문장으로 하여 질의-응답 쌍으로 이루어진 데이터로 만들었다. 이런 과정을 통해 총 90,729개의 대화 쌍으로 이루어진 데이터를 구축하였다. 표 2를 통해 실험에서 사용한 데이터에 대한 간략한 통계 자료를 확인할 수 있다.

표 2 데이터 구성 단위 별 데이터 통계 (단위:개)

구성 단위	대화 쌍 수	어휘 수	전체 등장한 토큰 수
어절	90,729	72,936	914,441
형태소	90,729	14,578	2,253,300
음절	90,729	1,512	3,111,117

표 3 모델 별 ROUGE F1 SCORE 측정 결과

모델	ROUGE-1	ROUGE-2	ROUGE-L
기존 모델 (어절)	0.056	0.014	0.055
기존 모델 (형태소)	0.217	0.080	0.178
기존 모델 (음절)	0.233	0.103	0.185
제안 모델 (형태소)	0.223	0.091	0.183
제안 모델 (음절)	**0.251**	**0.128**	**0.200**

4.2 실험 구성

학습 과정에서는 각 데이터의 구성 단위 어휘 수가 5만개가 넘어 갈 경우 빈도 수가 높은 상위 5만개의 어휘만을 사용하였다. 모든 모델에서 레이어는 한 층만 쌓도록 설정하였다. Hidden state 크기는 모두 1,000을 사용하였다. 모델 학습은 배치 크기를 64로 하는 미니-배치 학습 방법을 사용하였으며 Stochastic gradient descent 알고리즘을 통해 학습하였다. 기본 모델에서 어절 단위 실험의 경우 learning rate를 0.05, learning rate의 감소 비율을 0.95로 하였으며 나머지 네 모델의 경우 learning rate 0.01, 감소 비율은 설정하지 않았다. 학습 횟수는 어절 단위 데이터를 이용한 기본 모델의 경우 200회를 학습하였으며 나머지 모델은 800회를 학습하였다. 제안 모델의 노이즈 함수에서 p_{drop}과 p_{swap}은 0.1로 설정했다. 모든 실험에서 학습 데이터를 8:1:1의 비율로 학습, 검증, 평가 셋으로 나누어 진행하였으며, 검증 오류의 변화를 살펴 모델이 과다 학습(overfitting)되는 것을 막았다.

4.3 비교 모델

실험을 위해 다섯 가지 서로 다른 비교 모델을 설정하였다. 다섯 모델은 디노이징 메커니즘을 적용한 제안 모델 두 가지와 디노이징 메커니즘을 적용하지 않은 세 가지 기존 모델로 나뉜다. 기존 모델 세 가지는 각각 어절 단위, 형태소 단위, 음절 단위로 학습한 모델이며, 제안 모델 두 가지는 형태소 단위와 음절 단위로 학습한 모델이다. 음절 단위 학습은 모델이 다루어야할 어휘 수를 급격히 줄여주는 효과가 있다. 영어권에서는 음절에 해당한다고 볼 수 있는 문자 단위 학습 방법으로 의미 있는 LSTM 언어모델을 학습할 수 있음을 보였다 [10]. 또한 한국어의 경우 어근이 한 음절로 이루어진 용언과 체언이 많아 음절 단위로 시퀀스를 취급하고 노이즈를 적용하는 것이 영어의 문자 단위 학습보다 의미가 있을 수 있다.

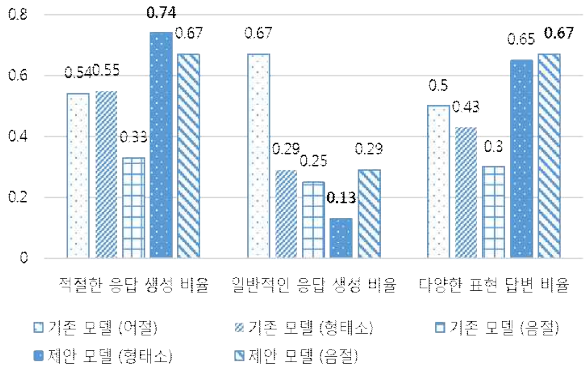


그림 3 각 모델별 정성 평가 성능 비교

4.4 정량 평가

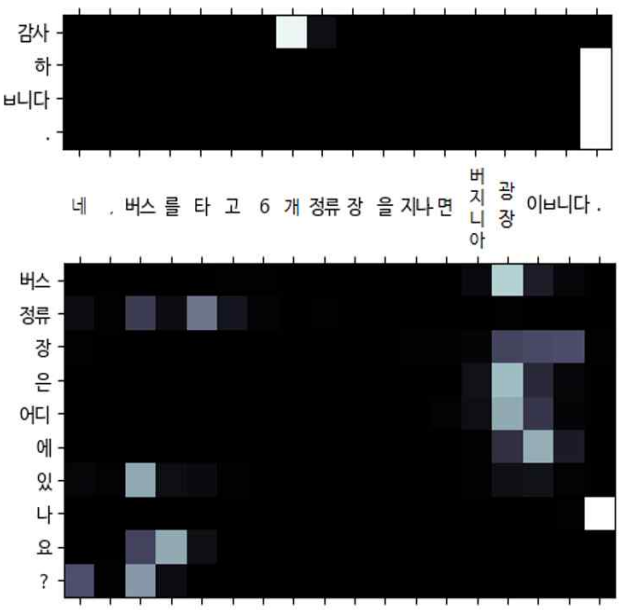
본 논문에서는 각 모델의 성능을 측정하기 위한 지표로 ROUGE F1 점수[11]를 사용했다. ROUGE F1 점수는 모델이 생성한 자연어의 품질을 정량 평가하기 위한 지표로써 모델이 생성한 문장 내의 n-gram 시퀀스들이 실제 정답 문장에 얼마나 포함되어있는지를 수치화한다.

표 3은 테스트 데이터에 대한 각 모델의 ROUGE F1 점수 측정 결과이다. 제안 모델 중 음절을 사용했을 경우가 모든 측정 방식에 대해 가장 좋은 수치를 기록하였다. 그러나 ROUGE 점수의 경우 모델이 다룰 어휘의 수가 적을수록 값이 높아질 수 있으므로 학습 토큰 단위가 같은 모델끼리 비교하는 것이 좀 더 정확하게 성능을 평가하는 방법이 될 수 있다. 이 경우에도 제안하는 방법의 형태소 모델이 기존 방법의 형태소 모델보다 최대 13.7%, 음절 모델에서 최대 24% 가량의 성능 향상을 보였다.

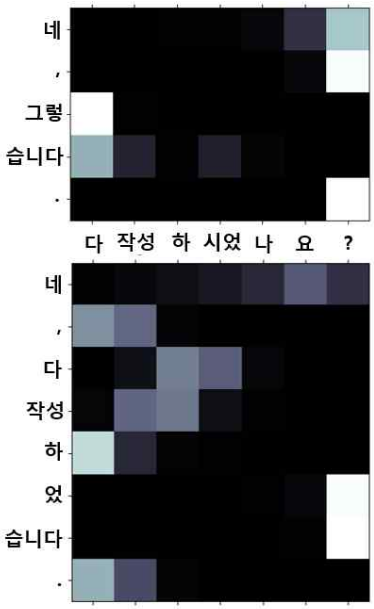
4.5 정성 평가

ROUGE 점수는 요약 등의 자연어 생성 문제에서 정량적 품질 평가 방법으로 널리 사용되고는 있지만, 대화의 경우 같은 의미를 나타내는 굉장히 다양한 자연어 표현이 가능하며 응답의 의미가 다르다 하더라도 충분히 적절한 응답일 수 있기 때문에 평가 도구로써 한계가 명확하다. 따라서 본 연구에서는 사람에 의한 정성 평가를 시행하였다. 정성 평가 방법은 모델이 생성한 응답에 대해 사람이 {적절한 응답, 부적절한 응답}을 {1, 0}으로 평가하여 전체 응답 내에서의 적절한 응답 비율을 살펴보았다. 평가는 테스트 데이터에서 임의로 929개를 추출하여 진행했다.

그림 3의 가장 왼쪽 그래프는 각 모델에서 생성한 응답에 대해 해당 응답이 적절하다고 판단된 경우에 대한 결과가 나타나 있다. 그 결과 제안 모델 중 형태소 모델이 가장 좋은 결과를 보였다. 기존 모델들 간의 결과를 살펴보면 음절 모델이 가장 좋지 않은 성능일 보였는데, 이는 문장을 음절로 쪼개어 학습한 결과 응답 생성 시 문법에 맞는 의미 있는 문장을 만들어내는 데 실패했기 때문이다. 그러나 제안하는 방법의 음절 모델의 경우 기존의 모든 모델보다 더 우수한 성능을 얻었다. 이는 제안하는 모델의 디노이징 메커니즘이 모델을 정규화(regularization)하는 힘이 있기 때문으로 보인다.



가. 질의: 네, 버스를 타고 6개 정류장을 지나면 버지니아 광장입니다.



나. 질의: 다 작성 하셨나요?

그림 4 질의에 대한 응답 생성 시 모델별 어텐션 가중치 차이 예시. 각 그림의 가로축은 질의문이며 세로축은 모델이 생성한 응답. 색이 밝을수록 어텐션 가중치가 높음. 위: 기존 모델(형태소). 아래: 제안 모델(형태소).

제안하는 디노이징 응답 생성 모델은 소위 *I don't know* 문제에도 더 강건한 성능을 보였다. 한국어 대화에 대한 응답 모델 학습 시에도 '네. 그렇습니다.' 나 '네. 알겠습니다.' 등의 지나치게 일반적인 응답을 빈번하게 생성하는 문제가 나타난다. 그림 3의 가운데 그래프는 각 모델 별 적절한 응답 중 이러한 일반적인 응답의 비율을 나타낸 것이다. 이 평가에서도 제안하는 형태소 단위의 디노이징 응답 생성 모델이 가장 적은 일반적인 응답을 생성했으며, 두 번째로 좋은 성능의 모델보다 일반적인 응답의 비율을 절반 가까이 줄였음을 확인할 수 있다. 즉, 그림 3의 왼쪽과 가운데 결과를 통해 형태소 디노이징 응답 생성 모델이 다른 모델보다 더 적절한 응답을 잘 하면서도 더 다양한 응답을 해낸다는 것을 알 수 있다.

본 논문에서 해결하고자 하는 *사오정 문제*에 대한 정성 평가 역시 수행하였다. 이를 위해 테스트 데이터 중 임의의 20개의 대화 쌍을 선택한 후 각 대화 쌍의 질의문을 여러 형태로 변환하였다. 각 질의에 대한 변환은 1. 어순 변경, 2. 특정 단어 제거, 3. 단어 변경, 4. 어미 변경, 5. 혼합의 5가지 방법을 적용하였다. 이를 통해 만들어진 총 120개의 질의에 대해 각 모델에서 생성한 응답의 적절성 평가를 실행하였다. 그림 3의 오른쪽 그래프는 각 모델 별 120개 응답 중 적절하다고 평가된 응답 비율을 나타낸 것이며, 평가한 결과 제안한 디노이징 응답 생성 모델이 기존 모델보다 두드러지게 좋은 성능을 보였다. 그 중 음절 디노이징 응답 생성 모델이 가장 좋은 성능을 보였으나 형태소 모델과 큰 차이를 보이지는 않았다.

표 4은 기존 테스트 발화 외에 앞에서 언급한 5가지 방식의 다양한 표현에 대해 형태소 모델이 생성한 응답을 보인 것으로 밑줄 친 응답은 적절치 못한 응답이다. 첫 예시의 경우 '이 바지 한번 입어봐도 될까요?' 라는 질문의 모든 변형에 대해 제안 모델은 '네, 그렇습니다.' 라는 다소 일반적인 대답을 포함하지만 적절한 응답을 생성함을 알 수 있다. 그러나 기존 모델은 마지막 변형에 대해 '네, 말씀하십시오.' 라는 엉뚱한 대답을 하였다. 두 번째 예시의 경우 제안 모델도 다소 맥락에 맞지 않는 응답을 하기도 하지만 적절한 답변도 해낸 반면, 기존 모델은 모든 질문에 대해 전혀 맥락과 다른 응답을 생성함을 확인할 수 있다.

4.6 어텐션 메커니즘을 통한 분석

제안하는 디노이징 응답 생성 모델이 모든 평가에서 좋은 결과를 보였다. 이는 제안하는 모델이 입력 데이터에 디노이징을 적용하여 학습하는 과정에서 문장의 핵심적인 의미를 학습할 수 있기 때문일 것이다. 이 과정에서 어텐션 메커니즘 또한 영향을 받는다. 그림 4는 같은 질의문에 대해 기존 모델과 제안 모델이 질의의 다른 부분에 주목을 한 결과 전혀 다른 응답을 생성하게 되는 것을 잘 보여준다. 가. 질의에 대해 기존 모델은 '개' 와 '.' 에 주목하고 '감사합니다.' 라는 매우 전형적인 응답을 생성하였다. 반면 제안 모델은 '버스', '타', '광장' 등의 질의 내용 중 중요한 단어에 주목을 함으로써 '버스정류장은 어디에 있나요?' 라는 적절하면서도 뻔하지 않은 응답을 생성할 수 있었다. 나. 질의에서도 기존 모델은 '다' 와 '?' 에만 강력히 주목한 반면 제안 모델은 '다', '작성', '하', '?' 등에 효과적으로 주목함으로써 결과적으로 '네, 그렇습니다.' 와 같은 전형적인 응답이 아닌 '네, 다 작성했습니다.' 로 보다 질문에 특화된 응답을 생성한

표 4 모델 별 같은 뜻의 다양한 표현의 발화에 대한 응답 예시
(변형1 : 어순 변경, 변형2 : 특정 단어 제거, 변형3 : 단어 변경, 변형4 : 어미 변경, 변형5 : 혼합)

예 1) Q: 이 바지 한번 입어봐도 될까요?
A: 네, 탈의실에서 입어보세요.

	기본	변형1	변형2
	Q: 이 바지 한번 입어봐도 될까요?	Q: 한번 이 바지 입어봐도 될까요?	Q: 이 바지 입어봐도 될까요?
기존 모델 (형태소)	네, 입어 보세요.	네, 입어 보세요.	네, 입어 보세요.
제안 모델 (형태소)	네, 입어 보세요.	네, 입어 보세요.	네, 입어 보세요.
	변형3	변형4	변형5
	Q: 이 바지 한번 착용 해봐도 될까요?	Q: 이 바지 한번 입어봐도 됩니까?	Q: 한번 이 바지 착용해봐도 됩니까?
기존 모델 (형태소)	아니요. 마음에 들세요.	네, 입어 보세요.	네, 말씀 하십시오.
제안 모델 (형태소)	네, 그렇습니다.	네, 입어 보세요.	네, 그렇습니다.

예2) Q: 아, 감사합니다. 버스는 몇 분마다 있나요?
A: 자주 있어요. 23분마다 한 대씩 오니까 조금만 기다리시면 될 겁니다.

기본	변형1	변형2
Q: 아, 감사합니다. 버스는 몇 분마다 있나요?	Q: 아, 감사합니다. 몇 분마다 버스가 있나요?	Q: 감사합니다. 버스는 몇 분 마다 있나요?
네 , 바로 알겠습니다.	한 번 더 타세요.	한번에 버스가 있 습니다.
그것은 10 분 마다 한 대씩 있습니다.	네, 3번에 한 번씩 타시면 됩니다.	10분 정도 소요 됩니다.
변형3	변형4	변형5
Q: 아, 감사합니다. 버스는 얼마 마다 있나요?	Q: 아, 감사해요. 버스는 몇 분 마다 있어요?	Q: 감사해요. 얼마마다 버스 는 있어요?
저쪽에 있는 정류장에서 타시면 됩니다.	저도 모르겠어요. 감사해요.	그것만 5천원이요.
3천원 입니다.	그것은 10 분 마다 한대씩 오시면 돼요.	안내원이에요.

것을 알 수 있다.

5. 결론

본 논문에서는 대화에서 나타날 수 있는 다양한 질의에 대해 강건하고 다양한 응답을 생성할 수 있는 디노이징 응답 생성 모델을 제안하였다. 제안하는 모델은 대화 응답 생성을 위해 널리 사용되는 시퀀스-투-시퀀스 모델에 디노이징 메커니즘을 추가하였다. 이를 통해 다양한 질의에 대해 적절한 응답을 생성하도록 설계했다. 실험을 통해 제안한 모델이 같은 뜻의 다양한 형태의 질의에 대해 기존 모델보다 적절한 응답을 생성할 수 있음을 확인 할 수 있었으며 부가적으로 일반적인 응답 생성 비율도 감소함을 보였다. 이를 통해 제안하는 모델의 데이터 증감 효과와 정규화 능력이 질의에 대한 강건한 의미 벡터 표현을 학습하도록 함으로써 일반적인 응답을 생성하는 비율을 줄일 수 있음을 확인하였다.

사사

이 논문은 2017년도 정부(미래창조과학부)의 재원으로 정보통신기술진흥센터의 지원을 받아 수행된 연구임 (2013-0-00109, WiseKB: 빅데이터 이해 기반 자가학습형 지식베이스 및 추론 기술 개발)

참고문헌

[1] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to Sequence Learning with Neural Networks," In *Proceedings of NIPS*, pp. 3104-3112, 2014.

[2] O. Vinyals and Q. V. Le, "A Neural Conversational Model," arXiv preprint arXiv:1506.05869, 2015.

[3] J. Li, M. Galley, C. Brockett, J. Gao, and B. Dolan, "A Diversity-Promoting Objective Function for Neural Conversation Models," In *Proceedings of NAACL-HLT*, pp. 110-119, 2016.

[4] J. Li, W. Monroe, and D. Jurafsky, "Data Distillation for Controlling Specificity in Dialogue Generation," arXiv preprint arXiv:1702.06703, 2017.

[5] C. Xing, W. Wu, Y. Wu, J. Liu, Y. Huang, M. Zhou, and W. Y. Ma, "Topic Aware Neural Response Generation," In *Proceedings of AAAI*, pp. 3351-3357, 2017.

[6] J. Weizenbaum, "ELIZA - A computer program for the study of natural language communication between man and machine," *Communications of the ACM*, Vol. 9, No. 1, pp. 36-45, 1966.

[7] F. Hill, K. Cho, and A. Korhonen, "Learning Distributed Representations of Sentences from Unlabelled Data," In *Proceedings of NAACL-HLT*, pp. 1367-1377, 2016.

[8] 조휘열, 강우영, 한동식, 장병탁, "Konvbot: 한국어 대화 모델," 한국정보과학회 학술발표논문집, pp. 624-626, 2016.

[9] D. Bahdanau, K. Cho, and Y. Bengio, "Neural Machine Translation by Jointly Learning to Align and Translate," arXiv preprint arXiv:1409.0473, 2014.

[10] A. Karpathy, J. Johnson, and L. Fei-Fei, "Visualizing and Understanding Recurrent Networks," arXiv preprint arXiv:1506.02078, 2015.

[11] C. Y. Lin, "Rouge: A Package for Automatic Evaluation of Summaries," In *Proceedings of the ACL 2004 Workshop on Text Summarization Branches Out (WAS)*, pp. 74-81, 2004.

S^2-Net: SRU 기반 Self-matching Network를 이용한 한국어 기계 독해

박천음*○, 이창기*, 홍수린**, 황이규**, 유태준**, 김현기***
강원대학교*, 마인즈랩**, 한국전자통신연구원***
{parkce, leeck}@kangwon.ac.kr, {lynn, yghwang, joon}@mindslab.ai, hkk@etri.re.kr

S^2-Net: Korean Machine Reading Comprehension with SRU-based Self-matching Network

Cheoneum Park*, Changki Lee*, Sulyn Hong**, Yigyu Hwang**, Taejoon Yoo**, Hyunki Kim***
Kangwon National University*, Mindslab**, Electronics and Telecommunications Research Institute***

요 약

기계 독해(Machine reading comprehension)는 주어진 문맥을 이해하고, 질문에 적합한 답을 문맥 내에서 찾는 문제이다. Simple Recurrent Unit (SRU)은 Gated Recurrent Unit (GRU)등과 같이 neural gate를 이용하여 Recurrent Neural Network (RNN)에서 발생하는 vanishing gradient problem을 해결하고, gate 입력에서 이전 hidden state를 제거하여 GRU보다 속도를 향상시킨 모델이며, Self-matching Network는 R-Net 모델에서 사용된 것으로, 자기 자신의 RNN sequence에 대하여 어텐션 가중치 (attention weight)를 계산하여 비슷한 의미 문맥 정보를 볼 수 있기 때문에 상호참조해결과 유사한 효과를 볼 수 있다. 본 논문에서는 한국어 기계 독해 데이터 셋을 구축하고, 여러 층의 SRU를 이용한 Encoder에 Self-matching layer를 추가한 S^2-Net 모델을 제안한다. 실험 결과, 본 논문에서 제안한 S^2-Net 모델이 한국어 기계 독해 데이터 셋에서 EM 65.84%, F1 78.98%의 성능을 보였다.

주제어: 기계 독해, 질의응답, Simple Recurrent Unit, 셀프 매칭 네트워크, 한국어 기계 독해 데이터셋

1. 서론

기계 독해(Machine Reading Comprehension)는 기계가 주어진 문맥을 이해하는 능력을 말하며, 이를 질의응답(Question Answering)에 적용하여 질문에 올바른 정답을 문맥 내에서 찾을 수 있다. 예를 들어, 기계 독해 시스템은 "국내 건조기 시장 점유율 1위 누구야?"와 같은 질문에 대하여, 문맥 "2004년 건조기 시장에 ... 의류 건조기 중 LG전자는 점유율 77.4%로 1위를 차지했다."을 이해하고, 해당 문맥 내에서 정답 "LG전자"를 찾아 출력한다.

기계 독해는 스탠포드의 SQuAD, 페이스북의 bAbi, 마이크로소프트의 MS-MARCO 등[1, 2, 3]과 같은 데이터셋이 있으며, DrQA, fastQA, R-Net, AoA reader, Bi-Directional Flow (BiDAF), Match-LSTM 등[4-9]과 같은 end-to-end 딥 러닝 모델들이 주로 연구되고 있다. 이러한 딥 러닝 모델들은 주어진 문맥과 질문에 대한 매칭 및 인코딩을 수행하고, 어텐션 매커니즘(attention mechanism)[10]을 기반으로 한 포인터 네트워크 모델(Pointer Networks)[11]을 이용하여 질문과 유사한 정답의 경계 인덱스(즉, 정답의 시작과 끝 위치)를 출력한다.

포인터 네트워크는 RNN Encoder-decoder 모델을 확장한 것으로 주어진 입력 열에 대응되는 위치를 결과로 출력하는 모델로서, 주어진 문맥에서 정답의 경계 인덱스를 찾아 결과로 출력하는데 적합하다. Self-matching Network는 R-Net 모델에서 사용된 것으로, 자기 자신의 RNN sequence에 대하여 어텐션 가중치를 계산하여 비슷한 의미의 문맥 정보를 볼 수 있기 때문에 상호참조해결과 유사한 효과를 볼 수 있는 모델이다.

Simple Recurrent Unit (SRU)은 Gated Recurrent Unit (GRU)[12]이나 Long Short-Term Memory (LSTM)[13]와 같이 neural gate를 이용하여 RNN에서 발생하는 vanishing gradient problem을 해결한 모델이다. SRU는 gate 입력에서 이전 hidden state를 제거하여 GRU와 LSTM 보다 메모리 셀의 계산 과정을 간단하게 하고 병렬화와 CUDA level 최적화를 수행하여 Convolutional Neural Network (CNN)과 유사한 속도를 보이고, cuDNN-optimized LSTM보다 5-10배 빠른 속도를 보인다. 또한 highway network를 포함하고 있어서 여러 층의 레이어를 쌓는 경우, 성능향상을 보인다[14].

본 논문에서는 기계 독해 한국어 데이터 셋을 구축하고, 여러 층의 SRU를 이용한 문맥 Encoder에 Self-matching Network를 추가한 기계 독해 모델인 S^2-Net을 제안하며, 질문 문장에 해당하는 자질을 추가하여 성능 향상을 시도한다.

2. Simple Recurrent Unit

SRU는 GRU, LSTM과 같이 neural gate를 두어 RNN의 오류 역전파(back-propagation)를 수행할 때 발생하는 vanishing gradient problem을 해결하고 gate 입력에서 이전 hidden state를 제거하여 속도를 향상시킨 새로운 recurrent unit 모델이다. SRU는 input gate i_t와 forget gate f_t, reset gate r_t, highway network를 이용하며, 그 식은 아래와 같다.

$$\tilde{x_t} = Wx_t$$
$$i_t = (1 - f_t)$$
$$f_t = \sigma(W_f x_t + b_f)$$
$$r_t = \sigma(W_r x_t + b_r)$$
$$c_t = f_t \odot c_{t-1} + i_t \odot \tilde{x_t}$$
$$h_t = r_t \odot g(c_t) + (1 - r_t) \odot x_t$$

Input gate i_t는 $\tilde{x_t}$와 element-wise 곱을 수행하여 입력 정보 반영 여부를 결정하고, forget gate f_t는 c_{t-1}과 element-wise 곱

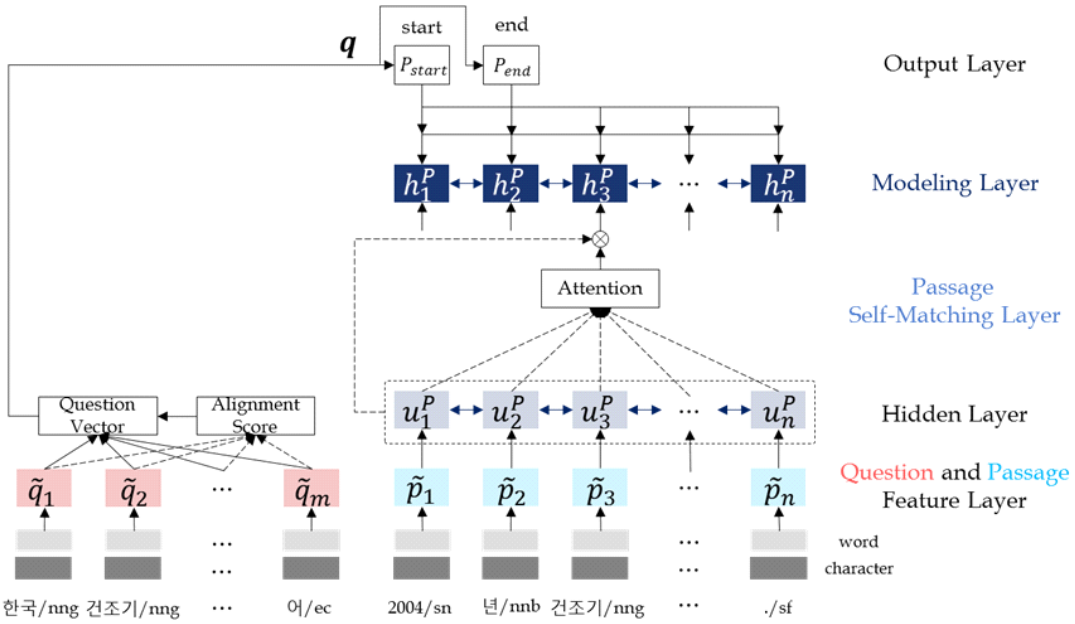


그림 2. SRU 기반 Self-matching Networks 구조

을 수행하여 이전 internal state 정보를 얼마나 반영할지를 결정한다. 여기서 $\tilde{x}_t$는 입력 x_t와 가중치 W를 곱하여 선형 변환된 결과이며, i_t는 $i_t = 1 - f_t$와 같고, f_t는 입력 x_t에 대하여 Feed-forward Neural Network (FFNN)을 수행하고 sigmoid를 적용한 결과이다. 이때 기존 RNN 모델(GRU, LSTM)들의 forget gate는 $f_t = \sigma(W_f x_t + R_f h_{t-1} + b_f)$와 같이 이전 hidden state Rh_{t-1}을 포함하였지만, SRU는 gate 계산에 FFNN을 적용하여 계산량을 줄이고, 병렬 계산을 가능하게 한다. c_t는 internal state로 입력 x_t와 이전 internal state c_{t-1}로부터의 정보 전달을 조절하고, 활성함수(activation function) $g(\cdot)$를 적용하여 internal state의 출력결과를 만든다. Hidden state h_t는 internal state 출력과 입력 x_t에 대한 highway network를 수행한 결과이다. 여기서 internal state 출력 $g(c_t)$는 reset gate r_t와 element-wise 곱을 수행하여 internal state 출력을 hidden state로 얼마나 반영할지 결정하고, 입력 x_t는 $(1 - r_t)$와 element-wise 곱으로 계산하여 입력 x_t의 반영 여부를 결정한다. [그림 1]은 SRU를 나타낸다.

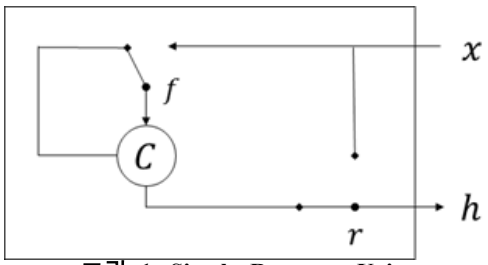


그림 1. Simple Recurrent Unit

3. SRU 기반 Self-matching Networks (S²-Net)를 이용한 한국어 기계 독해

기계 독해를 수행하기 위하여 각 모델들은 질문(Q), 문단(P), 정답(Y)의 데이터 셋이 주어진다. 질문은 m개의 단어 $Q=\{q_1, q_2, ..., q_m\}$로 구성되며, 문단은 n개의 단어 $P=\{p_1, p_2, ..., p_n\}$로 구성되고, 이를 인코딩하여 포인터 네트워크로 시작 경계 $y_1(P_{start})$, 마지막 경계 $y_2(P_{end})$를 출력한다.

본 논문에서는 한국어 기계 독해를 수행하기 위하여 SRU 기반 Self-matching 네트워크(S²-Net)를 이용하며, S²-Net 모델은 [그림 2]와 같다. S²-Net은 자질 레이어(Feature Layer)에서 문단과 질문에 대한 자질 임베딩(feature embedding)을 수행하고, 히든 레이어(Hidden Layer)에서 문단 인코딩(paragraph encoding)과 질문 인코딩(question encoding)을 수행한다. 셀프 매칭 레이어(Self-matching Layer)에서 문단 인코더 벡터(paragraph encoder vector)에 대한 셀프 어텐션을 적용하며, 모델링 레이어(Modeling Layer)에서 셀프 어텐션이 적용된 문단 인코더 벡터를 모델링하고, 출력 레이어(Output Layer)에서 정답에 대한 포인팅을 수행한다. 자질 레이어에 대한 수식은 다음과 같다.

$$\tilde{p}_t = [f_{emb}(p_t); f_{c\_emb}(p_t); f_{exact\_match}(p_t); f_{tf}(p_t); f_{align}(p_t)]$$
$$\tilde{q}_t = [f_{emb}(q_t); f_{c\_emb}(q_t); f_{exact\_match}(q_t); f_{tf}(q_t); f_{align}(q_t)]$$

$\tilde{p}_t$와 $\tilde{q}_t$는 입력된 자질 벡터이며, 이를 만들기 위하여 추출한 자질은 다음과 같다(질문인 경우 p_t대신 q_t가 입력으로 주어진다).

- 단어 표현(word embedding): $f_{emb}(p_t) = E(p_t)$
- 음절 표현(character embedding): $f_{c\_emb}(p_t) = CE(p_t)$
- 정확한 매치(exact match): $f_{exact\_match}(p_t) = \Pi(p_t \in q)$
- 토큰 자질(token feature): $f_{tf}(p_t) = TF(p_t)$
- 정렬된 질문 표현(aligned question embedding):

$$f_{align}(p_t) = \sum_j \alpha_{t,j} E(q_j),$$
$$\alpha_{t,j} = \frac{\exp(\alpha(E(p_t)) \cdot \alpha(E(q_j)))}{\sum_{j'} \exp(\alpha(E(p_t)) \cdot \alpha(E(q_{j'})))}$$

본 논문에서 단어 표현(word embedding)은 10만 단어에 대한 2년치 뉴스기사를 Neural Network Language Model (NNLM)[15]으로 학습한 것을 사용하며, 음절(또는 문자) 표현(character embedding)은 임의의 값으로 초기 값을 설정하고, CNN을 이용하여 단어에

대한 임베딩(embedding) 값을 학습한다. 정확한 매치(exact match) 자질은 문단 단어 p_t가 질문에 포함되는지 확인하는 자질(1 또는 0)이며, 문단과 질문의 각 단어는 "형태소/품사태그"로 구성된다. 토큰 자질(token feature)은 각 단어의 빈도($TF(p_t)$)를 정규화하여 자질로 사용한다. 정렬된 질문 표현(aligned question embedding)은 문단 표현과 질문 표현에 대한 얼라인먼트 스코어(alignment score)를 구하고, 문맥 인코더 벡터와 곱하여 매칭 문맥 벡터(matching context vector)를 계산하는 방법이다.

질문 자질벡터 $\tilde{q_t}$에 대하여 인코딩을 수행할 경우에는 질문 문장의 모든 hidden state를 하나의 벡터로 인코딩한다. 이때 질문 문장의 hidden state를 정규화하여 얼라인먼트 벡터 b를 만들고 이것을 질문 문장의 hidden state와 계산하여 질문 벡터(question vector) $\boldsymbol{q}$를 만든다. 질문 벡터 $\boldsymbol{q}$에 대한 수식은 다음과 같다.

$$\boldsymbol{q} = \sum_j b_j q_j$$

$$b_j = \exp(w \cdot q_j) / \sum_{j'} \exp(w \cdot q_{j'})$$

문단 인코딩을 수행하는 히든 레이어는 bidirectional SRU (BiSRU)로 구성되며, 수식은 아래와 같다. 여기서 u_t^P는 문단 입력 hidden state $\tilde{p}_t$에 대하여 인코딩된 hidden state이다.

$$u_t^P = BiSRU_P(u_{t-1}^P, \tilde{p}_t)$$

u_t^P는 모델링 레이어 h_t^P의 입력으로 사용되며, 셀프 매칭 레이어의 문맥 벡터 c_t와 연결($[u_t^P; c_t]^*$)되어 인코딩이 수행된다. $[u_t^P; c_t]^*$는 gated attention-based recurrent networks이며, sigmoid가 적용된 비선형 게이트 레이어 g_t와 $[u_t^P; c_t]$에 대하여 element-wise sum을 수행한 것이다. 셀프 매칭 레이어는 입력으로 주어진 열(sequence)을 대상(즉, 자기 자신)으로 얼라인먼트 스코어를 구하고 인코딩된 벡터들과 곱하여 문맥 벡터를 만드는 방법이며, 입력열에서 유사한 hidden state 간에 높은 얼라인먼트 스코어를 계산하고 인코딩 벡터들에 곱하여 어텐션 가중치를 조절한다. 모델링 레이어 h_t^P에 대한 수식은 아래와 같다.

$$h_t^P = BiSRU(h_{t-1}^P, [u_t^P; c_t]^*)$$

$$g_t = sigmoid(W_g [u_t^P; c_t])$$

$$[u_t^P; c_t]^* = g_t \odot [u_t^P; c_t]$$

$$s_j^t = v^T \tanh(W_u^P u_j^P + W_u^{\tilde{P}} u_t^P)$$

$$a_i^t = \exp(s_i^t) / \sum_{j=1}^{n} \exp(s_j^t)$$

$$c_t = \sum_{i=1}^{n} a_i^t u_i^P$$

S^2-Net은 포인터 네트워크의 포인팅 방법을 기반으로, 모델링 레이어에서 만들어진 h_t^P와 질문 벡터 $\boldsymbol{q}$를 bi-linear sequence attention으로 계산하여 질문에 적합한 정답의 위치를 문단에서 찾아 출력한다. 이때 출력 결과는 정답(answer span)의 시작(P_{start})과 끝(P_{end})의 위치이며, 이에 따른 수식은 다음과 같다.

$$P_{start}(t) \propto \exp(h_t^P W_s \boldsymbol{q})$$

$$P_{end}(t) \propto \exp(h_t^P W_e \boldsymbol{q})$$

본 논문에서는 정답의 시작과 끝을 출력할 때 최대 길이(max_len)를 50 형태소로 제한하였다.

4. 관련 연구

기계 독해를 해결하기 위한 기존 연구들에는 어텐션 메커니즘 기반 포인터 네트워크가 적용되며, DrQA, fastQA, R-Net, BiDAF 등의 모델이 있다. 본 논문에서 제안한 S^2-Net은 DrQA를 기반으로 음절 표현 자질과 질문 문장에 대한 자질, R-Net의 Self-matching layer를 추가하였고, 인코더에서 여러 층의 SRU를 사용하여 학습 속도 및 성능을 향상시켰다.

4-1. FastQA

FastQA는 임베딩 레이어, 인코더, 정답 레이어(Answer Layer)로 간단히 구성된다. 임베딩 레이어에서 입력 열에 단어 표현과 highway network를 적용하고, 각 단어들에 대한 정확한 매치와 정렬된 질문 표현 자질을 추출하여 모두 연결(concatenation)하여 사용한다. 본 논문에서는 히든 레이어에서 SRU를 기반으로 인코딩을 수행하는데, SRU는 highway network를 포함하고 있어, 추가적인 highway layer가 필요하지 않으며, 보다 많은 레이어를 쌓을 수 있다.

FastQA의 인코더에서는 bidirectional LSTM을 적용하여 인코딩을 수행하며, 질문과 문단의 weight matrix를 서로 공유하여 학습한다. 그 후, FFNN을 수행하는데, 여기서 사용되는 weight matrix B는 질문과 문단 벡터 각각 독립적으로 적용된다. 본 논문에서는 질문과 문단의 벡터를 인코딩할 때 서로 공유하지 않고, 정렬된 질문 표현 자질을 질문과 문단 모두 추출하여 질문과 문단 매칭을 수행하였다. 마지막으로 정답 레이어에서는 본 논문과 같이 질문 벡터 $\boldsymbol{q}$를 만들고, 문단 인코딩 벡터와 함께 ReLU기반 2-layer FFNN에 적용하여 정답의 시작과 끝을 출력한다. 본 논문에서는 ReLU기반 2-layer FFNN 레이어 없이 질문 벡터 $\boldsymbol{q}$와 문단의 모델링 벡터 h_t^P를 bi-linear sequence attention로 계산한다.

4-2. DrQA

DrQA는 웹에서 질문과 관련된 문서를 찾는 문서 검색(Document Retriever) 모듈과 찾은 문서들로부터 질문에 적합한 정답을 찾기 위하여 기계 독해를 수행하는 문서 리더(Document Reader) 모듈로 구성된다.

DrQA의 인코더는 bidirectional RNN으로 구성되며, GRU나 LSTM을 이용한다. 본 논문에서는 실험을 위하여 학습 속도가 더 빠르고 highway network를 포함하여 여러 층을 쌓을수록 성능이 증가하는 특성을 가진 SRU를 적용하였다. DrQA는 본 논문과 달리 Self-matching Layer를 포함하지 않고, 음절 표현을 사용하지 않았으며, 자질벡터 $\tilde{p_t}$에 대하여 단어 표현, 정확한 매치, 토큰 자질, 정렬된 질문 표현을 사용하였다. 여기서 토큰 자질은 $TF(p_t)$뿐만 아니라 품사태그 정보 $POS(p_t)$와 개체명 정보 $NER(p_t)$을 추가로 사용하였는데, 본 논문에서는 입력되는 단어가 "형태소/품사태그"이기 때문에 품사태그 정보를 보는 자질은 추가로 사용하지 않았다.

- 토큰 자질(token feature):

$$f_{token}(p_t) = (POS(p_t), NER(p_t), TF(p_t))$$

문단 인코딩 이후의 레이어는 Self-matching layer를 제외하고 본 논문의 방법과 같으며, 출력 결과를 계산하는 함수의 입력으로 본 논문의 모델링 레이어의 인코딩인 h_t^P 대신 문단 자질 임베딩 p_t가 주어진다.

$$P_{start}(t) \propto \exp(p_t W_s \boldsymbol{q})$$

$$P_{end}(t) \propto \exp(p_t W_e \boldsymbol{q})$$

4-3. R-Net

R-Net은 gated attention-based matching layer에서 질문과 문단을 매치시켜 질문의 의미를 포함한 문단 표현(passage representation)을 만들고, 해당 문단 표현에 대하여 셀프 매칭 어텐션 메커니즘(self-matching attention mechanism)을 기반으로

자기 자신에 대한 얼라인먼트 스코어를 계산하고 인코딩 된 hidden state와 곱하여 출력 결과를 구하는 딥 러닝 모델이다. R-Net의 경우에는 본 논문과 같이 단어 표현과 음절 표현을 사용하지만, 정렬된 질문 표현 자질 대신 질문-문단 매칭 레이어에서 어텐션 가중치를 계산하여 hidden state에 적용하고 모델링을 수행한다. 출력 레이어에서 포인터 네트워크로 정답 어텐션 가중치를 계산할 때 질문 인코딩에 대하여 셀프 어텐션으로 모델링하여 문단 인코딩과 함께 출력 결과에 대한 어텐션 스코어의 확률 분포를 구한다. R-Net은 모든 어텐션 메커니즘에 concat score를 적용하지만, 본 논문에서는 R-Net의 어텐션 스코어 방법과 달리 Bi-linear sequence attention 기반 포인터 네트워크를 이용하여 정답 경계의 위치를 출력한다.

4-4. Bi-Directional Attention Flow (BiDAF)

BiDAF는 6개의 레이어로 구성된 계층적 다단계 프로세스 모델이며, 양방향 어텐션 플로우 메커니즘(bi-directional attention flow mechanism)을 기반으로 한다. 양방향 어텐션 플로우는 Query2Context $\widetilde{H}$와 Context2Query $\widetilde{U}$를 말하며, 얼라인먼트 스코어를 계산할 때 Query2Context $\widetilde{H}$와 Context2Query $\widetilde{U}$를 문단 인코딩 H와 함께 계산하여 어텐션 가중치 G를 만든다. 그 후, 모델링 레이어에서 G를 입력으로 하여 bi-directional RNN을 수행하여 인코딩 M을 만든다.

$$G_{:t} = \beta(H_{:t}, \widetilde{U}_{:t}, \widetilde{H}_{:t})$$
$$M_{:t} = BiRNN(G_{:t})$$

위와 같이 계산 된 G와 인코딩된 M은 질문에 대한 정답을 출력하기 위하여 출력 레이어에서 서로 연결하고, Linear attention weight를 계산하여 정답의 시작(P_{start})과 끝(P_{end})을 구한다.

$$P_{start} = softmax(w^T_{(P_{start})}[G;M])$$
$$P_{end} = softmax(w^T_{(P_{end})}[G;M])$$

본 논문에서는 BiDAF의 양방향 어텐션 플로우 메커니즘과 달리, 정렬된 질문 표현 자질을 이용하여 문단 인코딩 벡터 u^P_t를 만들고, 이것을 입력으로 셀프 매칭 레이어에서 모델링을 수행하여 h^P_t를 만든 다음, 출력 레이어에서 질문 벡터와 함께 Bi-linear sequence attention을 수행한다.

5. 한국어 기계 독해 데이터셋

본 논문에서 제안한 S^2-Net을 이용한 한국어 기계 독해의 데이터 셋(MindsMRC Data Set)은 연예, 일반 도메인을 대상으로 뉴스와 위키피디아로부터 수집한 문단과 질문-정답 쌍으로 구성되며, [그림 3]과 같이 SQuAD 데이터 셋과 유사한 포맷을 따른다.

```
set -
    |-  version (str)
    -   data[]  -
                |-  title (str)
                -   paragraphs[]-
                                |-  context_original (str)
                                |-  dp[]-
                                |       |-  head (int)
                                |       |-  id (int)
                                |       |-  label (str)
                                |       |-  weight (double)
                                |       |-  text (str)
                                |       -   mods [(int)]
                                |
                                |-  qas[]-
                                |       |-  question_original (str)
                                |       |-  id (str)
                                |       |-  question_dp [[[(str)]]]
                                |       |-  question
                                |       |-  answer  -
                                |                   |-  text_original (str)
                                |                   |-  text [[[(str)]]]
                                |                   |-  answer_end int)
                                |                   -   answer_start (int)
                                |
                                -   context[[[(str)]]]
```

그림 3. 한국어 기계 독해 데이터 셋 구조

한국어 기계 독해 데이터 셋 구조는 JSON기반으로 구성되며, [그림 3]에서 set은 데이터 셋 전체를 포함하는 key이다. Version은 데이터 셋의 현재 버전을 의미하고 문자열로 표현되며, data는 현재 데이터 셋에 있는 모든 데이터를 포함하는 리스트이다. data의 title은 문서의 제목을 문자열로 나타내고, paragraphs는 문서 제목에 해당하는 문단과 질문, 정답 정보를 포함하는 리스트이다. context_original은 위키나 뉴스로부터 수집한 실제 텍스트이고, dp는 context_original에 해당하는 텍스트의 의존 구문 분석 결과를 포함하는 리스트이며, qas는 해당 문서의 질문과 정답 정보를 포함하는 리스트, context는 해당 텍스트의 형태소 정보를 포함하는 리스트이다. dp는 의존관계를 나타내는 head (중심어)와 mods (수식어), 해당 관계의 레이블 정보를 나타내는 label, 각 어절의 id, text, weight로 구성되며, weight는 의존 관계에 대한 가중치 결과이다. qas는 해당 문서의 질문과 정답 정보로 구성되며, qas에는 질문 실제 문자열인 question_original과 질문의 id, 질문의 의존 구문 분석 결과인 question_dp 리스트, 실제 질문에 대한 형태소 결과를 포함하는 question, 그에 따른 정답 정보를 포함하는 answer key가 있다. 질문에 해당하는 정답은 answer가 되며, answer는 정답의 실제 문자열인 text_origianal과 정답의 형태소 정보인 text 리스트, 정답이 문서 내에서 등장하는 시작과 끝의 인덱스 정보인 answer_start, answer_end로 구성된다. 한국어 기계 독해 데이터 셋 예제는 [표 1]과 같다.

표 1. 한국어 기계 독해 데이터 예제

Title		
장마철에도 빳빳하게... 물 만난 의류건조기		
Paragraphs		
Context_original	2004년 건조기 시장에 가장 먼저 뛰어든 LG전자를 비롯해 올해 초 삼성전자와 중견 기업까지 건조기 판매에 나서면서 국내 건조기 생산량은 급격히 늘고 있다. 건조기의 대당 판매가격을 고려했을 때 1~2년 내에 연간 시장 규모는 1조 원을 넘을 것으로 예상된다. 국내 건조기 시장은 LG전자가 주도하고 있다. 가격비교사이트 다나와리서치에 따르면 올 1월부터 6월까지 판매된 의류 건조기 중 LG전자는 점유율 77.4%로 1위를 차지했다. 가스식·전기식을 모두 판매하는 LG전자는 올해 초부터 전기식 건조기 사업에 주력하고 있다. 회사는 올해 용량과 사용 편의성을 업그레이드한 트롬 전기식 건조기 신제품 2종을 출시했다. 올해 제품에는 냉매를 순환시켜 발생한 열을 활용하는 '인버터 히트펌프' 기술을 적용했다.	
Context	[[['2004/sn', '년/nnb'], ['건조기/nng'], ['시장/nng', '에/jkb'], ['가장/mag'], ['먼저/mag'], ['뛰어들/vv', 'ㄴ/etm'], ['LG/sl', '전자/nng', '를/jko'], ['비롯하/vv', '어/ec'], ['올해/nng'], ['초/nnb'], ['삼성전자/nng', '와/jc'], ['중견/nng'], ['기업/nng', '까지/jx'], ['건조기/nng'], ['판매/nng', '에/jkb'], ['나서/vv', '면서/ec'], ['국내/nng'], ['건조기/nng'], ['생산량/nng', '은/jx'], ['급격히/mag'], ['늘/vv', '고/ec'], ['있/vx', '다/ef', './sf']], ...]]	
Question_original	한국 건조기 시장 점유율 1위 어딘지 알려줘	
Question	[[['한국/nng'], ['건조기/nng'], ['시장/nng'], ['점유율/nng'], ['1/sn', '위/nnb'], ['어디/np', '이/vcp', 'ㄴ지/ec'], ['알려주/vv', '어/ec']]]	
Answers	text_original	LG전자
	text	[[['LG/sl', '전자/nng']]]
	answer_start	95
	answer_end	97

[표 1]에서는 title, paragraphs, context_original, context, question_original, question, answers 정보에 대한 예를 보인다. Title은 문서(뉴스 또는 위키피디아)의 제목을 나타내며,

paragraphs는 문서의 본문과 질문-정답 쌍을 나타낸다. Context_original과 question_original은 문단과 질문의 raw text이며, S^2-Net으로 학습 및 예측하기 위하여 context와 question과 같이 형태소분석 수행 결과를 만든다. Answers는 정답의 raw text (text_original)와 형태소분석 결과(text)를 포함하고 있으며, 문단에서의 정답 텍스트 시작 위치(answer_start)와 마지막 위치(answer_end)로 구성된다.

6. 실험

본 논문에서 제안하는 S^2-Net와 실험에 사용한 모든 모델은 PyTorch로 구현하였으며, 실험은 Intel i7-4790 CPU (3.60GHz), 32GB RAM, TITAN X (Pascal), Ubuntu 16.04 OS에서 수행되었다.

실험에 사용된 데이터 셋은 뉴스 도메인 36,931 문서, 93,835 질문과 위키 도메인 7,119 문서, 17,948 질문으로 구성되며, [표 2]와 같이 학습 셋과 개발 셋을 9:1의 비율로 나누었다. 본 논문에서 질문 데이터를 제작할 때 유사한 질문들을 2개씩 짝지어 만들었으며, 고유한 질문의 수는 [표 2]의 고유 질문 수와 같다.

표 2. 실험에 사용한 데이터 셋 개수

데이터 셋			
	문단 수	질문 수	고유 질문
개발 (dev)	5,188	13,066	6,533
학습 (train)	39,645	100,395	50,198

본 논문에서는 S^2-Net을 이용한 한국어 기계 독해에 대하여 다음과 같이 실험을 하였다. 학습은 *Adam*[16]을 이용하고, 학습율(learning rate)을 0.1로 설정하였다. 히든 레이어와 어텐션 레이어에 대한 활성함수는 모두 tanh를 적용하였으며, 모든 RNN 레이어는 SRU (CUDA level optimization)를 이용하였다. 드랍아웃은 0.2로 고정하고, 음절 표현의 차원 수는 50 그리고 단어 표현의 차원수는 100, 히든 레이어의 차원 수는 128로 설정하였다. 음절 표현은 윈도우 사이즈 [2,3,4,5,6]의 필터(filter)를 사용하고, 필터의 크기는 30으로 설정하였다. 미니 배치의 배치 크기는 32로 설정하였으며, 매 epoch마다 개발 셋으로 성능 평가를 수행하였다. 성능 측정의 척도는 EM (Exact Match)과 F1을 사용하였다[1].

[표 3]은 본 논문에서 제안한 S^2-Net과 DrQA[4], DrQA+BiSRU [14], BiDAF, BiDAF+SM (BiDAF+Self-matching)의 성능을 나타낸다. 실험이 수행된 모델 중에서 DrQA가 실험의 baseline으로 RNN type이 LSTM이며, 나머지 모델은 모두 SRU를 적용하였다. Encoder RNN의 레이어는 3과 5로, 모델링 레이어는 1과 2로 각각 설정하였고, 그 외의 하이퍼 파라미터는 모두 동일하게 적용하였다.

표 3. SRU기반 한국어 기계 독해 모델 별 성능 (dev, %)

Model	Layers	Modeling layer	RNN type	EM	F1
DrQA(baseline)[4]	3	1	LSTM	59.25	74.37
DrQA+BiSRU[14]			SRU	64.16	77.55
BiDAF				64.01	77.29
BiDAF+SM				63.97	77.71
S^2-Net (our)				**64.37**	**78.16**
DrQA+BiSRU[14]	5	2	SRU	64.94	78.16
BiDAF				64.76	78.03
BiDAF+SM				61.78	75.67
S^2-Net (our)				**65.84**	**78.98**

실험 결과, Encoder RNN 레이어 3, 모델링 레이어 1 일 때 실험의 baseline인 DrQA는 F1 74.37%의 성능을 보였지만, DrQA에 SRU를 적용한 DrQA+BiSRU는 F1이 78.16%로 DrQA에 비하여 3.79% 향상된 성능을 보였다. 그 외로 BiDAF나 BiDAF+SM은 DrQA와 같은 실험 설정일 때 각각 F1 77.29%, F1 77.71%의 성능을 보였으며, 이에 따라 RNN type에 LSTM을 적용할 때보다 SRU를 적용할 때 성능이 향상됨을 알 수 있다. 또한 본 논문에서 제안한 S^2-Net이 EM 64.37%, F1 78.16%로 다른 모델들에 비하여 좋은 성능을 보였다.

Encoder RNN 레이어 5, 모델링 레이어 2일 때 DrQA+BiSRU는 F1 78.16%, BiDAF는 F1 78.03%로 Encoder RNN 레이어 3, 모델링 레이어 1일 때보다 각각 0.61%, 0.74% 향상된 성능을 보였지만, BiDAF+SM은 F1 75.67%로 레이어를 더 쌓으니 −2.04% 더 낮은 성능을 보였다. 본 논문에서 제안한 S^2-Net은 F1 78.98% (EM 65.84%)로 가장 좋은 성능을 보였으며, 같은 실험 환경(5-layer, 2-layer)에서 DrQA+BiSRU보다 0.82%, BiDAF보다 0.95%, BiDAF+SM보다 3.31% 더 좋은 성능을 보였고, DrQA (baseline)보다 4.61% 더 좋은 성능을 보였다.

7. 결론

본 논문에서는 SRU 기반 Self-Matching Network (S^2-Net)를 이용한 한국어 기계 독해 모델을 제안하였고, 한국어 기계 독해 데이터 셋과 그 구축 방법에 대하여 설명하였으며, S^2-Net과 DrQA, DrQA +BiSRU, BiDAF, BiDAF+SM에 대한 비교 실험을 수행하였다.

실험 결과, 한국어 기계 독해 데이터 셋에 대하여 본 논문에서 제안한 방법인 S^2-Net이 Encoder RNN 레이어 5, 모델링 레이어 2일 때 EM 65.84%, F1 78.98%로 가장 좋은 성능을 보였다. 같은 실험 환경에서 DrQA+BiSRU는 EM 64.94%, F1 78.16%, BiDAF는 EM 64.76%, F1 78.03%, BiDAF+SM이 EM 61.78%, F1 75.67%의 성능을 보였다.

향후 연구로는 기계 독해에 대한 학습 데이터를 더 구축할 것이며, hierarchical RNN 등과 같은 모델을 적용할 예정이다.

감사의 글

본 연구는 과학기술정보통신부 및 정보통신기술진흥센터의 정보통신·방송 연구개발 사업의 일환으로 하였음. [2013-0-00131, (엑소브레인-1세부) 휴먼 지식증강 서비스를 위한 지능진화형 WiseQA 플랫폼 기술 개발]

참고문헌

[1] P. Rajpurkar, J. Zhang, K. Lopyrev and P. Liang. Squad: 100,000+ questions for machine comprehension of text. *arXiv preprint arXiv:1606.05250*, 2016

[2] F. Hill, A. Bordes, S. Chopra and J. Weston. The Goldilocks Principle: Reading Children's Books with Explicit Memory Representations. *arXiv preprint arXiv:1511.02301*, 2015.

[3] T. Nguyen, M. Rosenberg, X. Song, J. Gao, S. Tiwary R. Majumder and L. Deng. MS MARCO: A Human Generated MAchine Reading COmprehension Dataset, *arXiv preprint arXiv:1611.09268*, 2016.

[4] D. Chen, A. Fisch, J. Weston and A. Bordes. Reading Wikipedia to Answer Open-Domain Questions, *arXiv preprint arXiv:1704.00051*, 2017.

[5] D. Weissenborn, G. Wiese and L. Seiffe. Making Neural QA as Simple as Possible but not Simpler, *Proceedings of the 21st Conference on Computational Natural Language Learning (CoNLL 2017)*. 2017.

[6] W. Wang, N. Yang, F. Wei, B. Chang and M. Zhou. Gated Self-Matching Networks for Reading Comprehension and Question

Answering, *In: Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers),* pp. 189-198, 2017.

[7] Y. Cui, Z. Chen, S. Wei, S. Wang, T. Liu and G. Hu. Attention-over-Attention Neural Networks for Reading Comprehension, *arXiv preprint arXiv:1607.04423*, 2016.

[8] M. Seo, A. Kembhavi, A. Farhadi and H. Hajishirzi. Bidirectional Attention Flow for Machine Comprehension. *arXiv preprint arXiv:1611.01603*, 2016.

[9] S. Wang and J. Jiang. Machine Comprehension Using Match-LSTM and Answer Pointer, *arXiv preprint arXiv:1608.07905*, 2016.

[10] D. Bahdanau, et al. Neural machine translation by jointly learning to align and translate. *Proc. of ICLR' 15*, arXiv:1409.0473, 2015.

[11] O. Vinyals, M. Fortunato and N. Jaitly. Pointer Networks. *Advances in Neural Information Processing Systems*, pp. 2674-2682, 2015.

[12] K. Cho, et al. Learning phrase representation using RNN encoder-decoder for statistical machine translation. *Proc. of EMNLP' 14*, 2014.

[13] S. Hochreiter and J. Schmidhuber. Long Short-Term Memory. *Nueral computation, 9(8)*, pp.1735-1780, 1997.

[14] T. Lei and Y. Zhang. Training RNNs as Fast as CNNs. *arXiv preprint arXiv:1709.02755*, 2017.

[15] 이창기, 김준석, 김정희. 딥 러닝을 이용한 한국어 의존 구문 분석. *제 26회 한글 및 한국어 정보처리 학술대회*, pp. 87-91, 2014.

[16] D. Kingma and J. Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Dual Bi-Directional Attention Flow를 이용한 한국어 기계이해 시스템

이현구○, 김학수, 최정규*, 김이른*
강원대학교 컴퓨터정보통신공학과, LG 전자 SW센터 인공지능연구소*
nlphglee@kangwon.ac.kr, nlpdrkim@kangwon.ac.kr, stanley.choi@lge.com, yireun.kim@lge.com

Korean Machine Comprehension using Dual Bi-Directional Attention Flow

Hyeon-gu Lee○, Harksoo Kim, Jungkyu Choi*, Yi-reun Kim*
Kangwon National University Computer and Communication Engineering
AI Lab., SW Center, LG Electronics*

요 약

기계이해 시스템은 주어진 문서를 이해하고 질의에 해당하는 정답을 출력하는 방법으로 심층 신경망을 활용한 주의집중 방법이 발달하면서 활발히 연구되기 시작했다. 본 논문에서는 어휘 정보를 통해 문서와 질의를 이해하는 어휘 이해 모델과 품사 등장 정보, 의존 구문 정보를 통해 문법적 이해를 하는 구문 이해 모델을 함께 사용하여 기계이해 질의응답을 하는 Dual Bi-Directional Attention Flow모델을 제안한다. 한국어로 구성된 18,863개 데이터에서 제안 모델은 어휘 이해 모델만 사용하는 Bi-Directional Attention Flow모델보다 높은 성능(Exact Match: 0.3529, F1-score: 0.6718)을 보였다.

주제어: 기계이해, 질의응답 시스템, 주의집중

1. 서론

질의응답 시스템(Question Answering System)은 자연어로 이루어진 질의를 시스템이 이해하고 적절한 답변을 출력해주는 시스템이다. 최근 심층 신경망 기술이 발전함에 따라 이러한 질의응답 시스템의 방법 중 기계이해(Machine Comprehension) 방식의 질의응답이 활발하게 연구되고 있다. 기계이해 시스템은 정보가 포함된 문서와 그 문서에 포함된 정보를 질문하는 질의 간의 관계를 이해시키고 이를 통해 문서에 나타나는 정보의 위치를 찾아주는 질의응답 방법이다. 다양한 정보가 빠르게 증가하여 제한된 시간 안에 많은 정보 습득이 중요해진 만큼 정보 습득을 도와주는 기계이해 시스템이 활발히 연구되고 있다. 특히 영어권에서는 SQuAD(Stanford Question Answering Dataset)[1]를 사용하는 기계이해 시스템 경진대회를 통해 다양한 모델이 연구되고 있다. 본 논문에서는 한국어 문서와 질의간의 관계를 이해시키고 정답을 찾아내기 위해 어휘와 구문 정보를 함께 활용하는 한국어 기계이해 시스템을 제안한다.

2. 관련 연구

최근 SQuAD 데이터를 사용하는 기계이해 시스템이 활발하게 연구되고 있다. 문서 어휘에서 질의 어휘의 주의집중(Attention)과 질의 어휘에서 문서 어휘의 주의집중을 계산하여 정답의 위치를 반환하는 Bi-Directional Attention Flow[2], 문서에 질의의 주의집중을 계산한 후 self-matching을 통해 문서를 다시 확인하는 R-Net[3], 문서와 질의의 벡터를 내적하는 단순한 방법으로 관계를 계산하고 언어모델(Language Model)을 통해 정답을 찾아내는 Interactive AoA Reader[4]등이 연구되고 있다. 또한 구문적 정보를 반영하기 위해 구조 정보를 임베딩하여 입력으로 반영한 연구[5], 개체명과 품사를 반영한 연구 등 언어의 특성을 반영하기 위한 연구[6]도 진행되었다. 본 논문에서는 Bi-Directional Attention Flow모델을 한국어에 맞게 변형하고 품사 등장 정보 및 의존 구문 정보를 추가하여 성능을 향상시키는 Dual Bi-Directional Attention Flow를 제안한다.

3. Dual Bi-Directional Attention Flow

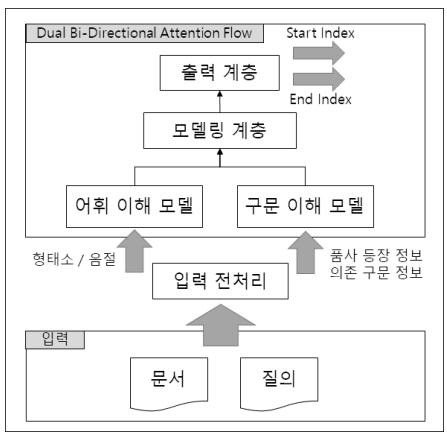


그림 1. 제안 모델의 구조도

[그림 1]은 제안 모델의 구조도를 보여준다. 제안 모델은 문서와 질의 사이의 관계를 이해하기 위해 어휘 정보를 통해 단어와 단어 간의 관계를 찾아내는 어휘 이해 모델과 품사 등장 정보 및 의존 구문 정보를 통해 문법적 관계를 찾아내는 구문 이해 모델로 구성된다. 어휘 이해 모델은 형태소와 음절을 통해 어절 임베딩을 생성하고 문서와 질의 간의 주의집중을 계산한다. 구문 이해 모델은 문장에 나타나는 어절의 품사 등장 정보와 의존 구문분석을 통해 얻어진 의존 구문 표지의 최단 경로 정보를 통해 임베딩을 생성하고 문서와 질의 간의 주의집중을 계산한다. 두 가지 모델을 통해 얻어진 어휘 주의집중 벡터와 구문 주의집중 벡터를 Long Short-Term Memory(LSTM) 순환신경망(Recurrent Neural Network)[7]으로 구성된 모델링 계층에 전달하여 어절 간의 정보를 모델링하고 출력 계층에서 정답에 해당하는 문서의 어절 위치를 반환한다.

3.1. 한국어 어휘 이해 모델

본 논문에서는 문서와 질의 사이의 관계를 이해시키기 위해 먼저 형태소와 음절을 통해 어절 임베딩을 생성한다. 다음으로 문서와 질의에 나타나는 어절 간의 주의집중을 계산하여 문서에 나타나는 어절과 질의에 나타나는 어절 간의 연관성을 찾아낸다. [그림 2]는 한국어 어휘 이해 모델이다.

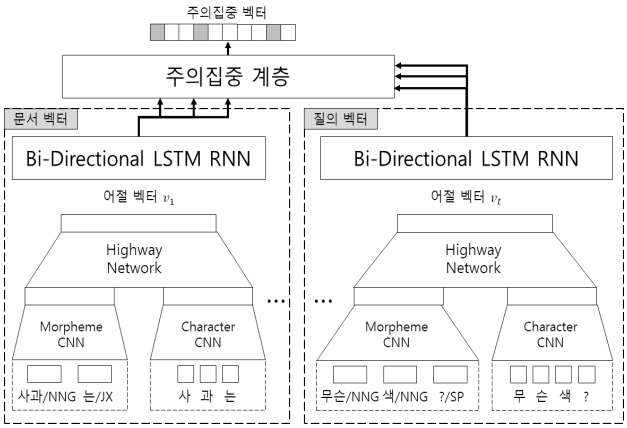


그림 2. 어휘 이해 모델

[그림 2]에서 형태소와 음절을 합성곱 신경망(Convolutional Neural Network)[8]을 통해 어절단위로 각각 임베딩 한 후 생성된 임베딩을 Highway Network[9]에 전달하여 형태소와 음절이 반영된 어절 임베딩을 생성한다. Highway Network는 LSTM과 같은 게이트 이론이 반영된 모델로 기울기 소실 문제(Vanishing gradient problem)[10]를 해결해 벡터를 확실히 전달할 수 있게 하는 모델이다. 각 어절별 벡터가 생성된 후 양방향 LSTM 순환 신경망을 통해 어절에 문장 단위 정보를 반영한다. 마지막으로 생성된 문서의 어절 임베딩과 질의의 어절 임베딩을 주의집중 계층에 입력하여 주의집중 관계를 계산한다. 주의집중 계층은 식 (1)과 같이 계산된다.

$$
\begin{aligned}
&V_{ij} = \alpha(S_i, Q_j) \\
&\alpha(S_i, Q_j) = W^T[S_i; Q_j; S_i \circ Q_j] \\
&a_i = softmax(V_i) \\
&\widetilde{Q}_i = \sum_{j=0}^{n} a_{ij} Q_j \\
&b = softmax(\max(V_i)) \\
&\widetilde{S}_i = \sum_{i=0}^{m} b_i S_i \\
&F_i = \beta(S_i, \widetilde{Q}_i, \widetilde{S}_i) \\
&\beta(S, \widetilde{Q}, \widetilde{S}) = [S; \widetilde{Q}; S \circ \widetilde{Q}; S \circ \widetilde{S}]
\end{aligned}
\tag{1}
$$

식 (1)에서 S_i는 문서의 i번째 어절의 벡터, Q_j는 질의의 j번째 어절의 벡터, $S_i \circ Q_j$는 S_i와 Q_j의 요소별 곱셈(elementwise multiplication)이다. $\widetilde{Q}$는 질의가 문서에 작용되는 주의집중 벡터, $\widetilde{S}$는 문서가 질의에 작용되는 주의집중 벡터이다. 즉, 식 (1)에서는 문서와 질의간의 주의집중 가중치를 구하고 문서의 벡터와 결합하여 질의가 문서의 어떤 어절에 중요하게 작용하는지를 찾아낸다.

3.2. 한국어 구문 이해 모델

3.1절에서 언급한 어휘간의 이해 외에도 문법적 관계를 이해하기 위해 어절의 품사 등장 정보 및 의존 구문 정보를 사용하는 구문 이해 모델을 제안한다. [그림 3]은 한국어 구문 이해 모델이다.

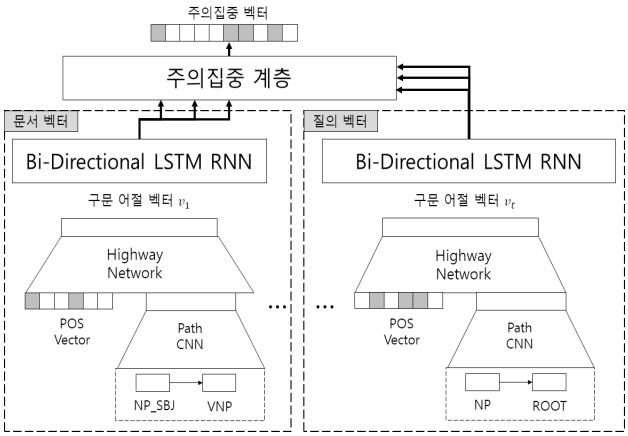


그림 3. 구문 이해 모델

[그림 3]의 품사 등장 정보는 현재 어절에 나타나는 품사를 이진 벡터(어절에서 나타나면 1 아니면 0; 세종 품사 45차원)로 표현하고 의존 구문 정보는 현재 어절에서 문장의 루트 어절까지의 의존 구문 표지 최단 경로 정보로 3.1절의 어휘 벡터 생성과 같이 합성곱 신경망을 적용하여 벡터를 생성한다. 생성된 의존 구문 정보 벡터와 품사 등장 이진 벡터를 Highway Network에 입력하여 구문 어절 벡터를 생성하고 양방향 LSTM 순환 신경망에 입력하여 문장 정보가 반영된 구문 어절 벡터를 생성한다. 마지막으로 3.1절과 마찬가지로 식 (1)을 적용하여

구문 정보가 반영된 문서와 질의의 구문 어절 벡터 간의 주의집중 관계를 계산한다. 계산된 주의집중은 문서에서 정답을 찾기 위해 질의의 품사와 의존 구문 표지가 문서의 어떤 품사, 의존 구문 표지와 연관되는지를 찾아낸다.

3.3. 한국어 기계이해 질의응답 시스템

본 논문에서 제안하는 모델은 한국어 어휘 이해 모델과 구문 이해 모델에서 계산된 주의집중 벡터를 통해 정답에 해당하는 문서의 어절 위치를 반환한다. 정답을 찾기 위한 계층은 어절별로 출력된 주의집중 벡터를 양방향 LSTM 순환신경망을 이용하여 문장단위로 정보를 찾아내는 모델링 계층과 찾아진 정보를 통해 정답의 시작 위치와 끝 위치를 반환하는 출력 계층으로 이루어진다. 모델링 계층은 어휘 이해 모델의 주의집중 벡터와 구문 이해 모델의 주의집중 벡터를 연결한 후 2층으로 구성된 양방향 LSTM 순환신경망에 입력하는 구조로 이루어져있다. 다음으로 모델링 레이어의 출력 결과를 정답의 어절 위치를 찾아내는 출력 계층의 입력으로 사용한다. 출력 계층은 식 (2)와 같이 계산한다.

$$start = softmax(W_{start}^{T}[F_{lexicon}; F_{syntax}; M]) \\ end = softmax(W_{end}^{T}[F_{lexicon}; F_{syntax}; M^{2}]) \quad (2)$$

식 (2)에서 $F_{lexicon}$은 어휘 이해 모델의 주의집중 벡터 F_{syntax}은 구문 이해 모델의 주의집중 벡터, M은 모델링 계층의 결과 벡터, M^{2}은 $start$값을 또 다른 양방향 LSTM 순환신경망에 입력하여 출력된 벡터이다.

3.4. 손실 함수

본 논문에서는 학습을 위해 식 (3)과 같은 손실 함수(loss function)를 사용한다.

$$L(\theta) = -\frac{1}{N}\sum_{i=0}^{N}\log(start_{y_i^{start}}) + \log(end_{y_i^{end}}) \quad (3)$$

식 (3)에서 $start_{y_i^{start}}$와 $end_{y_i^{end}}$는 출력 계층의 $start$와 end의 확률 분포에서 실제 정답의 위치에 해당하는 실수값을 나타낸다.

4. 실험 및 평가

4.1. 실험 준비

본 논문에서는 한국어 기계이해 질의응답 시스템을 실험하기 위해 자체 제작한 질의응답 데이터 18,863개를 사용한다. 질의응답 데이터는 여러 문장으로 구성된 문서, 질의, 질의에 해당하는 답변 및 어절 위치로 구성되어 있으며 학습 데이터 18,456개, 평가 데이터 357개, 개발 확인 데이터 50개를 무작위로 나누어 사용한다. 실험에서 사용한 형태소 분석기와 구문 분석기는 Kacteil 언어 분석기[11-12]를 사용했으며 성능은 형태소 정확도 95.21%, UAS 87.21% LAS 85.28%이다. [그림 4]는 실험에서 사용하는 데이터의 예시이다.

Context	데니스 매캘리스테어 리치(1941년 9월 9일 ~ 2011년 10월 12일)는 미국의 저명한 전산학자이자 현대 컴퓨터의 선구자이다. C와 유닉스로 알려져있다. 1983년에 켄 톰프슨과 “범용 운영체제 이론개발, 특히 유닉스 운영체제의 구현에 대한 공로”로 튜링상을 수상했다.
Question	1983년 누구와 함께 튜링상을 수상했나?
Answer	켄 톰프슨과
Index	start : 1 end : 3

그림 4. 실험 데이터의 예

4.2. 실험 평가

본 논문에서 모델의 성능을 확인하기 위해 기계이해 시스템에서 많이 사용되는 완전 일치율(Exact Match)과 형태소 단위의 F1-score를 성능 지표로 사용한다. 완전 일치율은 모델이 예측한 시작 위치에서부터 끝 위치까지의 형태소가 정답과 모두 일치하는 경우를 의미하고 형태소 단위 F1-score는 모델이 예측한 위치 사이에 존재하는 형태소가 실제 정답과 얼마나 일치하는지를 F1-score로 측정한 성능이다. 실험에 사용되는 모델은 어휘 이해 모델만을 사용하는 Bi-Directional Attention Flow(BiDAF), 어휘 이해 모델과 구문 이해 모델을 함께 사용하는 제안 모델 Dual Bi-Directional Attention Flow(Dual BiDAF)가 있다. 표 1은 모델의 성능을 보여준다.

표 1. 모델별 성능 비교 (%)

모델	Exact Match	F1-Score
BiDAF	33.33	64.58
Dual BiDAF	35.29	67.18

표 1에서 보는 것과 같이 어휘 이해 모델에 구문 이해 모델을 같이 사용하는 Dual Bi-Directional Attention Flow가 어휘 이해 모델만 사용하는 Bi-Directional Attention Flow보다 완전 일치율 1.96%p, F1-Score 2.6%p 높게 나왔다. 즉, 어휘만을 사용했을 때보다 품사 등장 정보, 의존 구문 정보의 주의집중 벡터가 정답의 경계를 찾아내는데 도움을 주는 것을 알 수 있다.

영어권에서 연구되고 있는 SQuAD(Stanford Question Answering Dataset) 데이터를 사용하는 기계이해 시스템의 경우 완전 일치율 0.7, F1-score 0.8정도로 본 논문에서 제안한 한국어 모델과 큰 성능차이를 보이고 있다. 이는 SQuAD 학습 데이터가 87,599개로 매우 많은 양을 학습하는 반면 한국어 데이터는 18,456개 밖에 학습을

하지 않기 때문이다. 따라서 제안 모델이 데이터가 증가될수록 성능이 올라 갈 수 있는 가능성을 보기 위해 학습 데이터의 개수에 따른 성능을 측정한다. [그림 5]는 학습 데이터의 개수 변화에 따른 성능의 차이를 나타낸다.

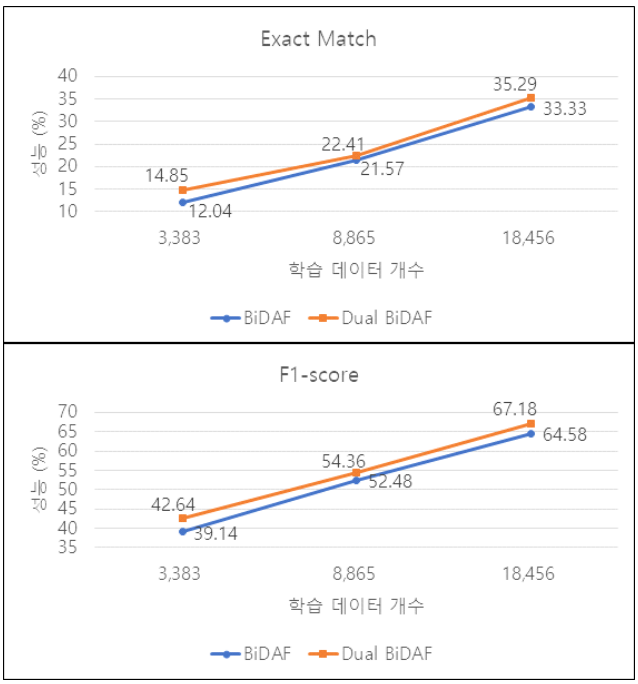


그림 5. 학습 데이터 개수에 따른 성능 변화

[그림 5]에서 학습 데이터의 개수가 증가됨에 따라 완전일치율과 F1-score 성능이 모두 향상되는 것을 확인할 수 있다. 이는 심층 신경망 모델의 복잡함으로 인해 많은 양의 파라미터를 학습해야하지만 데이터가 적을 경우 다양한 어휘 및 구문간의 관계를 파악하는 파라미터를 충분히 학습하지 못했기 때문이다. 즉, 많은 양의 학습 데이터를 확보할 경우 보다 높은 성능을 달성할 수 있을 것으로 예상된다.

5. 결론 및 향후 연구

본 논문에서는 어휘 이해만을 사용하는 기존의 Bi-Directional Attention Flow모델에 구문 이해를 추가한 한국어 Dual Bi-Directional Attention Flow모델을 제안하였다. 실험 결과 어휘 이해 모델만을 사용하는 Bi-Directional Attention Flow모델에 품사 등장 정보, 의존 구문 정보를 주의집중 계층을 통해 반영해주는 구문 이해 모델을 추가하여 완전 일치율, F1-score가 각각 1.96%p, 2.6%p 향상되는 결과를 얻었다. 향후 연구로 완전 일치율의 성능을 향상시키기 위해 어절 간의 연결 오류를 형태소의 기능어를 통해 해결해 볼 예정이다.

감사의 글

본 연구는 LG전자 산학연구용역 과제의 지원을 받아 수행되었음.

참고문헌

[1] P. Rajpurkar, J. Zhang, K. Lopyrev and P. Liang, SQuAD: 100,000+ Questions for Machine Comprehension of Text, *arXiv preprint arXiv:1606.05250*, 2016.

[2] M. Seo, A. Kembhavi, A. Farhadi and H. Hajishirzi, Bidirectional attention flow for machine comprehension, *arXiv preprint arXiv:1611.01603*, 2016.

[3] W. Wang, N. Yang, F. Wei, B. Chang and M. Zhou, Gated Self-Matching Networks for Reading Comprehension and Question Answering, *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Vol. 1. 2017.

[4] Y. Cui, Z. Chen, S. Wei, S. Wang, T. Liu and G. Hu, Attention-over-Attention Neural Networks for Reading Comprehension, *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Vol. 1. 2017.

[5] R. Liu, J. Hu, W. Wei, Z. Yang and E. Nyberg, Structural Embedding of Syntactic Trees for Machine Comprehension, *arXiv preprint arXiv:1703.00572*, 2017.

[6] B. Pan, H. Li, Z. Zhao, B. Cao, D. Cai and X. He, MEMEN: Multi-layer Embedding with Memory Networks for Machine Comprehension, *arXiv preprint arXiv:1707.09098*, 2017.

[7] S. Hochreiter and J. Schmidhuber, Long short-term memory, *Neural computation* 9.8, pp. 1735-1780, 1997.

[8] Y. Kim, Convolutional neural networks for sentence classification, *arXiv preprint arXiv:1408.5882*, 2014.

[9] R. K. Srivastava, K. Greff and J. Schmidhuber, Highway networks, *arXiv preprint arXiv:1505.00387*, 2015.

[10] S. Hochreiter, The vanishing gradient problem during learning recurrent neural nets and problem solutions, *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems* 6.02, pp. 107-116, 1998.

[11] 최맹식, 김학수, “기계학습에 기반한 한국어 미등록 형태소 인식 및 품사 태깅”, *정보처리학회논문지 제 18-B권 1*, pp. 45-50, 2011.

[12] 최맹식, 정석원, 김학수, “CRFs를 이용한 의존구조 분석 및 의존 관계명 부착”, *정보과학회논문지 : 소프트웨어 및 응용 41(4)*, pp. 302-308, 2014.

● 구두발표 3: 정보검색

TextRank 알고리즘과 주의 집중 순환 신경망을 이용한 하이브리드 문서 요약

정석원○, 이현구, 김학수
강원대학교
nlpsw@kangwon.ac.kr, nlphglee@kangwon.ac.kr, nlpdrkim@kangwon.ac.kr

Hybrid Document Summarization using a TextRank Algorithm and an Attentive Recurrent Neural Networks

Seok-won Jeong○, Hyeon-gu Lee, Harksoo Kim
Kangwon National University

요 약

문서 요약은 입력 문서가 가진 주제를 유지하면서 크기가 축약된 새로운 문서를 생성하는 것이다. 문서 요약의 방법론은 크게 추출 요약과 추상 요약으로 구분된다. 추출 요약의 경우 결과가 문서 전체를 충분히 대표하지 못하거나 문장들 간의 호응이 떨어지는 문제점이 있다. 최근에는 순환 신경망 구조의 모델을 이용한 추상 요약이 활발히 연구되고 있으나, 이러한 방법은 입력이 길어지는 경우 정보가 누락된다는 문제점을 가지고 있다. 본 논문에서는 이러한 단점들을 해소하기 위해 추출 요약으로 입력 문서의 중요한 일부 문장들을 선별하고 이를 추상 요약의 입력으로 사용했을 때의 성능 변화를 관찰한다. 추출 요약을 통해 원문 대비 30%까지 문서를 요약한 후 요약을 생성했을 때, ROUGE-1 0.2802, ROUGE-2 0.1294, ROUGE-L 0.3254의 성능을 보였다.

주제어: 문서 요약, TextRank, 순환 신경망

1. 서론

문서 요약(Document summarization)은 입력 문서가 가진 주제를 유지하면서 크기가 축약된 새로운 문서를 생성하는 과정이다. 문서 요약의 방법론은 크게 추출 요약(extractive summarization)과 추상 요약(abstractive summarization)으로 구분된다. 추출 요약은 원본 문서가 가진 문장들을 그대로 활용하여 요약하는 것으로, 문장들의 상대적 중요도에 따라 가장 중요한 일부 문장을 선별함으로써 문서를 요약한다. 추출 요약은 과거부터 활발히 연구되었으며[1, 2], 비교적 단순한 방법으로도 그럴듯한 결과들을 보였다. 그러나 추출 요약의 결과로 선별된 문장들이 문서 전체를 충분히 대표하지 못하거나, 선별된 문장들 간의 호응이 떨어지는 문제점들이 존재한다. 최근에는 순환 신경망(Recurrent Neural Network) 구조의 모델을 이용해 원본 문서에 없는 새로운 문장들을 생성해내는 추상 요약이 활발히 연구되고 있다[3, 4, 5]. 그러나 추상 요약은 특정 출력을 반복하거나 입력이 길어지는 경우 일부 정보들이 누락되는 등의 문제점들이 있다. 본 논문에서는 이러한 문제들을 해소하기 위해 추출 요약을 통해 중요 문장들을 선별하고, 이를 추상 요약의 입력으로 사용하여 원문에 없는 새로운 문장을 생성하는 하이브리드 문서 요약을 제안한다.

본 논문의 구성은 다음과 같다. 2장에서는 관련 연구들에 대해서 살펴보고, 3장에서는 제안 시스템의 추출 요약과 추상 요약 방법에 대해서 설명한다. 4장에서는 실험에 대해서 살펴보고 5장에서 끝을 맺는다.

2. 관련 연구

PageRank[6]는 하이퍼링크를 통해 연결된 웹 문서 간의 상대적 중요도를 계산하는 그래프 기반 순위화 알고리즘으로, 중요한 페이지일수록 더 많은 인용을 받는다는 것을 기초로 한다. TextRank[7]는 PageRank의 개념을 자연어 처리에 응용한 것으로 문장, 단어와 같은 특정 단위들 간의 중요도를 계산하는 알고리즘이다. 문서 내의 각 문장을 그래프의 정점(vertex)으로 가정하는 경우 중요한 문장들을 선별할 수 있으며, 이를 통해 문서 요약이 가능하다. 같은 원리로 각 단어를 정점으로 가정할 경우 중요 키워드를 선별할 수 있다. 본 논문에서는 TextRank를 이용한 추출 요약으로 입력 문서의 중요한 문장들을 선별하여 추상 요약의 입력으로 사용한다.

최경호 외[8]는 주의 집중 순환 신경망(Attentive RNN)[9]을 비롯한 다양한 모델을 이용해 한국어 문서 요약을 수행하였으며 음절, 형태소, 음절+형태소 혼합 등 입력 형태에 따른 추상 요약의 성능을 비교하였다. 이 결과를 통해 본 논문에서는 형태소 단위의 입력으로 추상 요약을 수행한다.

3. 제안 시스템

제안 시스템은 추출 요약을 통해 입력 문서를 먼저 요약한 뒤, 그 결과를 추상 요약의 입력으로 사용하는 파이프라인 형태의 구조이다. 구조도는 [그림 1]과 같다.

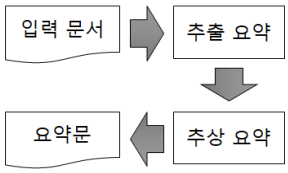


그림 1 시스템 구조도

3.1 TextRank를 이용한 추출 요약

본 논문에서는 TextRank를 이용한 추출 요약으로 입력 문서의 문장들을 선별한다. 이를 위해 입력 문서의 각 문장들에 대해 형태소 분석을 수행하고, 체언류와 용언류의 $TF \cdot IDF$를 계산하여 문장-단어 행렬을 생성한다. 그 뒤 생성된 문장-단어 행렬의 전치 행렬을 구하여 서로 곱해주면 문장 간의 상관관계(correlation)를 나타내는 행렬을 얻을 수 있다. 상관행렬을 구하는 예시는 아래 [그림 2] 같다.

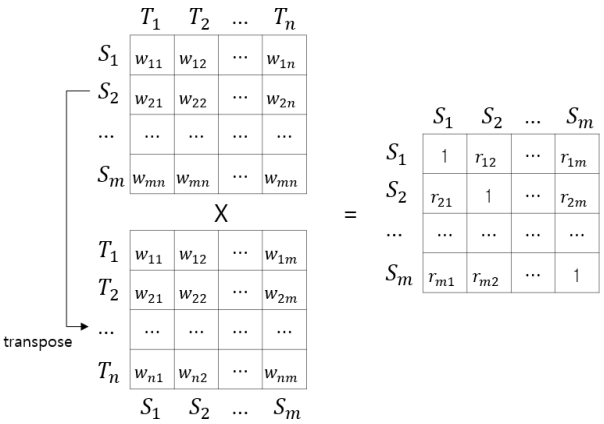


그림 2. 문장 간 상관행렬 예

문장 간 상관행렬은 문장 간의 가중치 그래프로 나타낼 수 있으며, TextRank 알고리즘을 통해 각 문장의 중요도를 구할 수 있다. TextRank의 수식은 아래 식 (1)과 같다.

$$TR(V_i) = (1-d) + d^* \sum_{V_j \in \mathrm{In}(V_i)} \frac{w_{ji}}{\sum_{V_k \in Out(V_j)} w_{jk}} TR(V_j) \quad (2)$$

TextRank를 통해 구한 중요도 순으로 문장들을 정렬한 뒤, 상위 n개 문장 외의 나머지를 제거하고 남은 문장들을 출현 순서대로 재배치하면 요약 결과를 얻을 수 있다. TextRank를 이용한 추출 요약의 결과 예시는 [표 1]과 같다.

(기사 출처 : http://www.insight.co.kr/news/87744)

표 1. TextRank를 이용한 추출 요약 예

원본 문서	22일 서울 여의도 국회에서 진행중인 최순실 국정농단의혹사건 진상규명을 위한 국정조사특별위원회 제5차 청문회에서 위원들은 우병우 전 수석을 향해 끊임없이 "최순실을 아느냐"고 질문했다. 이에 한결같이 "모른다"는 답변만 늘어놓은 우 전 수석에게 손혜원 더불어민주당 의원은 일침을 가했다. 손 의원은 "우병우 증인은 거짓말을 할 때 눈을 깜빡 깜빡 3번한다"면서 "지금도 역시 눈을 깜빡 거리고 있다"고 소리쳤다. 이 말을 들은 우 전 수석은 웃음이 나는지 이를 참기 위해 애쓰는 모습을 보이기도 했다. 손 의원이 질문을 바꿔 "차은택도 모르냐"고 호통했고, 우 전 수석은 역시 "모른다"고 답변했다. 그러자 손 의원은 "차은택은 평소에도 우병우 수석이 봐준다고 했다"며 참고인으로 출석한 K스포츠 재단 노승일 부장을 향해 "우병우가 정말 차은택을 모르는 것 같냐"고 질의했다. 이에 노 전 부장은 "차은택의 법조 조력자가 김기동인데, 우병우가 김기동을 소개시켜줬다는 이야기를 고영태에게 들었다"고 증언했다. 이처럼 참고인 노 전 부장을 비롯한 다른 증인들과 우 전 수석이 전혀 다른 주장을 펼치면서 누군가는 위증을 하고 있다는 의혹이 커지고 있는 상황이다. 한편 이날 청문회에서 우 전 수석은 질의를 받는 동안 불량한 태도로 김성태 국조특위 위원장에게 지적을 받기도 했다.
요약 결과	이에 한결같이 "모른다"는 답변만 늘어놓은 우 전 수석에게 손혜원 더불어민주당 의원은 일침을 가했다. 이처럼 참고인 노 전 부장을 비롯한 다른 증인들과 우 전 수석이 전혀 다른 주장을 펼치면서 누군가는 위증을 하고 있다는 의혹이 커지고 있는 상황이다.

3.2 주의 집중 순환 신경망

주의 집중 순환 신경망은 기본적인 형태의 순환 신경망 인코더-디코더(RNN encoder-decoder)에 주의 집중(attention mechanism)을 추가한 구조이다. 인코더-디코더 구조의 신경망은 입력을 길이에 상관없이 고정된 크기의 벡터로 인코딩하며, 이로 인해 누락되는 정보가 생길 수 있다. 주의 집중 순환 신경망은 고정 길이 벡터의 사용이 인코더-디코더 구조의 성능을 향상시키는 데 있어 병목 현상이 되는 것을 방지하기 위해, 모델이 자동으로 입력의 적절한 부분을 찾도록 확장한다[9]. 이를 통해, 주의 집중 순환 신경망은 기계 번역 분야에서 좋은 성능을 보였으며, 다른 자연어처리 분야에도 효과적으로 활용된다. 주의 집중 순환 신경망의 구조도는 아래 [그림 3]과 같다.

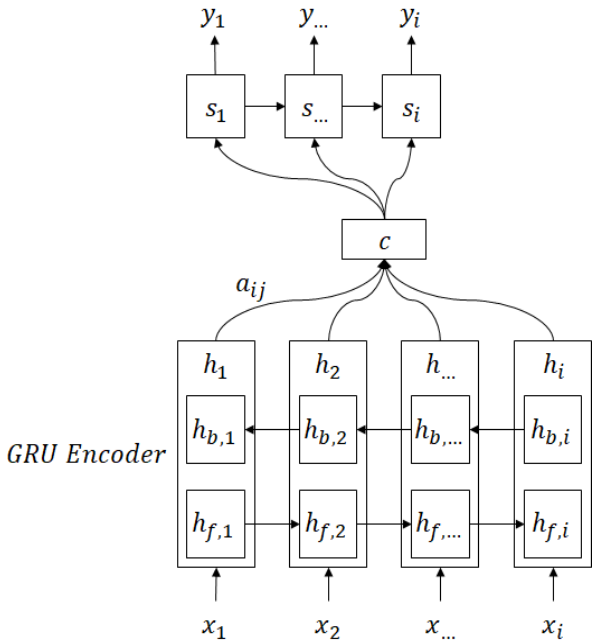


그림 3. 주의 집중 순환 신경망 구조

주의 집중 순환 신경망 모델의 인코더로는 양방향 GRU(Gated Recurrent Unit)를 사용한다. 모델을 수식으로 표현하면 아래 식 (2)과 같다.

$$
\begin{aligned}
h_{f,i} &= GRU(x_i, h_{f,i-1}) \\
h_{b,i} &= GRU(x_i, h_{b,i+1}) \\
h_i &= [h_{f,i}; h_{b,i}] \\
e_{ij} &= f(s_{i-1}, h_j) \\
\alpha_{ij} &= \frac{\exp(e_{ij})}{\sum_{k=1}^{T_x} \exp(e_{ik})} \\
c_i &= \sum_{j=1}^{T_x} \alpha_{ij} h_j \\
s_i &= f(s_{i-1}, c_i)
\end{aligned}
\qquad (3)
$$

식 (2)에서 입력 열 $X = \{x_1, x_2, \cdots, x_i\}$는 GRU를 통해 인코딩되며 $h_{f,i}$와 $h_{b,i}$는 각각 정방향, 역방향의 은닉 계층을 나타내고, h_i는 두 은닉 계층의 결합을 나타낸다. a_{ij}는 주의 집중 가중치를 나타내며 c_i는 주의 집중 가중치 및 입력을 통해 생성된 문맥 벡터, s_i는 디코더의 은닉 계층을 나타낸다.

제안 시스템은 위에서 설명한 주의 집중 순환 신경망을 이용하여 추상 요약을 수행한다. 신경망의 입력은 TextRank를 통해 일부 요약된 결과이며 출력으로 요약된 문장을 출력한다. 주의 집중 순환 신경망을 이용한 추상 요약의 예제는 아래 [표 2]와 같다.

표 2. 주의 집중 순환 신경망을 이용한 추상 요약 예

입력	이에 한결같이 "모른다"는 답변만 늘어놓은 우전 수석에게 손혜원 더불어민주당 의원은 일침을 가했다. 이처럼 참고인 노 전 부장을 비롯한 다른 증인들과 우 전 수석이 전혀 다른 주장을 펼치면서 누군가는 위증을 하고 있다는 의혹이 커지고 있는 상황이다.
출력	우병우 청와대 민정수석이 '최순실 게이트' 진상규명을 위해 자신의 집 앞에 섰다 .
정답 요약	최순실 국정농단 사태 진실규명을 위한 청문회에 증인으로 출석한 우병우 청와대 전 민정수석은 '최순실을 아느냐'는 질문에 "모른다"로 일관했다.

[표 2]에서 출력은 추상 요약의 결과이다. 정답 요약은 모델 학습 시에 정답으로 사용한 문장으로, 기자가 해당 기사에 작성한 코멘트이다. 실제 입력 및 출력은 형태소 단위로 이루어지며, [표 2]에서는 가독성을 위해 일반 문장으로 표시하였다.

4. 실험

4.1 실험 데이터

본 논문에서는 실험을 위해 인사이트 뉴스 기사 21,072 문서를 수집하였다. 전체 데이터 중 19,999 문서를 학습에 사용하였으며, 1,073 문서를 평가에 사용하였다. 학습을 위한 정답으로는 기사를 작성한 기자가 남긴 코멘트를 사용하였다. 요약 결과를 평가하기 위한 지표로는 ROUGE[10]를 사용하였으며, 그 중에서도 ROUGE-1, ROUGE-2, ROUGE-L을 사용하였다.

4.2 성능

제안 시스템의 성능은 아래 [표 3]과 같다.

표 3. 입력에 따른 요약 성능 비교

입력 길이	ROUGE-1	ROUGE-2	ROUGE-L
100%	0.2961	0.1453	0.3348
70%	0.2901	0.1395	0.3317
30%	0.2802	0.1294	0.3254

[표 3]에서 성능은 각 ROUGE의 F1-점수이며, 입력 길이는 추상 요약에 사용한 입력의 길이를 나타낸다. 입력 길이 100%는 추출 요약을 거치지 않은 원본 문서를 나타내며, 입력 길이 70%의 경우 TextRank를 통해 원본 문서 대비 70% 수준으로 문서를 요약한 것을 나타낸다.

성능 평가 결과, 중요한 일부 문장을 선별하여 입력할

경우 성능이 향상될 것이라는 예상과 달리 입력의 길이를 줄이더라도 추상 요약의 성능이 향상되지는 않았다. 그러나 원본 문서 대비 30% 수준까지 문서를 축약하여 입력하였음에도 불구하고 성능 감소가 크지 않은 것을 확인할 수 있었다.

5. 결론 및 향후 연구

본 논문에서는 추출 요약과 추상 요약의 결합을 통해 두 가지 방법론이 가진 단점을 해소함으로써 문서 요약 성능의 개선을 시도하였다. 그러나 원본 문서를 의미 있는 방법으로 축약하더라도 추상 요약의 성능이 향상되지는 않았으며, 출력된 문장이 요약으로서 충분하지 않은 경우도 많았다. 분석 결과, 이는 문서 요약 분야에서 현재의 신경망 모델 및 데이터가 가진 한계로 보인다. 그러나 입력되는 문서의 크기를 크게 줄이더라도 요약 성능이 크게 하락되지는 않았다는 점으로 미루어 볼 때, 충분한 데이터를 확보하고 모델 구조를 개선한다면 의미 있는 결과를 보일 수 있을 것으로 생각된다.

향후 연구로 TextRank외에 다양한 방법을 통해 추출 요약 성능을 향상시킬 것이며, 더 많은 데이터를 수집하고, 개선된 형태의 신경망을 활용하는 것으로 추상 요약을 성능을 향상시킬 예정이다.

감사의 글

이 논문은 2016년도 정부(과학기술정보통신부)의 재원으로 정보통신기술진흥센터의 지원을 받아 수행된 연구임 (No.R-20160906-004163, 빅데이터 자동 태깅 및 태그 기반 DaaS 시스템 개발)

참고문헌

[1] K. Knight and D. Marcu, Summarization beyond sentence extraction: A probabilistic approach to sentence compression, *Artificial Intelligence*, 139(1), pp. 91-107, 2002.

[2] R. Mihalcea, Language independent extractive summarization, *In Proceedings of the ACL 2005 on Interactive poster and demonstration sessions,* pp. 49-52, 2005.

[3] J. Tan, X. Wan and J. Xiao, Abstractive Document Summarization with a Graph-Based Attentional Neural Model. *In Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics,* Vol. 1, pp. 1171-1181, 2017.

[4] P. Nema, M. Khapra, A. Laha and B. Ravindran, Diversity driven Attention Model for Query-based Abstractive Summarization. *arXiv preprint arXiv:1704.08300,* 2017.

[5] Q. Zhou, N. Yang, F. Wei and M. Zhou, Selective Encoding for Abstractive Sentence Summarization. *arXiv preprint arXiv:1704.07073,* 2017.

[6] L. Page, S. Brin, R. Motwani and T. Winograd, The PageRank citation ranking: Bringing order to the web. Stanford InfoLab. 1999.

[7] R. Mihalcea and P. Tarau, TextRank: Bringing Order into Text. *In EMNLP* Vol. 4, pp. 404-411, 2004.

[8] 최경호, 이창기. 복사 방법론과 입력 추가 구조를 이용한 End-to-End 한국어 문서요약. 정보과학회논문지, 44(5), pp. 503-509, 2017.

[9] D. Bahdanau, K. Cho and Y. Bengio, Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473,* 2014.

[10] Lin, Chin-Yew, "ROUGE: a Package for Automatic Evaluation of Summaries," *Proceedings of the Workshop on Text Summarization Branches Out*, 2004.

대규모 분류 체계에서 계층적 샘플링을 활용한 문서의 분류

홍성모○, 장헌석, 강인호

Naver Corporation
sungmo.hong@navercorp.com, heonseok.jang@navercorp.com, once.ihkang@navercorp.com

Classification using Hierarchical Sampling in Large Classification System

SungMo Hong○, HeonSeok Jang, Inho Kang
Naver Corporation

요 약

대규모 분류체계를 사용하는 경우, 기존 방법의 딥 러닝으로는 분류 정확도가 현저히 떨어진다. 이를 해결하기 위해 계층 구조를 활용한 네거티브 샘플링 방법을 제안한다. 학습 문서가 속한 카테고리의 상위 카테고리와 일정부분 겹치는 범위에서 네거티브 샘플을 선택하면, 하나의 큰 문제를 다수개의 하위 문제로 쪼개서 해결하는 학습 효과가 있다. 소규모 분류 체계와 대규모 분류체계 각각에서 샘플링 전략을 차용하였을 때를 비교한 결과, 대규모에서 효과가 좋았으며 그 때의 정확도가 150배 이상 차이가 나는 것을 보였다.

주제어: 문서 분류, 대규모 분류체계, 계층적 샘플링

1. 서론

데이터의 분류는 자료를 정보화하는 효율적인 수단이다. 하지만 데이터가 많아지면서 자연스럽게 데이터 분류 체계의 규모도 함께 늘어났다. 결국 증가한 데이터 개수 그리고 복잡한 분류체계로 인해 사람이 직접 데이터를 분류하기가 어려워졌다. 이를 해결하기 위해 토픽 모델링(Topic Modeling)에 대한 연구가 활발하게 진행되었다. 나이브 베이즈(Naïve Bayes)를 이용하여 글을 분류하거나, 잠재 디리클레 할당(Latent Dirichlet Allocation)을 사용하여 단어 분포를 가지고 문서의 주제를 예측하였다[1,2].

최근에는 딥 러닝을 이용하여 문서를 해석하고 분류하는 기술도 연구되었다. Yoon은 컨볼루션 신경망(Convolution Neural Network)를 사용하여 자연어 텍스트를 분류하는 방법을 연구하였다[3]. 이 방법은 소규모 분류체계에서 정확도가 높지만, 카테고리 개수가 많아질수록 정확도가 낮아진다. 이와는 달리 데이터의 유형에 따라 대규모 분류체계에서도 잘 동작하는 경우도 있다. 실제로 음성인식과 이미지 인식의 분야에서는 높은 정확도를 보여주는 모델 연구가 진행되었다[4,5].

자연어 처리 분야에서는 입력 문장을 다수개의 유형으로 분류한 연구결과가 있으나, 실험에 사용한 최대 분류 카테고리 개수는 6개이다[3]. 하지만 필요에 따라 대규모 분류체계를 사용할 수 있어야 하는데, 아직 이에 대한 연구는 많이 되지 않았다.

본 논문에서는 계층이 있는 대규모 분류체계 내에서 텍스트를 분류하고자 할 때, 학습에 사용하는 효과적인 샘플링 방법을 제시한다. 카테고리가 많은 특성 때문에 모델이 학습에 실패하는 현상을 극복하기 위해, 계층적 구조를 활용한 네거티브 샘플링(Negative Sampling)을 사용하여 학습을 국지적으로 수행하였다.

2. 관련 연구

잠재 디리클레 할당은 기 분류되어있던 문서의 단어 분포를 이용하여, 새로운 문서의 단어 분포를 보고 문서의 주제를 찾는다[1]. 하지만 이 방법은 단어 주머니(Bag-of-words) 방식이기 때문에, 단어가 속한 문맥을 정확하게 파악하는 데에는 어려움이 있다.

딥 러닝 기법인 컨볼루션 신경망과 순환 신경망(Recurrent Neural Network)은 단어가 속한 문맥을 포함하여 의미를 읽어낼 수 있다. Word2Vec은 흔히 볼 수 있는 텍스트를 학습 데이터로 사용하여 단어의 의미를 문장의 문맥에서 파악하고 임베딩한다[6]. 기계학습으로 텍스트의 의미를 파악하고 분류하는 연구도 진행되었다. Yoon은 입력 문장을 미리 정의한 카테고리들 중 하나로 할당하는 모델을 제시하였다. 이 모델의 구조는 Word2Vec, 컨볼루션 신경망, 최대값 풀링(Max Pooling), 그리고 완전 연결 신경망(Fully Connected Neural Network)을 함께 사용하여 만들었다. 하지만 본 논문의 실험 결과, 분류체계의 규모가 커지면 학습이 실패하여 분류 정확도가 줄어드는 것을 확인하였다.

하지만 음성 분야와 이미지 분야는 대규모 분류체계에서도 정확도가 높게 분류하고 있다. 음성인식 분야에서는 음성을 입력으로 순환 신경망을 사용하여 62개의 음소 카테고리 중 하나로 분류를 하였다[4]. 또한 이미지 분야에서는 컨볼루션 신경망을 사용하여 1000개의 카테고리 중 하나로 분류를 하였다[5].

본 논문은 Yoon이 제안한 모델의 구조를 변형하여, 완전 연결 신경망의 결과로 문서 벡터가 나오도록 학습한다. 문서 벡터와 가까운 카테고리 벡터를 주어진 문서의 카테고리로 할당할 수 있게끔 카테고리 벡터를 학습할 때, 카테고리 규모가 커서 학습이 되지 않는 문제점의

해결책을 제시한다.

3. 문서 분류를 위한 분류명 벡터 임베딩

동일한 벡터 공간에 분류명을 의미하는 벡터와 문서를 의미하는 벡터를 같이 표시할 수 있다면, 문서 벡터와 가까운 분류명 벡터를 찾아 문서의 분류명으로 예측하는 것이 가능하다. Zeynep은 벡터 공간에 입력 이미지에 대응하는 벡터와 이미지 레이블 벡터를 동시에 학습하여 주어진 이미지의 레이블을 예측하였다[7]. 이와 같이, 같은 공간에 다른 성격의 벡터를 함께 투사하는 것이 가능하다. 이를 텍스트 분류에 응용하여 문서 벡터와 분류명 벡터를 같은 공간에 투영하고, 최근접 이웃 탐색 알고리즘(K Nearest Neighbor)을 사용하여 주어진 문서를 분류하였다. 이처럼 벡터를 사용하는 이유는 계층적 분류체계에서 카테고리간의 거리가 다 똑같다고 규정할 수 없기 때문이다. 예를 들면, 동물 카테고리는 딥 러닝 카테고리보다 식물 카테고리와 거리가 가깝다고 할 수 있다. 이와 같은 이유로 각 카테고리를 벡터를 표현하여 카테고리간의 거리를 모델이 학습할 수 있게 구성한다.

Yoon의 모델은 점수 모델로 각 카테고리에 속할 상대적 확률을 출력한다. 학습 시에는 출력이 정답 카테고리의 확률만 1이 나오도록 확률적 경사 하강법(Stochastic Gradient Descent)으로 조절한다. 반면 본 논문에서 제시하는 모델인 벡터 모델은 문서를 의미하는 벡터를 출력한다. 문서 벡터와 정답 카테고리 벡터와의 유사도는 1이 되도록 그리고 오답 카테고리 벡터와의 유사도는 0이 되도록 학습한다. 본 논문에서의 유사도는 두 벡터 요소간 곱을 모두 합한 값을 사용하였다.

그림 1은 점수 모델과 벡터 모델의 학습과정을 설명하

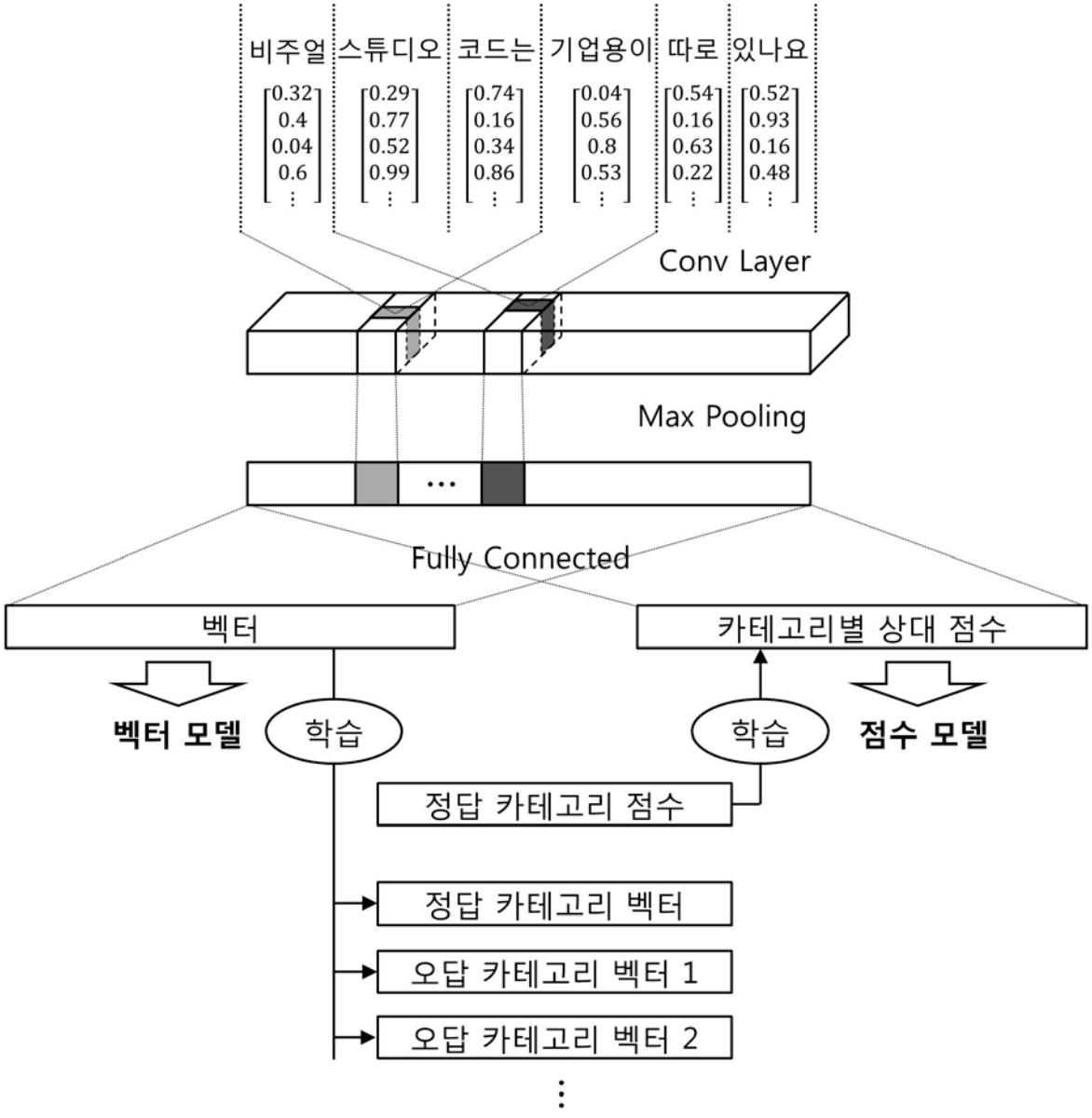


그림 1 벡터 모델과 점수 모델의 차이점 비교

고 그 차이를 보여준다. 텍스트를 컨볼루션 신경망 계층과 맥스 풀링(max pooling) 계층에 통과시키는 부분까지 동일하나, 완전 연결 계층을 거친 후의 결과를 해석하는 방식과 학습하는 방식이 다르다. 특히 벡터 모델이 문서 벡터의 결과를 사용해 카테고리 벡터를 학습시키는 점과, 다수개의 오답 카테고리도 학습에 영향을 받는다.

4. 대규모 분류체계에서의 학습 방법

분류체계의 규모가 큰 경우, 점수 모델을 사용해도 신경망의 가중치 값이 올바른 최적값에 도달하지 못하여 예측의 정확도가 현저히 낮아진다. 그 이유는 분류체계의 규모가 카테고리 벡터의 위치를 학습하기에 상대적으로 크기 때문이다.

이런 현상을 해결하기 위해 분류체계의 규모가 크면 분류가 계층적 구조를 차용하는 점을 이용한다. 계층적 구조의 특성에 따라 같은 분류명을 공유하는 하위 분류체계 또한 독립적인 분류체계의 특징을 가지고 있다. 그러므로 하위 분류 내에서 분류명 벡터 위치를 학습하는 하위문제(subproblem)들로 쪼개서 해결하면 전체 문제를 해결할 수 있다. 하위 문제를 효과적으로 해결하는 방법은 네거티브 샘플을 전략적으로 선택하는 것이다. 정답 카테고리의 상위 카테고리 중 하나의 카테고리를 대분류로 지정하여, 대분류 이하의 작은 분류체계를 학습하는 하위 문제로 재정의할 수 있다. 학습 데이터 한 개마다 다수개의 네거티브 샘플을 선택할 수 있기 때문에, 대분류를 다양하게 바꿔가며 고르게 표집하는 것이 효과적이다. 이와 같이 고른 샘플링 전략을 세우는 이유는 하나의 네거티브 샘플로 하위 문제를 해결함과 동시에 상위/하위 분류체계와도 연관성을 유지할 수 있기 때문이다.

그림 2는 샘플링 전략을 보여주는 구체적인 예시이다. 분류명 “라”에 속한 문서를 학습하고자 할 때, 각 숫자로 묶인 분류에서 임의로 하나의 분류명씩 네거티브 샘플로 선택하면 된다. 자세히 말해 학습 데이터의 분류가 상위 분류명부터 가-나-다-라인 경우, 첫 번째 샘플은 가-나-다에 속하면서 가-나-다-라에 속하지 않는 범위인 1번 그룹에서, 두 번째 샘플은 가-나에 속하면서 가-나-다의 모든 하위 분류에 속하지 않는 범위인 2번 그룹에서, 세 번째 샘플은 가에 속하면서 가-나의 모든 하위 분류에 속하지 않는 범위인 3번 그룹에서, 마지막 샘플은 가의 모든 하위 분류에 속하지 않는 범위에서 추출하는 것이다. 네거티브 샘플을 이용한 학습 방법은 이전 절에서 논의한 방법과 동일하다.

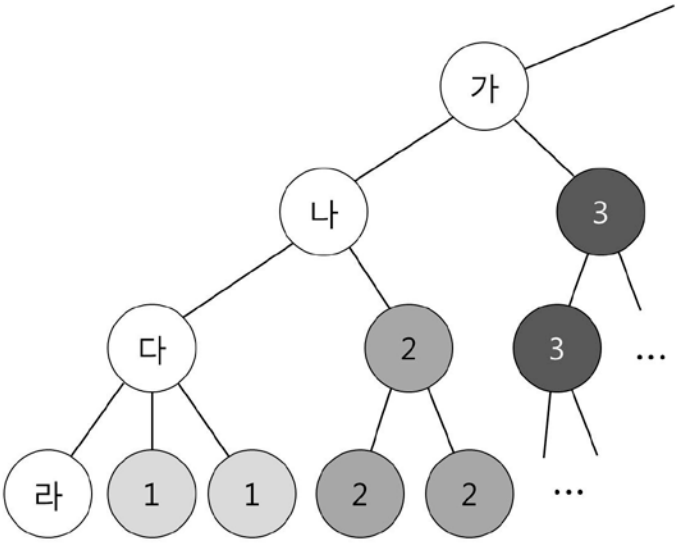


그림 2 계층적 구조에서의 네거티브 샘플링

5. 실험

5.1. 실험 데이터

네이버 지식인은 사용자 간 질의응답 플랫폼이다. 질문을 등록하는 분류체계의 규모가 크고 계층적 구조를 가지고 있다. 사용자가 지식인에 작성한 문서 중 800,852개를 학습에 사용하고, 19,624개를 개발용으로 사용, 39,249개를 평가에 사용하였다. 질문 데이터의 최상위 카테고리만 추출하여 13개 카테고리 규모의 소규모 분류체계를 만들고, 세번째 깊이와 그 상위 카테고리를 모아서 총 798개의 대규모 분류체계를 만들었다. 표 1은 분류체계 별 데이터의 특징을 보여준다.

표 1 분류체계별 데이터 특성

	소규모	대규모
카테고리 개수	13	798
단일 카테고리 내 최대 문서 수	122,261	10,881
단일 카테고리 내 최소 문서 수	4581	1
카테고리 내 평균 문서 수	61,604	1,003.5
표준편차	36954.5	1343.2
중앙값	58,920	836

5.2. 평가

평가 데이터의 문서를 사용, 가장 점수가 높은 분류명 N개에서 사용자가 선택한 분류명이 있는지 확인한다. 사용자가 선택한 분류명이 있으면 해당 예측은 정답으로, 없으면 오답으로 표시한다.

5.2.1. 소규모 분류체계

데이터에 등록된 분류명을 최상위 분류로 압축하면 총 13개의 분류로 모든 문서를 나눌 수 있다.

표 2와 그림 3은 소규모 분류체계에서 문서를 분류하였을 때, 상위 N개의 추천에서 적합한 분류명을 찾을 정답률이다. 분류명 수가 적을 때의 정답률은 벡터 모델에 비해 점수 모델이 미세하지만 소폭 나은 성능을 보였다. 이는 비계층적 형태의 분류체계이기 때문에 샘플링 전략

이 아무런 의미가 없었고, 카테고리 벡터간의 거리가 정확도에 영향을 미칠 만큼 다양하지 않기 때문으로 해석할 수 있다.

표 2 소규모 분류체계에서 N개 추천시 정답률

	벡터 모델	점수 모델
1개 추천	0.145	0.162
2개 추천	0.243	0.306
3개 추천	0.329	0.430

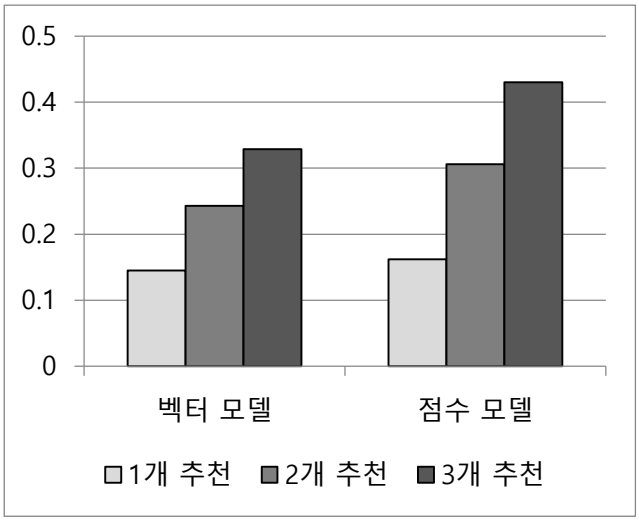


그림 4 소규모 분류체계에서 N개 추천시 정답률

5.2.2. 대규모 분류체계

데이터에 등록된 분류명을 깊이 3까지 압축하면 총 798개의 분류로 모든 문서를 나눌 수 있다. 마찬가지로 두 모델에서 점수가 높은 분류명 N개를 예측하고 정답률을 비교해 보았다.

표 3과 그림 4는 대규모 분류체계에서 문서를 분류하였을 때, 상위 N개의 추천에서 적합한 분류명을 찾을 정답률이다. 네거티브 샘플링(negative sampling)을 전체에서 무작위로 선발한 경우는 논문 제시 모델의 학습이 전혀 되지 않는 현상이 발견된다. 제시한 샘플링 원칙대로 선발할 경우, 논문 제시 모델의 성능은 좋아졌고 비교 모델의 성능은 나빠지는 경향을 보인다. 이는 벡터화 단계와 네거티브 샘플링 전략이 대규모 계층적 분류 체계 학습에 더 알맞다고 할 수 있다.

표 3 대규모 분류체계에서 N개 추천시 정답률

	벡터 모델	점수 모델
1개 추천	0.411	0.002
2개 추천	0.570	0.003
3개 추천	0.658	0.004

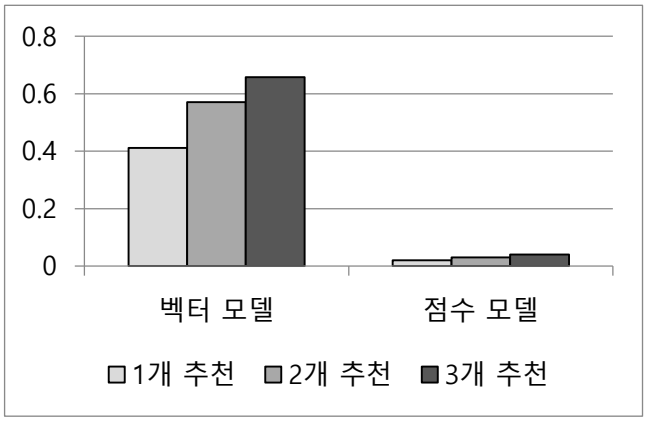


그림 3 대규모 분류체계에서 N개 추천시 정답률

표 4는 본 논문에서 제시한 모델로 일반 자연어를 분류했을 때, 잘 된 경우와 그렇지 않은 경우이다. 올바른 분류 판단 기준은 사용자의 질문과 사용자가 직접 등록한 카테고리를 비교하여 둘이 같은 경우에만 정답으로 판단한다. 첫 번째 오분석 예제의 경우 “ISP” 라는 단어를 정확하게 인지하지 못한 현상으로 보이며, 두 번째는 “몬츠”라는 의류 브랜드를 잘 학습하지 못한 현상으로 보인다.

표 4 분석과 오분석 예제

분석 예제	비주얼 스튜디오 코드는 기업용이 있나요? 회사 컴퓨터에 깔고 싶은데, 기업용이면 제한이 있어서..문의 드립니다.
	컴퓨터통신>프로그래밍
	리패키지?앨범 스밍 힘든건가요 1위하는게 어렵네여 ㅜ
	엔터테인먼트, 예술>음악>음악인
오분석 예제	isp 문자발송 관련건 질문 isp로 결제하게되면 카드명의 휴대폰으로 문자가 가나요?? 제 체크카드 명의가 부모님으로 되어있는데 문자 발송 가는건 별로 원하질 않아서욧....
	쇼핑>예약, 예매>여행상품
	몬츠 매장 ㅠㅠㅠ 몬츠 오프라인 매장 없나요? 작게 되있는 곳 말고 좀 크게 입점된 곳이요
	쇼핑>취미, 오락, 문구류>모형, 완구

6. 결론

분류체계의 규모가 크고 그 구조가 계층적일 때, 문서와 분류명을 임베딩하고 계층적 네거티브 샘플을 이용해 학습하여 적합한 분류명을 찾는 것이 일반적인 학습 방법에 비해 정확도가 150배 이상 높다. 반대로 분류 체계가 작고 단순한 경우에는 각 분류에 속한 문서의 주제가 방대해지기 때문에 하나의 카테고리 벡터로 표현하기 어렵다. 이러한 이유로 벡터 모델 방식은 대규모 분류 체계를 사용하는 경우에 오히려 성능이 낮아진다.

향후 사용자의 피드백을 받아, 기존의 학습된 벡터를 효과적으로 변경하는 방법에 대해 연구가 필요할 것으로 보인다.

참고문헌

[1] D. Blei, A. Ng, and M. Jordan. Latent dirichlet allocation. Journal of machine learning research 3. Jan pp.993-1022, 2003.

[2] S. B. Kim, K. S. Han, H. C. Rim, and S. H. Myaeng, Some effective techniques for naïve bayes classification, IEEE transactions on knowledge and data engineering, 18(11), pp.1457-1466, 2006.

[3] Yoon Kim. Convolutional neural networks for sentence classification. In Proceedings of EMNLP. 2014.

[4] K. He, X. Zhang, S. Ren, and J. Sun, Deep Residual Learning for Image Recognition, In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp.770-778, 2016.

[5] A. Graves, A. Abel-rahman Mohamed, and G. Hinton, Speech recognition with deep recurrent neural networks, IEEE international conference on acoustics, speech and signal processing, pp.6645-6649, 2013.

[6] Tomas Mikolov, et al. Efficient estimation of word representations in vector space. ICLR Workshop, 2013.

[7] Zeynep Akata, et al. Label-embedding for image classification. IEEE transactions on pattern analysis and machine intelligence. 38.7: 1425-1438, 2016.

CNN을 이용한 발화 주제 다중 분류

최경호 ○, 김경덕, 김용희, 강인호
Naver RND center, Clova Dialogue, Naver RND center, Clova NLP
{k.h.choi, kyungduk.kim, yong.hee.kim.0402, once.ihkang}@navercorp.com

Multi-labeled Domain Detection Using CNN

Kyoungho Choi ○, Kyungduk Kim, Yonghe Kim, Inho Kang
Naver RND center, Clova Dialogue, Naver RND center, Clova NLP

요 약

CNN(Convolutional Neural Network)을 이용하여 발화 주제 다중 분류 task를 multi-labeling 방법과, cluster 방법을 이용하여 수행하고, 각 방법론에 MSE(Mean Square Error), softmax cross-entropy, sigmoid cross-entropy를 적용하여 성능을 평가하였다. Network는 음절 단위로 tokenize하고, 품사정보를 각 token의 추가한 sequence와, Naver DB를 통하여 얻은 named entity 정보를 입력으로 사용한다. 실험결과 cluster 방법으로 문제를 변형하고, sigmoid를 output layer의 activation function으로 사용하고 cross entropy cost function을 이용하여 network를 학습시켰을 때 F1 0.9873으로 가장 좋은 성능을 보였다.

주제어: Multi-label classification, Domain detection

1. 서론

최근 딥러닝으로 인한 음성인식기술과 자연언어 처리 기술의 발달로 인해 가상 비서, 인공지능 스피커 등의 다양한 형태로 대화형 인터페이스 기술이 본격적으로 서비스에 적용되고 있다.

대화형 인터페이스의 핵심인 대화시스템의 목적은 별도의 정해진 명령어나, 정해진키워드 없이 자연언어 형태의 입력을 받아, 사용자의 요청을 이해하고, 해당 요청을 수행하거나, 사용자의 발화에 응답하는 것이다. 대화시스템은 자연언어 형태의 사용자 발화를 시스템이 이해할 수 있는 형태의 명령으로 번역하는 기술을 필요로 하며, 이를 자연언어 이해(Natural Langue Understanding)라 한다.

일반적으로 대화시스템을 이용하는 사용자의 발화는 주제에 따라 다른 어휘적, 언어적 이해를 필요로 할 수 있다. 예를 들어 동명이인이 발화에 등장하였을 경우 해당 발화의 주제를 확인 할 수 있다면, 실제 지칭하는 대상이 될 수 있은 인물들의 폭을 좁힐 수 있다. 때문에 발화의 주제를 찾아내는 domain detection을 자연언어 이해 과정 전반부에서 수행하고, 해당 task 정확도를 신뢰할 수준으로 높일 수 있다면, 대화시스템에 요구되는 자연언어 이해 과정의 정확성을 향상시킬 수 있다.

사용자의 발화는 여러 주제를 동시에 포괄할 수 있으며, 또한 주제가 모호하여 특정 주제라 확정하기 어려울 수 있다. 예를 들어 “내일은 어때?” 라는 발화가 주어졌을 때 해당 발화만으로 시스템이 “weather”, “schedule” 중에 주제를 확정하기 어렵다. 이러한 경우 한 발화를 하나의 주제로만 분류하지 않고, 여러 개의 주제로 다중 분류 할 수 있다.

그러나 전통적으로 연구되었던 CRF(Conditional Random Fields), SVM(Support Vector Machine)모델들은 다중 분류 task에 그대로 사용하기 어렵다[1]. 또한 모델의 입출력을 자유롭게 설계할 수 있는 deep neural network를 사용할 경우에도, 적절한 학습 방법과, cost function을 사용하지 않는다면, 시스템의 성능을 보장할 수 없다.

본 논문에서는 발화 주제 다중 분류 task를 CNN(Convolutional Neural Network)을 이용하여 수행하여 보고, 발화 주제 분류 task를 수행할 때 multi-label task를 다루기 위한 방법론과 cost function에 따른 성능을 비교한다.

2. 관련 연구

2.1 Domain detection

기존의 domain detection 연구는 단일 domain 분류를 중심으로 수행되었다. 또 발화를 연구의 목적으로 한 domain보다, 웹 상의 text를 대상으로 한 topic detection(topic modeling)이 주로 연구되어 왔다.

최근 연구 에서는 주제 간의 계층을 자동으로 구축하여, 주제 다중 분류 문제를 해결하기도 하였다[2].

2.2 Multi-label Classification

전통적으로 사용되는 기계학습 모델들은 입력 데이터를 하나의 class로 분류하는 모델이 일반적이다. 때문에 다중 분류 문제를 해결하기 위해서는 기존 기계학습 모델을 다중 분류를 가능하도록 수정하여 사용하거나, class들을 cluster로 만들어 모델이 단일 cluster로 분류하도

록 task를 수정하여 수행 할 수 있다. 전자의 경우, 다중 분류가 가능하도록 수정 될 수 있는 기계학습 모델이 많지 않을 뿐만 아니라, 해당 모델을 학습하는 전략에 있어, 분석과 계획이 필요하다. 후자의 경우, cluster가 일정하게 형성될 보장이 없는 task이거나, class들이 다양한 조합으로 cluster될 수 있는 경우, 모델이 필요한 계산 량이 커질 수 있다.

Neural network의 경우 모델의 output layer를 수정하여 multi-label 분류에 사용할 수 있다. 이때 cost function과, activation function, 학습 전략 등에 의해 모델의 성능이 크게 달라질 수 있다.

3. Convolutional neural network for domain detection

본 연구에서는 multi-label classification task에 적합한 방법과, 해당 방법과 함께 사용하였을 때 우수한 성능을 보이는 cost function을 확인하기 위하여, Convolutional Neural Network를 사용하였다.

3.1 Network

연구에 사용된 모델은 [3]에서 소개된 모델을 기반으로 하여 입력으로 추가적인 입력을 사용할 수 있도록 그림1과 같이 수정한 모델로, 2개의 projection layer를 이용하여, 두 종류의 토큰을 입력으로 사용한다. 하나는 lexicon이며 음절을 그대로 사용할 경우 음절 자체가 갖는 중의성을 줄이기 위하여, 음절 단위로 품사를 부착한 lexicon을 사용하였고, 다른 하나는 형태소 단위로 사전을 조회하여 얻은 type 정보를 첫번째 입력과 align 하고 BIO tag를 추가하여 만든 feature이다[4]. 이 feature는 lexicon만으로는 알 수 없는 외부 지식을 네트워크가 학습 할 수 있게 한다.

각 입력은 서로 다른 각각의 projection layer를 이용하여 vector representation형태로 만든다. 이때 feature 입력은 time-step 당 여러 개의 label이 입력으로 사용 될 수 있으며, 여러 개의 label이 입력으로 사용되는 경우 각 입력을 projection 하고, element-wise sum하여 vector representation을 만든다.

그림1은 “알루미늄 틀어줘”를 network의 입력으로 사용한 예시이다. 입력 받은 문장의 품사를 분석하여 “알/PROPER”과 같이 각 음절에 “/”을 경계로 부착한다. 음절과 품사가 부착된 형태의 token을 one-hot representation으로 하여 projection한다. 입력 문장을 각 DB에 조회하면, “알루미늄=[singer, song, metal]”, “루미=[singer]”, “미=[singer]”, “틀=[singer, song]”, “어=[song]”를 확인 할 수 있다. 이를 음절 단위로 분리하여, BIO tag를 추가하여, 각 lexicon에 일치하는 feature로 사용한다

두 vector representation을 concatenate하여, 3, 4, 5 개의 window를 갖는 convolution layer의 입력으로 사용한다. 세 convolution layer의 출력을 max-over-time polling 하고, concatenate 하여 dropout을 적용하고, output layer의 입력으로 사용한다. Output layer는 cost function 에 따른 성능평가를 위해 별도의 activation function을 사용하지 않는다.

각 cost function을 평가할 때 cluster 방법을 사용한 경우에는 Network의 출력 값을 argmax 하여 사용하였고, multi-label 방법은 threshold(0.5)를 넘긴 출력들만 사용하여 평가하였다.

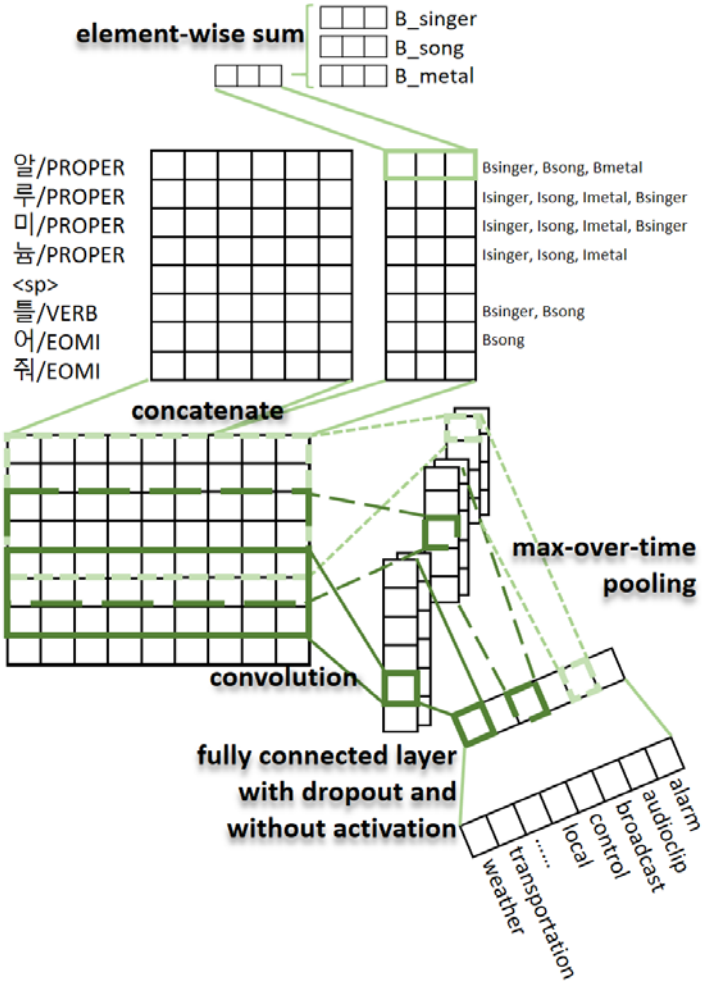


그림1. Model 구조

3.2 Cluster method

단 하나의 발화 만을 분석하여 주제를 추론할 때에 해당 발화의 주제를 명료하게 하나로 추론하기 어려울 수 있다. 이때 여러 개의 발화 주제를 동시에 해당 발화의 주제로 삼을 수 있다.

“매주 목요일 마다 알려줘”라는 발화가 있다면, 해당 발화의 주제는 해당 발화의 앞선 발화에 따라, alarm, control, reminder, memo모두 일 수 있다. 이때 이를 $cycletime 이라는 별도의 class를 만들어, 분류하는 방법을 cluster 방법이라 정의한다. Cluster 방법을 사용하였을 경우 앞서 그림1로 나타낸 network의 출력은 한 개한 class 만 가질 수 있도록 하며, 해당 network를 학습하고 평가할 때 사용할 데이터의 정답도 하나의 class 만 가질 수 있도록 작성한다.

Multi-label 방법론을 사용할 경우 그림1의 network의

출력은 임계값을 넘긴 여러 개의 class를 동시에 가질 수 있으며, 사용하는 데이터에서 발화 주제 또한 동시에 여러 개를 하나의 발화에 기록하여 사용한다.

3.3 Cost functions

본 연구에서는 총 세가지 cost function을 비교하여 보았다. Cross entropy cost function은 multi-class classification에 주로 쓰이는 함수로 다음 그림2의 수식으로 나타낼 수 있다. 그림2의 두 수식은 각각 앞서 소개한 network의 마지막 activation function을 포함하고 있으며, $\hat{y}$는 network의 출력을 뜻한다. L_{CE}는 network 마지막 layer의 출력($\hat{y}$)에 sigmoid 를 적용한 cross entropy cost이고, L_{SCE} 는 softmax 를 적용한 cross entropy cost이다. $\mathrm{sigmoid}(\hat{y})$ 는 각각의 output unit에 해당 class에 대한 0부터 1사이의 예측 값을 갖지만, $\mathrm{softmax}(\hat{y})$는 모든 class들 중 각 class가 예측 될 확률 값을 갖기 때문에($\sum_i \mathrm{softmax}(\hat{y}_i) = 1$), threshold를 이용한 multi-label task에 $\mathrm{L_{SCE}}(y, \hat{y})$는 사용 할 수 없다.

$$L_{\mathrm{CE}}(y, \hat{y}) = -\log\big(\mathrm{sigmoid}(\hat{y})\big)\, y - \log\big(1 - \mathrm{sigmoid}(\hat{y})\big)\,(1 - y)$$

$$\mathrm{L_{SCE}}(y, \hat{y}) = -\log\big(\mathrm{softmax}(\hat{y})\big)\, y - \log\big(1 - \mathrm{softmax}(\hat{y})\big)\,(1 - y)$$

그림2. Cross entropy cost functions

실험에 사용한 MSE(mean squared error)는 그림3의 수식으로 나타내었다.

$$\mathrm{L_{MSE}}(y, \hat{y}) = -(y - \mathrm{sigmoid}(\hat{y}))^2$$

그림3. Mean squared error

4. Data set

본 연구에서 사용한 말뭉치는 사내에서 다수의 연구원이 구축한 구어체의 발화 말뭉치로, 각 발화에 대해 적절한 도메인이 모두 기록되어 있는 multi-labeled 말뭉치이다. 총 14개의 도메인에 대해 33032 발화로 이루어져 있고, domain 별 발화의 개수는 다음 표1과 같다.

표1. Domain별 발화의 개수

Domain	Count	Domain	Count
alarm	2095	music	7720
audioclip	5720	news	5809
broadcast	4250	radio	5494
control	5907	reminder	593
local	1620	sports	2089
memo	848	transportation	2146
movie	2000	weather	4181

한 발화에 함께 표기된 domain들을 묶어 cluster로 만든 label별 발화의 개수는 다음 표2와 같다. 표 2에서 $media 는 music, radio, news, control, audioclip, broadcast 발화 주제를 묶어 표현한 cluster를 의미한다. $local은 local과 transportation을 동시에 표현한 cluster, $temp는 control과 weather, $cycletime은 alarm, control, reminder, memo, $domestic은 sport, news, transportation, local, weather 발화 주제를 함께 나타내는 cluster이다. 실험에 사용된 cluster들은 말뭉치 제작 단계에서 발견한 실생활에 사용 할 수 있다 판단한 발화들을 기준으로, 조합 가능한 다중 발화 주제를 산정하여 만들었다.

표2. Cluster별 발화의 개수

Domain	Count	Domain	Count
alarm	2000	reminder	498
audio clip	2000	sports	2000
broadcast	530	transportation	2000
control	2000	weather	4000
local	1474	$media	3720
memo	753	$local	57
movie	2000	$temp	92
music	4000	$cycletime	95
news	2000	$domestic	89
radio	1774		

5. 실험

입력 문장을 음절 단위로 나누고, 각 음절에 해당하는 품사 정보를 합쳐 하나의 token으로 사용하였다. Neural network의 입력으로는 token을 기준으로 별도의 pre-training을 수행하지 않은 100 차원의 word embedding과, Naver DB를 조회하여 얻은 type 정보를 16차원으로 embedding 하여 사용하였다. 데이터를 전처리 하였을 때 총 10회 미만으로 등장한 word token과, feature label은 unknown word로 치환하여 학습하였으며, 사용한 word token들의 사전 크기는 3274, feature label의 개수는 3313종이다.

Multi-label 방법과 cluster 방법에 대해 앞서 설명한 cost function들을 적용하여 실험하였다. Sigmoid-CE, MSE을 이용해서 양쪽 방법론의 성능을 측정하나, softmax-CE는 multi-label 방법에서는 사용할 수 없는 cost이므로 (3.2 cost function) Cluster 방법에 대해서만 성능을 측정하였다. 사용한 convolution layer의 window size는 각각 3, 4, 5 로 하였고, 각 convolution layer 의 filter 개수는 표3에 명시한 대로 64개와 128개씩 두어 평가하였다. CNN layer 출력에 0.5 확률의 dropout을 주었고, weight decay는 사용하지 않았다. 또한 batch size를 64로 고정하고, adam[5] 알고리즘을 이용하여 network를 학습하였으며, validation set을 바탕으로 learning rate decay를 사용하였다. 전체 발화를

8:1:1로 나누어 24644발화를 학습에 사용하고, 3219발화를 validation set으로 사용하였으며, 3219발화로 성능을 평가하였다.

시스템을 평가할 때에는 cluster 방법론을 사용하여 예측하더라도, 해당 cluster에 포함된 각 주제로 분리하여 평가하였다. 이때 하나의 발화에 대하여, 각 주제가 모두 일치할 경우에만 정답과 일치한다고 평가한 지표를, 표 3에 exact match 로 나타내었고, 한 발화에 대하여, 정답 주제들과, 예측 주제들을 xor 연산하여 합한 값을 전체 주제의 개수로 나누어 오류 비율을 나타낸 hamming error를 표3에서 Hamming으로 나타내었다. precision, recall, F1은 모두 macro 형식으로 측정하였다. 표3에서 epoch는 모델의 cost가 수렴하여 학습을 종료한 epoch을 의미한다.

Cluster 방법론을 사용한 모델을 평가할 때에는 각 cluster이 나타내는 발화주제들로 모두 치환하여 평가하였다. 가령 "아이유 노래 틀어줘"라는 발화를 $media로 분류하였을 때 해당 결과를 music, radio, news, control, audioclip, broadcast로 치환하여, 소개한 지표들로 평가하였다.

실험 결과 전반적으로 multi-label 방법을 사용 하였을 때 큰 성능 하락을 보였다. Cluster 방법을 사용하고, sigmoid를 output layer의 activation 함수로 사용, cross entropy 함수를 cost function으로 사용하였을 때 모든 metric에서 가장 우수한 score를 보였다.

6. 결론

실험에서 multi-label 방법과 cluster 방법론 간의 명료한 수준의 성능 차이를 발견하였다. 다만 현실 세계에서 실제로 입력으로 나타날 발화의 주제들이 cluster로 미리 정의되어 있지 않다면, cluster 방법론을 사용한 시스템은, 발화의 주제를 제대로 추론하지 못한다.

추후 기회가 된다면 모델이 cluster의 계층 관계를 학습할 수 없는 cluster방법론의 문제점을 개선하여, 발화의 주제를 여러 계층에 거쳐 표현하여 정의하고, 분류하는 방법인 hierarchy[2]를 이용하여 발화 주제 다중 분류 연구를 진행하고자 한다.

참고문헌

[1] Grigorios Tsoumakas, Ioannis Katakis, "Multi-Label Classification: An Overview" International Journal of Data Warehousing and Mining 3.3, 2006.

[2] Seonghan Ryu, Jaiyoun Song, Sangjun Koo, Soonchoul Kwon, Gary Geunbae Lee, "Detection Multiple Domain from User's Utterance in Spoken Dialog System", Natural Language Dialog Systems and Intelligent Assistants. Pp. 101-111, 2015.

[3] Kim, Yoon. "Convolutional neural networks for sentence classification." arXiv preprint arXiv:1408.5882, 2014.

[4] A. Borthwick, J. Sterling, E. Agichtein, and R. Grishman. "Exploiting diverse knowledge sources via maximum entropy in named entity recognition", In Sixth Workshop on Very Large Corpora New Brunswick, New Jersey. Association for Computational Linguistics., 1998.

[5] Kingma, D. P., & Ba, J. L. "Adam: a Method for Stochastic Optimization", International Conference on Learning Representations, 1-13, 2015.

표3. cost function에 따른 평가

	Filter	Multi-label						Cluster					
		Exact match	Hamming	Precision	Recall	F1	Epoch	Exact match	Hamming	Precision	Recall	F1	Epoch
Simoid Cross Entropy	64	88.78	0.040008	0.92513	0.8952	0.8999	39	98.1671	0.032805	0.9853	0.9862	0.9843	43
	128	**88.97**	**0.039342**	**0.93507**	**0.8985**	**0.9041**	32	**98.5089**	**0.031656**	**0.9887**	**0.9884**	**0.9873**	27
Softmax Cross Entropy	64							98.1671	0.038283	0.9847	0.9847	0.9835	26
	128							98.2914	0.035663	0.9863	0.9843	0.9853	27
MSE	64	88.78	0.041051	0.9089	0.8925	0.8952	28	97.0177	0.062545	0.98	0.9772	0.9756	34
	128	88.78	0.039875	0.9258	0.8953	0.9000	29	97.3284	0.056539	0.9812	0.9785	0.9777	**14**

단어의 위치정보를 이용한 Word Embedding

황현선[O,†], 이창기[†], 장현기[††], 강동호[††]

강원대학교[†], SK 주식회사 C&C[††]

{hhs4322, leeck}@kangwon.ac.kr, hktopx77@gmail.com, eastsky21@sk.com

Word Embedding using word position information

Hyunsun Hwang[O,†], Changki Lee[†], HyunKi Jang[††], Dongho Kang[††]

Kangwon National University[†], SK holdings C&C[††]

요 약

자연어처리에 딥 러닝을 적용하기 위해 사용되는 Word embedding은 단어를 벡터 공간상에 표현하는 것으로 차원축소 효과와 더불어 유사한 의미의 단어는 유사한 벡터 값을 갖는다는 장점이 있다. 이러한 word embedding은 대용량 코퍼스를 학습해야 좋은 성능을 얻을 수 있기 때문에 기존에 많이 사용되던 word2vec 모델은 대용량 코퍼스 학습을 위해 모델을 단순화 하여 주로 단어의 등장 비율에 중점적으로 맞추어 학습하게 되어 단어의 위치 정보를 이용하지 않는다는 단점이 있다. 본 논문에서는 기존의 word embedding 학습 모델을 단어의 위치정보를 이용하여 학습 할 수 있도록 수정하였다. 실험 결과 단어의 위치정보를 이용하여 word embedding을 학습 하였을 경우 word-analogy의 syntactic 성능이 크게 향상되며 어순이 바뀔 수 있는 한국어에서 특히 큰 효과를 보였다.

주제어: Word Embedding, Word2vec, GLoVe

1. 서론

자연어처리를 위한 기존의 기계학습 모델에서는 단어를 One-hot의 형태로 표현하여 차원의 수가 많아져서 딥 러닝(Deep Learning)에 적용하기 어렵다는 문제점이 있었으나 word embedding을 이용하여 저 차원에서 단어를 표현하며, 딥 러닝의 사전학습 효과를 얻을 수 있게 되었다[1,2].

Word embedding은 단어를 벡터 공간상에 표현하는 것으로 One-hot의 표현 보다 저 차원으로 단어를 표현하며, 기존의 방법으로는 어려웠던 단어 유사도를 코사인 유사도로 쉽게 구할 수 있다는 장점이 있다. 이러한 word embedding을 구할 때에는 대용량 코퍼스로부터 NNLM(Neural Network Language Model)[1]을 학습시켜 얻게 된다. 그러나 이러한 NNLM 모델은 복잡하고 속도가 느려 대용량 코퍼스를 학습하는 데에 많은 시간이 소모된다는 단점이 있어, word embedding을 빠르게 학습 시키는 다양한 모델들이 연구되었다. 본 논문은 이러한 word embedding 학습 모델들 중에서 word2vec[3]의 CBOW(Continuous Bag-Of-Words) 모델을 단어의 위치정보를 볼 수 있게 개선한 모델을 제안한다. 실험 결과 기존의 word2vec 모델들 보다 syntactic 성능이 향상되는 것을 볼 수 있었다.

2. 관련 연구

Word embedding은 대용량 코퍼스를 이용하여 문장에서 사람이 사용하는 단어의 패턴을 학습하게 된다. 이에 따라 word embedding의 성능은 학습데이터의 양이 중요하게 되는데, 기존의 NNLM 모델로 대용량 코퍼스를 학습하기에는 많은 시간이 소모됨에 따라 다양한 word embedding 학습 모델들이 연구되었다.

2.1 Word2vec

Word2vec[3]은 기존의 NNLM 모델을 단순화 시키는 것과 동시에 비슷하게 사용되는 단어들은 벡터 공간상에서도 유사하게 표현되도록 학습하는 모델이다. 그림 1은 word2vec의 CBOW 모델과 skip-gram 모델을 나타낸 그림이다. CBOW 모델은 기존의 NNLM 모델에서 hidden layer를 제거하였고 projection layer에서 주위 단어 벡터들(그림1에서 w(t-2), w(t-1), w(t+1), w(t+2))을 단순히 element wise sum을 한 뒤 단순 신경망을 통하여 단어 w(t)를 예측하게 된다. Skip-gram 모델은 이와 반대로 단어 벡터 w(t)를 이용하여 주위 단어들(그림1에서 w(t-2), w(t-1), w(t+1), w(t+2))을 예측하게 된다. Word2vec은 이러한 방법을 통해 학습속도가 빠르며 문장에서 비슷하게 사용되는 단어들은 벡터 공간상에서도 유사하게 표현된다는 장점이 있으나, 문장에서의 단어 위치정보를 이용할 수 없다는 단점이 있다.

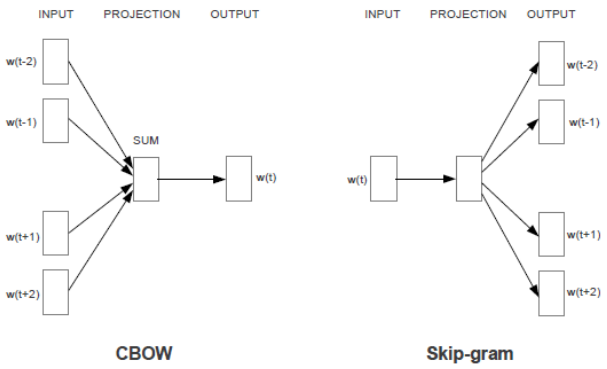


그림 1. Word2vec의 모델[3]

2.2 GLoVe

GloVe[4]는 word2vec과 달리 전체 학습데이터에서 등

장하는 단어들의 통계 정보를 활용하는 모델이다. GLoVe는 word embedding 학습 전에 학습데이터에서 각 단어들끼리의 한 문장에서 동시에 등장하는 확률들을 계산하며 그 예시는 표 1과 같다. 이때 단어의 수가 V개이면 해당 확률 표는 (V x V)의 크기를 가지게 된다. 이후 각각의 단어 벡터들은 서로 다른 단어에 대한 벡터 유사도를 표 1과 같은 동시 등장 확률과 비슷한 분포가 되도록 학습하게 된다. 이는 전체 학습데이터의 통계 정보를 가지고 학습하여 빠른 학습속도를 가지며 높은 성능을 낼 수 있다는 장점이 있으나, 학습 시 많은 메모리가 필요하다는 단점이 있다.

표 1. 한 문장에서 동시에 등장하는 단어들의 확률 예시

Probability and Ratio	$k = solid$	$k = gas$	$k = water$
$P(k\|ice)$	1.9×10^{-4}	6.6×10^{-5}	3.0×10^{-3}
$P(k\|steam)$	2.2×10^{-5}	7.8×10^{-4}	2.2×10^{-3}
$P(k\|ice)/P(k\|steam)$	8.9	8.5×10^{-2}	1.36

3. 단어의 위치정보를 이용한 Word Embedding

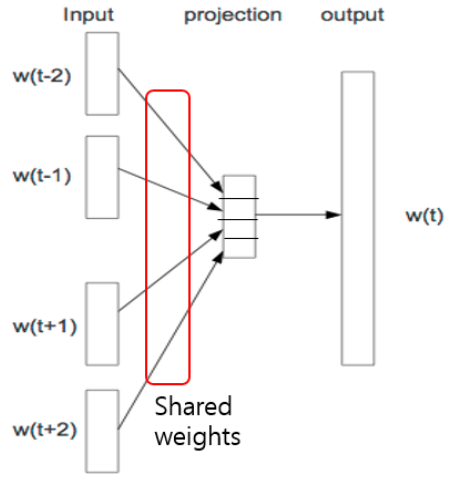


그림 2. 단어의 위치정보를 이용하는 word embedding 학습 모델

Word2vec의 CBOW 모델은 기존의 NNLM 모델을 단순화시켜 학습속도가 빠르고, 단어를 벡터 공간상에 효과적으로 표현할 수 있었다. 그러나 이 모델은 입력 단어 벡터들을 element wise sum을 하여 주위 단어들의 위치정보를 볼 수 없다는 단점이 있다. 이는 항상 어순이 고정되어있는 영어와 같은 언어에서는 영향이 적으나 한국어 같은 어순이 바뀔 수 있는 언어에 대해서는 문제가 생길 수 있다. 본 논문에서는 word2vec의 CBOW 모델을 단어의 위치정보를 볼 수 있게 적용한 모델을 제안한다.

그림 2는 본 논문에서 제안하는 단어의 위치정보를 이용하는 word embedding 학습 모델이다. 그림 1의 CBOW 모델에서 각 단어 벡터들을 element wise sum 하는 부분을 concatenate 하도록 수정하여, 입력으로 들어가는 주위 단어 벡터들(그림2에서 w(t-2), w(t-1), w(t+1), w(t+2))의 순서가 바뀌면 다른 입력으로 인식하게 만들어 문장에서의 해당 단어들의 위치정보를 볼 수 있게 만드는 효과가 있다. 그러나 이러한 방법은 CBOW 모델에 비하여 모델이 복잡해지는 효과가 있어 학습속도가 느려지게 된다.

4. 실험 및 결과

capital-common-countries
그리스/nnp 아테네/nnp 이라크/nnp 바그다드/nnp
그리스/nnp 아테네/nnp 태국/nnp 방콕/nnp
그리스/nnp 아테네/nnp 중국/nnp 베이징/nnp

capital-world
나이지리아/nnp 아부자/nng 가나/nnp 아크라/nnp
나이지리아/nnp 아부자/nng 알제리/nnp 알제/nnp
나이지리아/nnp 아부자/nng 요르단/nnp 암만/nnp

currency
알제리/nnp 디나르/nnp 앙골라/nnp 콴자/nng
알제리/nnp 디나르/nnp 아르헨티나/nnp 페소/nng
알제리/nnp 디나르/nnp 아르메니아/nnp 드램/nnp

city-in-state(korea)
수원시/nnp 경기도/nnp 화성시/nnp 경기도/nnp
수원시/nnp 경기도/nnp 성남시/nnp 경기도/nnp
수원시/nnp 경기도/nnp 광명시/nnp 경기도/nnp

그림 3. 한국어 word-analogy 평가데이터 예시

본 논문에서 제안한 단어의 위치정보를 이용한 word embedding 학습 모델과 다른 word embedding 학습 모델들과의 비교를 위해 한국어 데이터와 영어 데이터로 실험을 하였다. 한국어 학습 데이터는 10년치 금융 도메인 뉴스 기사를 크롤링 하여 형태소 분석기[5]를 사용하여 형태소 분석을 한 뒤 사용하였다. 영어 학습 데이터의 경우 Reuters에서 10년치 금융 도메인 뉴스 기사를 구매하여 NLTK python 패키지[6]의 토큰 분리기를 사용하여 토큰 분리 후 사용하였다. 이때 데이터 크기에 따른 성능 비교를 위해 영어 학습 데이터의 경우 소규모 데이터와 대규모 데이터로 나누어 실험을 하였다. 학습 파라미터의 경우 window size는 3으로, learning rate는 0.025으로, 학습 반복 횟수는 5로 고정하였다. GLoVe 모델은 다른 word embedding 학습 모델들과는 형태가 많이 달라 learning rate는 0.05으로, 학습 반복 횟수는 10으로 사용하였으며 word embedding의 차원은 50, 100, 200, 400으로 실험하였다. 성능 평가에는 word2vec의 word-analogy Top-1 accuracy를 사용하였으며, 한국어 word embedding 성능 평가의 경우, 영어 평가 데이터를 번역한 것과 한국어에 맞게 수정한 데이터를 사용하였으며 예시는 그림 3과 같다. 각각의 데이터 별 word embedding의 단어사전은 학습 모델들 모두 같으며, word-analogy의 threshold는 한국어는 100,000, 영어는 50,000을 사용하였다.

표 2는 word embedding 학습 모델 별 각각의 학습데이터에 대한 word-analogy 성능 표이다. 한국어 학습 데이터는 4억1900만 형태소 가량 되며, 영어 학습 데이터의 경우 각각 1억 8000만 토큰, 20억 9000만 토큰 가량이다. 실험 결과 본 논문에서 제안한 단어의 위치정보를 이용한 word embedding 학습 모델이 semantic 성능의 경우 CBOW 모델과 비슷하거나 낮았지만 syntactic 성능은

CBOW, skip-gram, GLoVe 모델보다 높은 것을 확인 할 수 있었다. 이는 단어의 위치정보를 추가 함에 따라 생긴 효과로 볼 수 있으며, 특히 영어에 비해 한국어의 성능이 크게 오른 것을 볼 수 있다. GLoVe 모델의 경우 전체 학습 데이터에 대한 통계 정보를 이용하여 semantic 성능이 높은 것을 볼 수 있었으며 단순히 단어 출현 빈도에 따른 학습으로 syntactic 성능이 낮은 것을 볼 수 있었다.

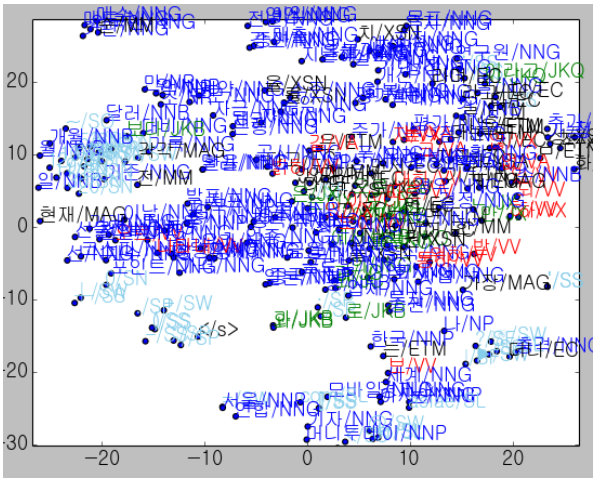


그림 4. CBOW로 학습한 한국어 word embedding

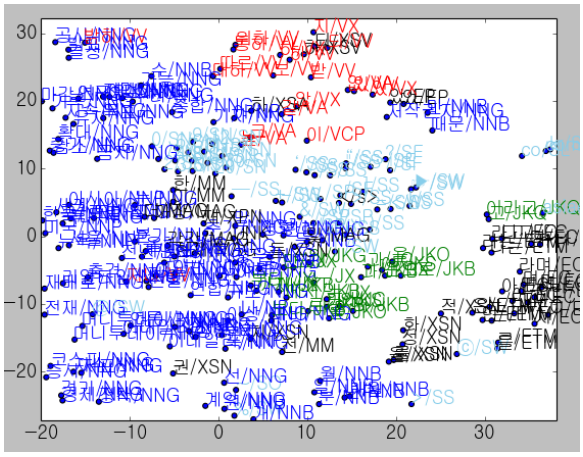


그림 5. 단어의 위치정보를 이용하여 학습한 한국어 word embedding

그림 4와 5는 50차원의 한국어 word embedding을 t-Distributed Stochastic Neighbor Embedding(t-SNE)을 이용하여 2차원으로 나타낸 것이다. 그림 4의 word embedding은 word2vec의 CBOW로 학습한 것이며 품사 구분 없이 분포가 나타나는 것을 볼 수 있다. 그러나 그림 5의 단어의 위치정보를 이용하여 학습 시킨 word embedding의 경우 같은 품사끼리 뭉쳐있는 것을 볼 수 있다. 이것은 단어의 위치정보를 넣어 단순한 단어 출현 빈도 뿐만 아니라 해당 단어의 문장에서의 위치적 역할까지 학습하였다고 볼 수 있다.

단어의 위치정보를 넣은 word embedding이 한국어 자연어처리에 미치는 영향을 확인하기 위해 [7]의 한국어 의존 구문 분석에 적용하였다. 한국어 뉴스기사 및 wiki 데이터(총 29.18억 형태소)를 학습 데이터로 사용하여 word2vec의 CBOW 모델과 본 논문에서 제안한 단어의 위치정보를 이용한 word embedding 학습 모델을 각각 학습시켰다. 의존 구문 분석 모델 및 학습데이터와 평가데이터는 [7]의 연구와 동일하게 사용하였다. 실험 결과 word2vec의 CBOW 모델로 학습한 word embedding을 사용한 경우 UAS 90.27%[7]의 성능이 나왔으나, 단어의 위치정보를 이용한 word embedding을 사용한 경우 UAS 90.49%로 한국어 의존 구문 분석의 성능이 0.22% 오른 것을 확인하였다.

5. 결론

본 논문은 기존의 word embedding 학습기중 하나인 word2vec의 CBOW 모델을 단어의 위치정보를 이용하게 개선하였다. 실험 결과 기존의 word embedding 학습 모델들 보다 syntactic에 강함을 보이며 한국어 형태소의 경우 벡터 공간상에서 비슷한 품사끼리 모이는 것을 볼 수 있었다. 향후 연구로는 각 word embedding 학습 모델들을 word-analogy 뿐만 아니라 개체명 인식, 의미역 결정 등 다른 자연어처리에 미치는 영향을 비교할 예정이다.

감사의 글

이 논문은 SK주식회사C&C의 지원을 받아 연구되었음.

참고문헌

[1] Yoshua Bengio, et al. A neural probabilistic language model. In NIPS, 2001.

[2] Ronan Collobert, et al. Natural language processing (almost) from scratch. The Journal of Machine Learning Research, 12, 2011.

[3] Tomas Mikolov, et al. Distributed representations of words and phrases and their compositionality. Advances in Neural Information Processing Systems. 2013.

[4] J. Pennington, R. Socher, C. D. Manning, “Glove: Global Vectors for Word Representation,” EMNLP2014

[5] 이창기, “Structural SVM을 이용한 한국어 띄어쓰기 및 품사 태깅 결합 모델”, 정보과학회논문지 : 소프트웨어 및 응용, 40(12), pp826-832, 2013.

[6] http://www.nltk.org/

[7] 이창기, 김준석, 김정희, "딥 러닝을 이용한 한국어 의존 구문 분석", 제26회 한글 및 한국어 정보처리 학술대회, pp. 87-91, 2014.

표 2. Word emedding 학습 모델 별 성능표

모델	차원	한국어(4억1900만 형태소)		영어(1억8000만 토큰)		영어(20억9000만 토큰)	
		semantic	syntatic	semantic	syntatic	semantic	syntatic
본 논문의 모델	50	2.65	10.16	6.41	34.03	15.69	47.62
	100	4.19	14.06	7.53	43.26	26.82	60.27
	200	4.19	13.28	7.72	**44.73**	30.29	**64.47**
	400	4.14	**16.41**	6.5	42.24	29.59	62.76
Word2vec(CBOW)	50	1.84	2.34	8.98	18.57	16.61	25.62
	100	2.82	1.56	10.10	27.41	22.96	36.86
	200	3.87	4.69	11.13	30.12	29.31	45.23
	400	4.10	4.69	9.64	30.30	30.12	48.62
Word2vec(skip-gram)	50	5.78	7.03	12.58	17.43	26.91	21.31
	100	7.00	5.47	16.79	24.67	39.96	29.92
	200	7.38	6.25	19.88	29.27	51.76	37.55
	400	6.51	6.25	19.5	28.77	**55.18**	43.32
GLoVe	50	5.88	3.12	7.02	19.37	10.65	26.26
	100	11.17	1.56	18.29	27.43	21.27	38.33
	200	16.60	2.34	24.70	30.90	32.67	45.27
	400	**18.69**	3.12	**27.13**	30.46	39.13	44.47

● 구두발표 4: 언어처리 활용

Sequence-to-sequence 모델을 이용한 로마자-한글 상호(商號) 표기 변환 시스템

김태현○, 정현근, 김재화, 김정길

사람인HR, 사람인LAB
{taehyun.kim, antkdi, jungkil, jaehwa.kim}@saramin.co.kr

Roman-to-Korean Conversion System for Korean Company Names Based on Sequence-to-sequence learning

Tae-Hyun Kim○, Hyun-Guen Jung, Jae-Hwa Kim, Jeong-Gil Kim
SaraminHR, SaraminLAB

요 약

상호(商號)란 상인이나 회사가 영업 활동을 위해 자기를 표시하는데 쓰는 명칭을 말한다. 일반적으로 국내 기업의 상호 표기법은 한글과 로마자를 혼용함으로 상호 검색 시스템에서 단어 불일치 문제를 발생시킨다. 본 연구에서는 이러한 단어 불일치 문제를 해결하기 위해 Sequence-to-sequence 모델을 이용하여 로마자 상호를 이에 대응하는 한글 상호로 변환하고 그 후보들을 생성하는 시스템을 제안한다. 실험 결과 본 연구에서 구축한 시스템은 57.82%의 단어 정확도, 90.73%의 자소 정확도를 보였다.

주제어: 로마자-한글 상호 표기 변환, Sequence-to-sequence, Machine Learning

1. 서론

상호(商號)란 상인이나 회사가 영업 활동을 위해 자기를 표시하는데 쓰는 명칭을 말한다. 일반적으로 상호는 한글 상호 표기와 로마자 상호 표기가 혼용되며, 이는 상호 검색 시스템에서 단어 불일치 문제를 야기한다.

상호의 표기 변환을 위해 국립국어원에서 제정한 국어의 로마자 표기법이나 외래어 표기법을 이용할 수 있지만 인명, 회사명, 단체명 등은 그동안 써 온 관습적 표기를 허용하고 있으며, 실제로 표기법을 따르지 않는 상호가 대부분이다. 또한, “LG”, “SK C&C”, “NHN Entertainment”와 같이 각 로마자를 그대로 한글 표기하거나 숫자 또는 특수 문자와 조합된 상호가 존재하기 때문에 표기법이나 규칙만으로 상호의 로마자-한글 표기를 변환하는 것에는 많은 어려움이 따른다.

따라서, 본 연구에서는 상호 도메인에 적합한 학습 데이터를 구축하는 방법과 Sequence-to-sequence(이하 Seq2seq) 모델[1]을 이용한 로마자-한글 상호 표기 변환 방법을 제안한다. Seq2seq 모델은 기계 번역 분야에서 주로 사용되는 모델로, 입력 시퀀스를 인코딩 및 디코딩하여 길이가 다른 출력 시퀀스를 생성한다. 제안하는 시스템은 규칙 및 자질 튜닝에 의존하는 기존의 연구와 달리 End-to-end 방식으로 접근하였다.

로마자 상호를 한글 상호로 표기 변환하는 시스템은 질의 확장이나 동의어 사전의 구축 등에 활용이 가능하며, 표기 혼용에 따른 단어 불일치 문제의 해결에 기여할 수 있을 것으로 기대한다.

본 논문의 구성은 다음과 같다. 2장에서는 관련 연구를 살펴보고, 3장에서는 전체적인 로마자-한글 상호 표기 변환 시스템의 구성을 소개한다. 4장에서는 제안 시스템의 실험 결과를 분석하고, 마지막 5장에서는 결론에 대해 기술한다.

2. 관련 연구

로마자-한글 상호 표기 변환에 대한 직접적인 연구는 없었으나 음차 표기의 연구가 있었다. 음차 표기란 외국어의 발음을 자국어로 표기하는 것으로 본 연구와 유사하다. 하지만 상호 표기의 경우 외국어의 원래 발음을 그대로 따르지 않는 경우가 대부분이라는 차이점이 있다. 음차 표기의 기존 연구로는 확률 모델을 이용한 연구[2]와 최대 엔트로피 모델을 이용한 연구[3], 메모리 기반 학습과 결정 트리를 이용한 연구[4]가 있었다.

상호 표기 변환에 관한 연구로는 한글-로마자 상호 표기 변환을 위한 부분 문자열 분석에 대한 연구가 있었다.[5] 한글-로마자 상호 표기 변환의 경우 많은 모호성(ambiguity)이 존재하며, [5]의 연구와 같이 부분 문자열 분석이 필수적이다. 예를 들어 한글 상호 “하이텍”은 “Hi-tech”, “High-tech” 등 여러 로마자 표기가 가능하다. 하지만 로마자-한글 상호 표기 변환의 경우 상대적으로 표기 변환 시 모호성이 적다. 따라서 본 연구에서는 로마자 상호에서 한글 상호로 표기를 변환하였다.

본 연구에서 이용한 Seq2seq 모델은 기계 번역[6] 분야 이외에도 형태소 분석 및 품사 태깅[7-8], 구구조 구문 분석[9] 등 다양한 자연어 처리 분야에 적용되고 있으며 우수한 성능을 보이고 있다.

3. 시스템 구축

이 장에서는 먼저 로마자-한글 상호 표기 변환의 시스템에 대해서 자세히 살펴보고, 그다음으로 학습 데이터를 구축하는 방법을 소개하겠다.

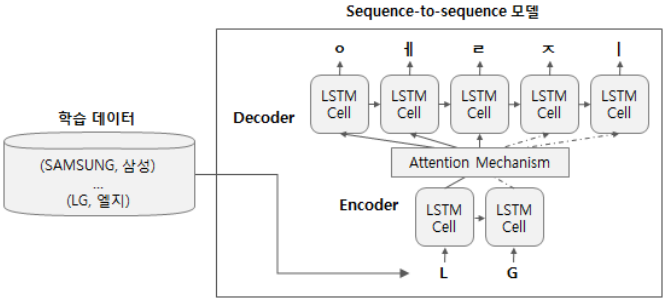
3.1 Sequence-to-sequence 모델을 이용한 로마자-한글 상호 표기 변환

본 연구에서는 로마자-한글 상호 표기 변환을 위해 Seq2seq 모델을 이용하였다. Seq2seq 모델은 입력 시퀀스를 인코딩 및 디코딩하여 길이가 다른 출력 시퀀스를 생성한다. 여기서 모델의 입력 시퀀스는 로마자 상호이며 모델의 출력 시퀀스는 한글 상호이다.

로마자-한글 상호 표기 변환의 문제는 텍스트 요약이나 기계 번역의 문제와 같이 입력 시퀀스와 출력 시퀀스 사이에 일정한 정렬(alignment)이 존재한다. 예를 들어, 로마자 'm', 'b'는 각각 한글 자음 'ㅁ', 'ㅂ'에 명확하게 대응된다. 따라서 Attention Mechanism을[10] 적용함으로써 성능의 향상을 기대하였다.

Attention Mechanism은 디코딩 과정 중 입력을 전역적으로 참고하여 중요한 정보가 있다고 판단되는 특정한 hidden state에 높은 가중치를 주기위한 방법이다.[11]

전체적인 시스템의 구조는 <그림 1>와 같다.



<그림 1> 시스템 구성

3.1.1 한글 표기의 후보 생성

상호 표기는 관습적 표기를 허용하므로 로마자 표기에 대응하는 한글 표기는 다양한 이형태가 존재할 수 있다. 예를 들어 로마자 표기 "RND"는 "알앤디", "알엔디" 등의 이형태가 존재한다. 이러한 이형태는 혼동 자소 때문에 발생하게 되는데 혼동 자소들을 단순하게 치환하는 방법은 불필요한 이형태를 과도하게 생성하여 비효율적이다.

본 연구에서는 모델의 출력 시퀀스에서 가장 모호성이 높은 자소들을 차 순위의 확률 값을 가지는 자소로 치환하여 한글 표기 후보들을 생성하였다.

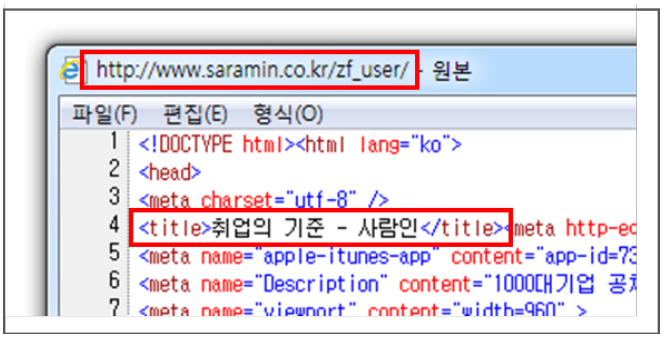
3.2 학습 데이터 구축

이 절에서는 로마자-한글 상호 표기 변환을 위해 Seq2seq 모델의 학습 데이터를 구축하는 방법을 소개한다. 학습 데이터는 <표 1>과 같이 로마자-한글의 쌍으로 구성된다.

<표 1> 학습 데이터의 구성

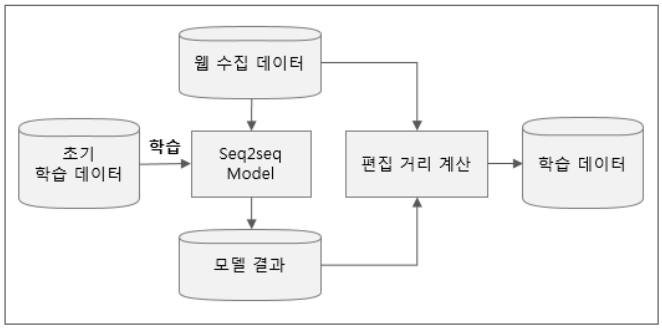
로마자	한글
LG	엘지
SK C&C	에스케이 씨앤씨
NHN Entertainment	엔에이치엔 엔터테인먼트
…	…

본 연구에서는 Seq2seq 모델의 학습 데이터를 구축하기 위해 임의의 웹 사이트의 URL과 TITLE 태그 정보를 이용하였다. URL과 TITLE 태그는 각각 로마자 상호와 한글 상호에 대응하는 경우가 많기 때문이다. 수집한 URL과 TITLE 태그의 예는 <그림 2>와 같다.



<그림 2> URL과 TITLE 태그의 예

학습 데이터를 구축하기 위한 시스템 구성은 <그림 3>과 같다. <그림 3>의 웹 수집 데이터는 URL과 TITLE 태그에서 불필요한 정보를 제거한 로마자-한글 쌍으로 구성된다. 학습 데이터 구축을 위해 우선 초기 학습 데이터를 이용하여 Seq2seq 모델을 학습시킨다. 그다음 웹에서 수집한 로마자-한글 쌍의 데이터 중 로마자 데이터를 학습된 모델의 입력으로 제공하여 한글로 표기 변환된 결과를 얻는다. 마지막으로 모델의 입력에 해당하는 한글 데이터와 모델 결과 사이의 편집 거리(Edit distance) 알고리즘을 계산하여 양질의 학습 데이터만 선별하였다. 편집 거리란 두 개의 문자열이 같아지기 위해 이루어져야 하는 삽입, 삭제, 치환의 최소 연산 개수를 말한다.



<그림 3> 학습 데이터 구축 방법

초기 학습 데이터로는 국립국어원의 외래어 용례와 위키 낱말 사전 등을 이용하였다. 편집 거리는 문자열의 길이로 나누어, 그 값이 0.3 이상인 로마자-한글 쌍들만 학습 데이터로 사용하였다.

3.2.1 한글의 자소 분리

<표 2>는 자소로 분리한 학습 데이터의 예이다.

<표 2> 자소 분리 학습 데이터의 구성

로마자	한글 자소
LG	ㅇㅔㄹㅈㅣ
SK C&C	ㅇㅔㅅㅡㅋㅔㅇㅣ ㅆㅣㅇㅐㄴㅆㅣ
…	…

한글은 19개의 초성과 21개의 중성, 27개의 종성으로 조합이 가능하다. 즉 11,172개의 음절 조합이 가능하며 이는 Vocabulary Size를 증가시켜 학습 데이터에 낮은 빈도로 출현하는 음절에 대한 정확도를 감소시킨다. 따라서, 본 연구에서는 한글을 초성, 중성, 종성의 자소로 분리하여 모델의 학습 데이터를 구축하였다.

4. 실험

4.1 실험 환경

실험 환경은 텐서플로우의 seq2seq 라이브러리를 이용하여 구현하였으며 사용된 데이터 셋은 <표 3>와 같다. 본 연구에서는 금융감독원의 DART 기업명 데이터의 5%인 939쌍을 각각 검증 및 평가 데이터로 사용하였다. 데이터 중 한글 상호와 영문 상호가 다르거나 번역에 해당하는 상호는 데이터 셋에서 제외하였다.

<표 3> 실험 데이터 구성

데이터 셋	갯수
외래어 용례	31898
위키 낱말 사전	783
DART 기업명 데이터	18771
구축한 학습 데이터	74024

Seq2seq 모델은 3-layer, 256 size의 LSTM으로 인코더, 디코더를 구성하였다. drop out은 0.5의 확률로 모든 레이어에 동일하게 적용하였으며 batch size는 64로 학습하였다.

모델의 성능 평가에는 단어 정확도와 자소 정확도를 사용하며, 식은 다음과 같다.

$$단어\ 정확도 = \frac{정답\ 단어\ 수}{전체\ 단어\ 수}$$

$$자소\ 정확도 = \frac{L-(i+d+s)}{L}$$

여기서 L은 원 자소 문자열의 길이를 나타내며, i,d,s는 각각 원 자소 문자열에서 목표 자소 문자열로 변환하기 위해 필요한 삽입, 삭제, 치환의 개수를 나타낸다. 만약 L < (i+d+s)이면 자소 정확도는 0으로 판단한다.[4]

4.2 학습 데이터에 따른 성능 실험

<표 4>은 학습 데이터에 따른 성능 평가 결과이다.

<표 4> 학습 데이터에 따른 실험 결과

데이터 셋	단어 정확도	자소 정확도
외래어 용례 + 위키 낱말 사전	23.85	77.72
DART 기업명 데이터	49.62	87.10
구축한 학습 데이터	53.67	88.46
DART 기업명 데이터 + 구축한 학습 데이터	**57.82**	**90.73**

실험 결과 국립국어원의 외래어 용례나 위키 낱말 사전 데이터의 경우 인명이나 화합물과 같이 일반적으로 상호에 사용하지 않는 데이터들이 많아 성능이 낮은 경향을 보였다. 하지만 본 연구에서 제안한 방법으로 구축한 학습 데이터는 모델의 성능을 향상시키는 것을 볼 수 있다.

4.3 자소 분리에 따른 성능 실험

본 연구에서는 한글을 초성, 중성, 종성의 자소로 분리하여 모델을 학습하였다. <표 5>는 자소 분리에 따른 모델의 성능 평가 결과이다. 모델의 Vocabulary Size는 로마자 50, 한글 자소 60으로 설정하였다.

<표 5> 학습 데이터의 자소 분리 실험 결과

	단어 정확도	자소 정확도
음절	48.56	82.88
자소	**57.82**	**90.73**

실험 결과 한글을 자소 단위로 분리하여 학습한 모델이 음절 단위로 학습한 모델보다 높은 정확도를 보였다. 한글을 자소로 분리함으로써 모델이 예측해야 하는 시퀀스의 길이는 증가하였지만, 빈번하게 출현하지 않는 음절에 대해서도 학습이 가능하기 때문에 모델의 성능을 향상 시킨 것으로 보인다.

<표 6> 자소 분리에 따른 표기 변환 결과

정답		음절 기반	자소 기반
Chulgab	철갑	철비비비	철겁
Bukak …	부각 …	부막 …	부각 …
… Mogul …	… 모굴 …	… 모걸스 …	… 모굴 …
Hawkeyes …	호크아이즈…	홍아이즈…	호크아이즈…
nskorea	엔에스코리아	코리아아아	엔에스코리아

<표 6>의 결과에서 보듯이 음절 단위로 학습한 모델의 경우, ‘…gab’, ‘…kak’, ‘…gul’, ‘hawk…’, ‘ns…’ 등 출현 빈도가 낮은 시퀀스는 제대로 예측하지 못하는 경향을 보였다. 하지만 자소 단위로 학습한 모델의 경우, 대부분 정답과 유사한 음절을 예측하였다. 따라서 로마자-한글 상호 표기 변환의 경우 학습 데이터를 자소로 분리하는 것이 효과적임을 알 수 있다.

4.4 Attention Mechanism 적용 실험

<표 7>은 Attention Mechanism 적용 유무에 따른 성능 평가 결과이다.

<표 7> Attention Mechanism 적용 실험 결과

	단어 정확도	자소 정확도
LSTM Cell	43.45	84.29
LSTM Cell + Attention Mechanism	**57.82**	**90.73**

실험 결과 Attention Mechanism은 모델의 성능을 향상시켰고, 이와 같은 결과는 로마자 표기와 한글 자소 표기 사이에는 명확하게 대응되는 일정한 정렬(alignment)이 존재하기 때문인 것으로 보인다.

4.5 한글 표기의 후보 생성 실험

본 실험에서는 3.1.1 절의 방법으로 생성한 한글 상호 표기의 후보들에 대한 성능을 평가한다. Top-N 단어 정확도는 N개의 후보들 중에 정답이 있을 확률이며, Top-N 자소 정확도는 N개의 후보들의 평균 자소 정확도이다. 실험 결과는 각각 <표 8>과 같다.

<표 8> 한글 표기 후보의 개수에 따른 실험 결과

후보 개수	Top-N 단어 정확도	Top-N 자소 정확도
1	57.82	90.73
2	65.06	87.36
3	67.73	85.22
5	**71.56**	**84.26**
10	72.31	82.54

실험 결과 한글 표기의 후보 개수를 늘릴수록 Top-N 단어 정확도는 증가하지만 Top-N 자소 정확도는 감소하는 결과를 보였다. 후보 개수를 5개 이상 늘릴 때부터 Top-N 단어 정확도의 증가치가 크게 감소하였다. 따라서 로마자 상호에 대한 5개의 한글 상호 표기 후보를 생성하는 것이 효율적이라 볼 수 있다.

5. 결론

본 연구에서는 Sequence-to-sequence 모델을 이용하여 로마자 상호 표기에 대한 한글 상호 표기와 그 후보들을 생성하는 시스템을 소개하였다. 또한 로마자-한글 상호 표기 변환을 위한 양질의 학습 데이터를 구축하는 방법을 제시하였다. 그리고 자소 분리 방법과 Attention Mechanism의 적용을 통해 모델의 성능을 향상시켰다.

하지만 본 연구의 실험 대상에서 번역에 해당하는 상호는 제외하였기 때문에, 이와 같은 경우는 상호의 표기 변환을 지원하지 않는다는 문제점이 남아있다.

향후 본 연구의 시스템을 질의 확장이나 동의어 사전 구축에 이용한다면 단어 불일치 문제의 해결에 기여할 수 있을 것이라 기대한다.

참고문헌

[1] Sutskever, Ilya, Oriol Vinyals, and Quoc V. Le. "Sequence to sequence learning with neural networks." Advances in neural information processing systems, 2014.

[2] 이재성, 최기선, "정보검색을 위한 외래어 자동표기 모델", 한국정보과학회 제4회 학술대회 논문집, pp. 17-24, 1997.

[3] 김태일, "최대 엔트로피 모델을 이용한 다국어 정보검색에서의 영-한 음차 표기 모델", 서강대학교 석사학위 논문, 1999.

[4] 오종훈,배선미,최기선, "글자 및 발음 기반 영-한 음차표기 모델", 한국정보과학회 봄 학술발표논문집, 제31권, 제1호, pp. 925-927, 2004.

[5] 황명진,조선호,권혁철, "한글 상호(商號)를 로마자로 변환하기 위한 고속 부분문자열 분석 알고리즘", 한국정보처리학회 2008년 추계 학술대회 논문집, 제15권, 제2호, pp. 0168-0170, 2008.

[6] K. Cho, D. Bahdanau, F. Bougares, H. Schwenk, Y. Bengio, "Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation," arXiv preprint arXiv:1406.1078, 2014.

[7] 이건일,이의현,이종혁, "Sequence-to-sequence 기반 한국어 형태소 분석 및 품사 태깅", 정보과학회논문지, 제44권, 제1호, pp. 57-62, 2017.

[8] 정의석,박전규, "seq2seq 주의집중 모델을 이용한 형태소 분석 및 품사 태깅", 제28회 한글 및 한국어 정보처리 학술대회 논문집, pp. 217-219, 2016.

[9] 황현선,이창기, "Sequence-to-sequence 모델을 이용한 한국어 구구조 구문 분석", 제28회 한글 및 한국어 정보처리 학술대회 논문집, pp. 20-24, 2016.

[10] Bahdanau, Dzmitry, Kyunghyun Cho, and Yoshua Bengio. "Neural machine translation by jointly learning to align and translation." arXiv preprint arXiv:1409.0473, 2014.

[11] 이현구, 김학수, "주의집중 및 복사 작용을 가진 Sequence-to-Sequence 순환신경망을 이용한 제목 생성 모델", 한국정보과학회논문지, 제44권, 제7호, pp. 674-679, 2017.

심층학습을 이용한 음절태깅 기반의 외래어 인식 시스템

박호민†○, 김창현‡, 천민아†, 노경목†, 김재훈†
†한국해양대학교, 컴퓨터정보공학과
‡한국전자통신연구원
homin2006@hanmail.net, chkim@etri.re.kr, kmq7542@gmail.com, minah0218@kmou.ac.kr, jhoon@kmou.ac.kr

Loanword Recognition Using Deep Learning

Ho-Min Park†○, Chang-Hyun Kim‡, Min-Ah Cheon†, Kyung-Mok Noh†, Jae-Hoon Kim†
†Department of Computer Engineering, Korea Maritime and Ocean University
‡Electronics and Telecommunications Research Institute

요 약

외래어란 외국어로부터 들어와 한국어에 동화되고 한국어로서 사용되는 언어이다. 나날이 우리의 언어사용 문화에서 외래어의 사용 비율은 높아져가는 추세로, 전문분야에서는 특히 두드러진다. 그러므로 더 효율적이고 효과적인 자연언어처리를 위해서 문서 내 외래어 인식은 중요한 전처리 과정이다. 따라서 본 논문에서는 bidirectional LSTM(이하 bi-LSTM)-CRF 모형의 심층학습을 이용한 음절태깅 기반의 외래어 인식 시스템을 제안한다. 제안하는 시스템의 외래어 인식 학습 과정은 다음과 같다. 첫째, 학습용 말뭉치 자료의 한글 음절들과 공백, 마침표(.)를 토대로 word2vec을 통해 학습용 피쳐(feature) 자료를 생성한다. 둘째, 학습용 말뭉치 자료와 학습용 피쳐 자료를 결합하여 bi-LSTM 모형 학습 자료를 구축한다. 셋째, bi-LSTM 모형을 거쳐 학습된 결과물을 CRF 모형에서 로그 가능도(log likelyhood)와 비터비(Viterbi) 알고리즘을 통해 학습 결과물을 내놓는다. 넷째, 학습용 말뭉치 자료의 정답과 비교한 뒤 모형 내부의 수치들을 조정한다. 다섯째, 학습을 마칠 때까지 반복한다. 본 논문에서 제안하는 시스템을 이용하여 자체적인 뉴스 수집 자료에 대해서 높은 정확도와 재현율을 기록하였다.

주제어: 외래어, 음절태깅, bi-LSTM-CRF, word2vec

1. 서론

한글 및 한국어 자연언어처리에 있어서 해당 문서의 주제어를 찾아내는 건 그 문서를 파악하는데 중요한 요소이다. 따라서 문서 분류(classification)나 문서 조직화(organization), 또는 주제어 추출(keyword recognition) 등의 작업에 있어서 반드시 전처리(preprocessing) 과정에 포함하는 경우가 많다. 그렇기에 효율적인 주제어 찾기는 자연언어처리에 있어서 중요한 주제이며 관련 연구가 활발하게 현재까지도 진행되고 있다[1-3].

일반적으로 한 문서에서 중요한 뜻을 가지는 단어의 품사는 명사이며, 그래서 주제어 추출은 해당 문서의 명사들을 추려내 그 중 중요도가 높은 명사를 찾는 일로 간주된다[4].

여러 다양한 분야에서 인터넷을 통한 외국과의 활발한 학문적 교류로 인해서 사회 전반적으로 외래어를 사용하게 되는 경향이 두드러지고 있다. 그러나 외래어는 사용 분야와 적용 범위, 새롭게 만들어지는 주기가 짧고 다양할 수밖에 없다. 그렇기에 사전에 등재될 때까지 외래어는 미등록어가 된다. 이러한 현상은 미등록어 문제를 일으키고 그것은 한국어 자연언어처리에 있어서 큰 걸림돌이다[5]. 따라서 외래어 인식은 중요하고 반드시 필요한 전처리 과정이라고 할 수 있다.

본 논문에서는 심층학습을 이용한 음절태깅 기반의 외래어 인식 시스템을 제안한다. 제안하는 시스템은 심층학습 모형인 bi-directional LSTM과 CRF 모형을 이용하여 외래어를 인식할 문서의 음절마다 태그를 부착하여 외래어를 인식한다.

적용되는 외래어의 범위로는 영어만을 규정했으며 그 이유는 외래어 중 가장 많이 사용되고[6] 현재까지의 관련 연구들 역시 영어 방면에 집중되어 있기 때문이다[5, 7-8].

본 논문의 구성은 다음과 같다. 2장에서 제안하는 심층학습을 이용한 음절태깅 기반의 외래어 인식 시스템에 대하여 소개하고, 3장에서는 결론 및 향후 연구에 대해 기술한다.

2. 심층학습을 이용한 음절태깅 기반 외래어 인식

심층학습은 여러 비선형 변환기법의 조합을 통해 높은 수준의 추상화를 시도하는 기계학습 알고리즘의 집합으로 정의된다. 풀어서 설명하면 큰 틀에서 사람의 사고방식을 컴퓨터에게 가르치는 기계학습의 한 분야라고 이야기할 수 있다[9].

텐서플로(Tensorflow)는 기계학습과 심층학습 모형을 프로그래밍 적으로 구현하기 위해 구글(Google) 사에서 제작 및 배포한 오픈소스 라이브러리이다. 프로그래밍 언어 중에서도 파이썬(Python) 위주로 제작되었으나 Java, Go, C언어 버전도 제공하며 운영체제별 Ubuntu, Mac OS X, Windows 버전 세 가지를 제공한다[10].

젠심(gensim)은 파이썬 프로그래밍 언어에서 문서의 분석 및 분류에 특화되어있는 오픈소스 라이브러리 모듈이다[11].

word2vec은 2013년 구글 사에서 Tomas Mikolov 외 (2013)[12]에서 제안된 단어 임베딩을 위한 기계학습 모형이다. 그를 위한 네트워크 모형 두 가지의 이름은 각각 CBOW(Continuous Bag-of-Words)와 skip-gram 모형이다.

그림 1에서의 CBOW 모형은 크게 입력층(input layer), 전개층(projection layer), 출력층(output layer)으로 이루어져 있으며 문서 내의 각각의 단어마다 앞·뒤의 단어들을 원-핫(one-hot) 변환을 실시하여 목표 단어를 맞추기 위한 학습망을 구성한다.

그림 2에서의 skip-gram 모형은 CBOW 모형과 마찬가지로 입력층, 전개층, 출력층으로 구성되지만 그와는 반대로 각각의 단어를 근거로 앞·뒤에 어떤 단어들이 등장할지 예측하여 맞추기 위한 학습망을 만들어서 멀리 떨어져있는 단어일수록 낮은 확률로 택하는 방법을 사용한다.

따라서 CBOW 모형과 Skip-gram 모형은 서로 반대의 방법을 취하고 있다고 볼 수 있으며 본 논문에서 제안하는 시스템에서는 Skip-gram 모형을 차용하여 음절 임베딩을 실시하였다.

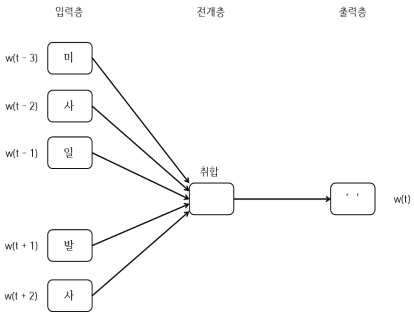


그림 1. CBOW 모형

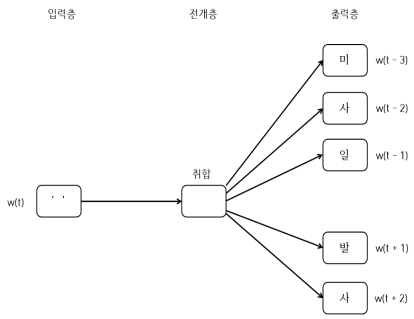


그림 2. Skip-gram 모형

그림3 에서의 LSTM(Long Short Term Memory)이란 인공신경망(artificial neural network)중 하나인 순환(recurrent)인공신경망(이하 순환신경망)에서 사용되는 뉴런 구조의 한 종류이다. 이전 단계의 출력이 다음 단계의 입력 자료가 되는, 기존 순환신경망의 한계인 장기의존성 문제를 해결하기 위해 고안되었다[13].

그림 4에서의 bi-LSTM이란 LSTM으로 구성된 순환신경망의 학습에 있어서 입력 자료에 대한 정보량을 증가시키는 목적으로 소개된 알고리즘으로, 기존의 순환신경망에서는 미래에 입력될 정보가 현재 상태 정보에게 영향을 줄 수 없었지만 bi-LSTM은 입력열의 정방향과 역방향으로 순환하는 두 개의 학습망을 통해서 심층학습을 진행한다[14].

본 논문에서는 학습 말뭉치 자료에 대한 학습 모형으로 bi-LSTM을 사용하여 입력되는 문장의 구문 정보를 양방향으로 제공하여 좀 더 효율적인 음절태깅을 수행하였다.

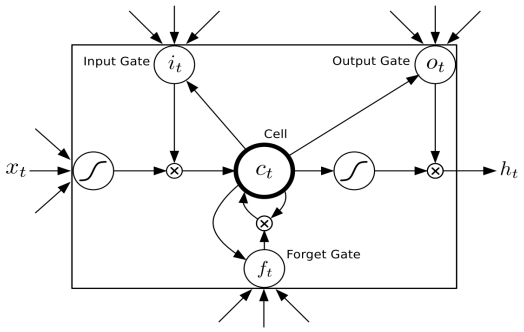


그림 3. LSTM 구조

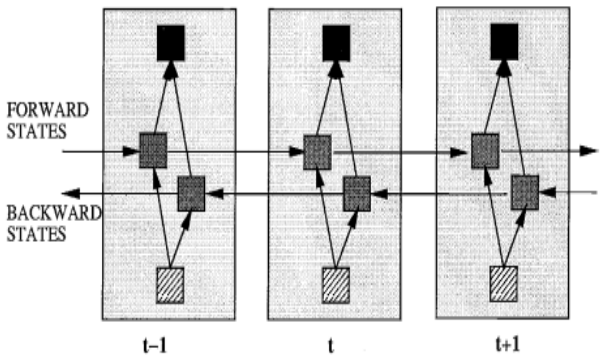


그림 4. bi-directional LSTM 망 구조 예시

CRF(Conditional Random Field)는 통계적으로 모형을 제작하는 방식으로 기계학습에서 모형의 구조에 따라 예측을 판단하는데 이용된다. 이론의 세부 내용으론 방향성이 없는 그래프를 제작하는데 그 그래프의 정점(vertex)은 입력되는 자료열들이 구성하며 각 정점마다 다른 정점으로 넘어갈 상태 전환 확률이 정점들을 연결하는 간선(edge)이 되어 그래프를 구성한다[8].

본 논문에서 제안하는 시스템에서는 bi-LSTM을 통해 학습되어 나온 결과값들을 이용하여 정점과 간선을 로그가능도(log likelyhood)로 구한다. 구하고난 뒤, 선형체인(linear chain) 형태 그래프를 제작하여 음절태깅을 진행한다.

비터비 알고리즘은 결과값이 나오지 않은 은닉 상태열로 구성된 그래프에서 가장 가능성이 높은(most likely) 예측 결과열을 생성하는 동적 프로그래밍 알고리즘이다. 은닉 상태열의 시작부터 끝까지 각 단계의 확률과 상태 전이 확률을 사용해서 해당 단계의 가장 가능성이 높은 결과를 선택하는데 그 선택된 결과들이 모여 예측 결과열(Viterbi path)이 된다[9].

본 논문에서 제안하는 시스템에서는 bi-LSTM을 통해 학습되어 나온 결과를 CRF를 통해 선형 체인 그래프로 만든 뒤 해당 알고리즘을 사용하여 각 학습 단계의 최종 결과물인 예측 결과열을 생성한다.

3. 심층학습 음절태깅 기반 외래어 인식 시스템

문서 내에서 외래어를 인식하기 위해 일반적인 어절이나 단어적, 형태소적 접근이 아닌 음절에 따른 한국어('K' 태그)와 외래어('E' 태그) 분류로 접근하였다. 본 논문의 시스템에서는 젠심 모듈의 word2vec 모형으로 음절 임베딩을 시행하는 전처리 단계와 그러한 전처리 결과물로 실질적인 음절태깅을 위한 학습을 진행하는 bi-LSTM 모형, 학습 결과물을 다듬고 차원 축소(dimentionality reduction)를 진행하여 예측 결과를 도출해내 외래어를 인식하는 CRF 모형과 비터비 알고리즘을 이용한 후처리로 이루어져 있다.

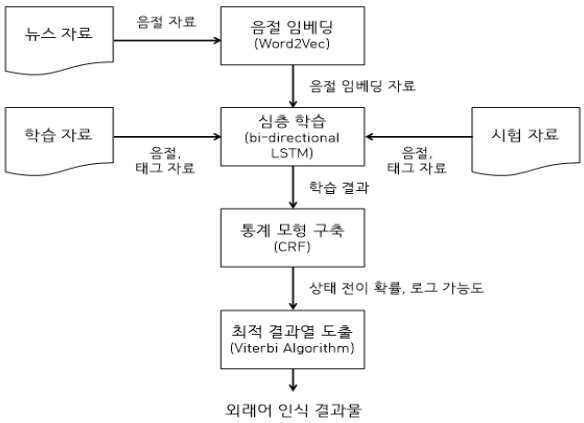


그림 5. 음절태깅을 이용한 외래어 인식 시스템

word2vec 모형에서 학습 자료로 사용된 뉴스 자료는 자체 수집한 뉴스 문서들을 이용하였다. 가능한 다양한 자료를 담기위해 문화, 경제, 연예, 국제, 과학, 지역, 정치, 사회, 스포츠의 9개 분야를 사용하였다. 분량은 약 2GB 정도이며 연합뉴스의 2017년도 분을 사용하였다.

음절 임베딩 단계에서는 skip-gram 방식을 이용한 word2vec 모형을 이용한다. 입력되는 각 음절에 대해 앞·뒤에 어떤 음절이 있을지를 예측한다. 가까이 있을수록 그 확률이 높아지고 멀리 있을수록 낮게 책정된다.

bi-LSTM 모형에서 학습 및 시험 자료로 사용된 자료는 자체 제작한 1만여 문장의 뉴스 보도 자료와 정답 자료를 사용했으며 80%를 학습에 사용했고 나머지 20%로 시험을 진행했다. 정답으로 쓰인 태그 종류는 총 네 가지로 표 1과 같이 설정하였다.

표 1. 태그의 종류

종류	설명
K	한국어 음절
E	외래어 음절
' '	띄어쓰기 된 부분
'.'	마침표

심층학습 단계에서는 학습 자료를 음절 단위로 분리하여 순차적으로 bi-LSTM 모형에 입력받는다. 그림 6과 같이 음절을 입력받아 학습 결과를 결합하여 다음 단계인 CRF 모형의 자질을 만들어낸다.

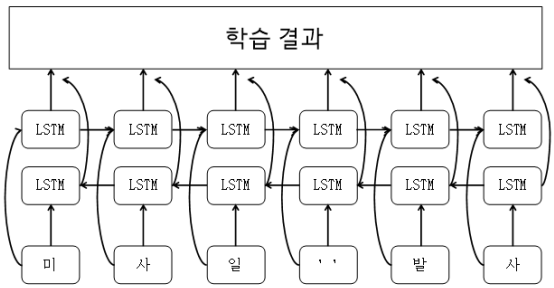


그림 6. bi-LSTM 학습 예시

통계 모형 구축 단계에서는 CRF 모형을 이용하여 선형 체인 형태의 그래프를 구성한다. 각 음절에 따른 정점과 학습 결과에 차원 축소를 진행한 값들을 간선으로 이용한다. 간선의 수치 계산 방법은 로그 가능도를 사용하며 그것을 최적 결과열 도출 단계에서 비터비 알고리즘을 적용할 때 각 정점에 대한 상태 전이 확률로 사용한다. 그렇게 비터비 알고리즘을 이용하여 동적 프로그래밍 방식으로 그림 7처럼 한 단계 한 단계 음절들에 대한 태그를 결정한 태그 예측열을 최종 결과물로 제출하게 되고 학습 과정에서는 정답과 결과물을 비교하여 학습율(learning rate)에 따라 내부 수치를 재조정하여 학습을 지속해 나간다.

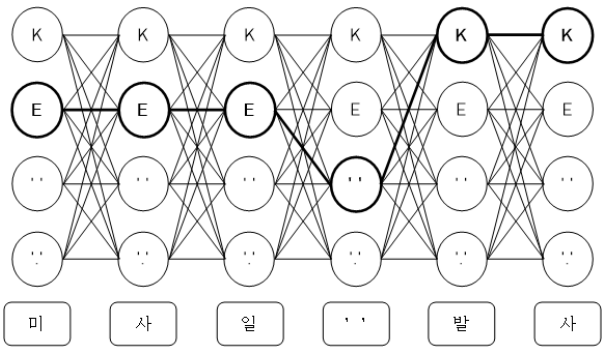


그림 7. CRF 그래프 구축 및 Viterbi Path 추적

4. 실험 결과

word2vec 모형 학습에 이용한 뉴스 자료는 연합뉴스의 2017년도 분의 문화, 경제, 연예, 국제, 과학, 지역, 정치, 사회, 스포츠의 9개 분야를 약 2GB 정도 수집하여 사용하였다. bi-LSTM-CRF 모형의 학습에 사용된 학습 자료와 시험 자료는 자체 제작한 1만여 문장의 KBS 뉴스 보도 자료와 그에따른 외래어, 한국어 태깅 결과 자료를 사용했다. 비율은 80 : 20으로 나누어 활용했다.

평가 방식은 단순 음절태그 예측 정확도(accuracy)-정확률(precision)-재현율(recall)-f1 measure 값, 한글 음절 임베딩 피쳐의 차원 수(50개, 100개), 태그 개수(2개('K', 'E'), 4개('K', 'E', ' ', '.'))에 따라 세 가지 방법으로 진행하였다. 음절 임베딩 자료 종류는 표 2의 내용과 같다.

표 2. 제작한 음절 임베딩 자료 종류

종류	설명
버전 1	임베딩 차원 = 50, 태그 수 = 2
버전 2	임베딩 차원 = 50, 태그 수 = 4
버전 3	임베딩 차원 = 100, 태그 수 = 2
버전 4	임베딩 차원 = 100, 태그 수 = 4

표 3. 방법에 따른 학습 결과

	정확도	정확률	재현율	f1 measure
버전 1	82.21	80.65	78.44	79.53
버전 2	85.62	84.37	83.85	84.11
버전 3	87.78	85.54	83.84	84.68
버전 4	90.53	88.39	86.19	87.28

표 4. 방법에 따른 실험 결과

	정확도	정확률	재현율	f1 measure
버전 1	76.43	75.07	73.59	74.32
버전 2	77.99	75.41	74.44	74.92
버전 3	80.21	79.54	78.93	79.28
버전 4	83.55	81.79	80.17	80.97

음절 임베딩의 차원이 50차원인 것보다 100차원일 경우에 평균적으로 높은 수치를 기록했으며, 태그를 2개 사용한 것 보다 띄어쓰기와 마침표를 넣어서 최소한의 문맥적 의미를 제공한 태그가 4개인 버전이 평균적으로 높은 수치를 기록했다. 이는 외래어 인식을 위한 올바른 음절태깅에 있어서 가능한 다양한 정보가 학습 모형으로 하여금 신뢰도 높은 예측을 하게 만든다는 것을 의미한다.

5. 결론

본 논문에서는 심층학습을 이용한 음절태깅 기반의 외래어 인식 시스템을 제안하였다. 해당 시스템은 파이썬 프로그래밍 언어의 젠심 모듈을 이용해 word2vec 모형을 제작하여 한글 음절 임베딩의 피쳐를 제작하였고, 제작한 한글 음절에 대한 음절 임베딩 자료를 bi-LSTM과 CRF 모형을 이용하여 문서의 음절마다 'K'(한국어) 태그, 'E'(외래어) 태그를 부여해 외래어 인식을 수행한다.

제작한 시스템 내부의 word2vec 모형을 위한 학습용 자료로써 자체 수집한 뉴스 자료를 이용하였고, bi-LSTM-CRF 모형을 위한 학습용 자료로써 자체 제작한 음절태깅을 진행한 뉴스 말뭉치를 사용하였다.

하지만 가장 치명적인 약점은 학습 자료에 존재하지 않았던 외래어를 만났을 때 인식율이 낮았으며, 학습 단계에 있어서 어려움은 각 단계마다 과적합(overfitting)이 될 수 있다는 것과 말뭉치 내 외래어 음절보다 한국어 음절의 절대 개수의 높은 차이로 인해 음절 태깅의 결과가 한국어 태그로 편중(bias)될 수도 있다는게 있었다. 첫 번째로 제시한 약점과 편중 문제는 학습 말뭉치 자료의 추가적인 확보 및 정제로 어느정도 해결할 수 있을거라 생각하며 과적합 문제는 학습율 조정 및 학습율 감퇴 적용 등을 추가적으로 연구할 예정이다.

본 논문에서 제안하는 시스템의 개선을 위하여 향후 연구로 외래어 사전 추가, 학습 말뭉치 추가 확보 및 정제, 음절에 대해 추가적인 정보 제공 방법 연구 등을 진행하여 음절태깅을 이용한 외래어 인식 시스템의 성능을 향상시킬 계획이다.

감사의 글

이 논문은 2017년도 정부(과학기술정보통신부)의 재원으로 정보통신기술진흥센터의 지원을 받아 수행된 연구임(R7119-16-1001, 지식증강형 실시간 동시통역 원천기술 개발)

참고문헌

[1] 배경만, 김성현, 고영중, 김종훈, "자연어 기반 인터페이스에서 개체명 패턴을 이용한 효과적인 개체명과 주제어 인식 방법", 한국정보기술학회논문지, 제12권 제1호, 121-129, 2014.

[2] 주길홍, 이주일, 이원석, "효율적인 문서 검색을 위한 연관 키워드 추출 및 확산 클러스터링 방법", 한국정보기술학회논문지, 제9권 제6호, 155-166, 2011.

[3] 유은순, 최건희, 김승훈, "TF-IDF와 소설 텍스트의 구조를 이용한 주제어 추출 연구", 한국컴퓨터정보학회논문지, 제20권 제2호, 121-129, 2015.

[4] 안희정, 김기원, 김승훈, "복합 명사구 합성 방법을 적용한 효과적인 도서 본문 주제어 추출", 한국컴퓨터정보학회논문지, 제22권 제3호, 107-113, 2017.

[5] 오종훈, 최기선, "은닉 마르코프 모델을 이용한 음차표기된 외래어의 자동인식 및 추출 기법", 인지과학, Vol.12 No.3, 19-28, 2001.

[6] 조남호, "한국어의 외래어 수용과 대응", 인문과학연구논총, 제35권 3호, 11-38, 2014.

[7] 박종혁, "유사 외래어 검출 알고리즘의 성능 향상", 충북대학교 석사학위논문, 2004.

[8] 고숙현, "문맥을 고려한 유사 외래어 검출 알고리즘", 충북대학교 석사학위논문, 2007.

[9] Y. Bengio, A. Courville, and P. Vincent., “Representation Learning: A Review and New Perspectives,” IEEE Trans. PAMI, special issue Learning Deep Architectures, 2013.
[10] [online]https://www.tensorflow.org, 2015.
[11] [online]https://radimrehurek.com/gensim, 2011.
[12] Tomas Mikolov et al. “Efficient Estimation of Word Representations in Vector Space”, 2013.
[13] Sepp Hochreiter and Jurgen Schmidhuber, “Long Short-Term Memory”, 1997.
[14] Mike Schuster and Kuldip K, “Bidirectional Recurrent Neural Networks”, 1997.
[15] John Lafferty, Andrew McCallum and Pernando C.N. Pereira, “Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data”, 2001.
[16] G. David Forney, Jr., “The Viterbi Algorithm: A Personal History”, 2005.

Distance LSTM-CNN with Layer Normalization을 이용한 음차 표기 대역 쌍 판별

이창수○, 천주룡, 김주근, 김태일, 강인호

네이버 검색

{ changsu.lee, juryong.cheon, joogeun.kim, eiji.kim, once.ihkang } @ navercorp.com

Verification of Transliteration Pairs Using Distance LSTM-CNN with Layer Normalization

Changsu Lee○, Juryong Cheon, Joogeun Kim, Taeil Kim, Inho Kang

Naver Corporation

요 약

외국어로 구성된 용어를 발음에 기반하여 자국의 언어로 표기하는 것을 음차 표기라 한다. 국가 간의 경계가 허물어짐에 따라, 외국어에 기원을 두는 용어를 설명하기 위해 뉴스 등 다양한 웹 문서에서는 동일한 발음을 가지는 외국어 표기와 한국어 표기를 혼용하여 사용하고 있다. 이에 좋은 검색 결과를 가져오기 위해서는 외국어 표기와 더불어 사람들이 많이 사용하는 다양한 음차 표기를 함께 검색에 활용하는 것이 중요하다. 음차 표기 모델과 음차 표기 대역 쌍 추출을 통해 음차 표현을 생성하는 기존 방법 대신, 본 논문에서는 신뢰할 수 있는 다양한 음차 표현을 찾기 위해 문서에서 음차 표기 후보를 찾고, 이 음차 표기 후보가 정확한 표기인지 판별하는 방식을 제안한다. 다양한 딥러닝 모델을 비교, 검토하여 최종적으로 음차 표기 대역 쌍 판별에 특화된 모델인 Distance LSTM-CNN 모델을 제안하며, 제안하는 모델의 Batch Size 영향을 줄이고 학습 시 수렴 속도 개선을 위해 Layer Normalization을 적용하는 방법을 보인다.

주제어: 음차판별, 음차검증, 정보검색, 딥러닝

1. 서론

외국어로 구성된 용어를 발음에 기반하여 자국의 언어로 표기하는 것을 음차 표기라 정의한다[1][2]. 국가 간의 경계가 허물어짐에 따라 외국어에 기원을 두는 용어(starbucks)를 설명하기 위해 뉴스, 블로그 등의 다양한 웹 문서에서는 외국어 표기와 한국어 표기를 혼용하여 사용하는 경우가 나날이 증가하고 있다. 특히 정보 검색에서는 외국어 표기(starbucks) 하나만을 사용하여 검색을 수행하면 한국어 표기(스타벅스)만으로 문서화되어 있는 양질의 문서들이 검색 결과에서 제외되어 원하는 검색 결과를 얻을 수 없는 문제가 발생한다. 이러한 문제를 해결하기 위해서는 외국어 표기와 더불어 사람들이 많이 사용하는 다양한 음차 표기(한국어 표기 등)를 함께 검색에 활용하는 것이 중요하다.

주어진 외국 용어 및 외국어 표기에 대한 다양한 음차 표기를 얻기 위한 연구로는 음차 표기 모델(Transliteration Model), 음차 표기 대역 쌍 추출(Transliteration Pairs Extraction), 음차 표기 대역 쌍 판별(Transliteration Pairs Verification) 등의 연구가 있었으며, 이 연구들은 주로 검색 품질 향상을 위해 질의 확장 및 언어 자원을 구축하는 목적으로 연구되었다[3].

음차 표기 모델(Transliteration Model)은 외국어 표기를 입력으로 하여 자국어의 표기를 자동으로 생성하는 방법이다. 주로 검색 품질 향상을 위해 번역사전에 존재하지 않는 외국 단어의 음차 표기를 자동으로 생성하기 위한 연구로 진행되었으며, 최근에는 딥러닝을 이용한 음차 표기 모델이 가장 좋은 결과를 보였다[1][4][5][6]. 하지만, 음차 표기 모델을 통해 생성된 음차 표현은 일반적으로 품질이 좋지 않으며, 검색 시스템에서 사용자들이 자주 사용하는 음차 표기가 아닌 경우가 많아, 상용 검색 시스템에 적용하기에는 문제가 있었다.

음차 표기 생성 방법과 달리 문서에서 음차 표기를 추출하기 위한 연구로, 음차 표기 대역 쌍 추출(Transliteration Pairs Extraction) 연구가 있었다[7]. 음차 표기 대역 쌍 추출은 이중 언어 문서에서 외국어와 외국어에 대응되는 음차 표기된 용어들을 자동으로 추출하기 위한 방법이다[3]. 주로 패턴 및 기계학습 등을 이용하여 음차 표기 대역 쌍 후보를 추출하고, 서로 다른 언어로 구성된 음차 표기 대역 쌍 후보를 하나의 언어 형태로 변환한 후, 편집거리 알고리즘을 이용하여 음성적 유사도를 계산하는 방법으로 음차 표기 대역 쌍을 추출했다[3][7]. 하지만 음성적 유사도를 적용하여 음차 대역 쌍을 추출하는 방법은 “starbucks - 스타박스”와 같이 음성적으로 유사하지만, 다른 의미를 가지는 용어(스타박스)가 동일한 음차로 추출되는 문제가 있었으며, 이처럼 잘못 추출된 음차 정보를 검색 시스템에 적용하게 되면 검색 품질에 심각한 문제를 야기하게 된다.

정보 검색에 활용하기 적합하며, 신뢰할만한 음차 표기 대역 쌍을 얻기 위한 연구로는 음차 표기 대역 쌍 판별(Transliteration Pairs Verification) 연구가 있었다. 이는 서로 다른 언어로 구성된 음차 표기 대역 쌍 후보

가 정확한 음차 관계인지 판별하는 것이며, 특히 인명 등과 같이 난도가 높은 음차 표기 대역 쌍(James Rodriguez - 하메스 로드리게스)을 정확하게 판별하기 위해 제안되었다[8][9].

본 논문에서는 검색 품질 향상을 위해, 웹 문서에서 자주 사용되는 다양한 음차 표현을 찾고 이를 언어 자원으로 구축하기 위한 음차 표기 대역 쌍 판별 모델을 제안한다. 기존에 제안되었던 음차 표기 모델과 음차 표기 대역 쌍 추출 모델은 저품질 문제와 더불어 검색 시스템에 적합하지 않는 음차 표기를 생성, 추출하는 문제가 있어 활용하기 어려우며, 두 음차 표기가 정확한 음차 관계인지 판별하는 음차 표기 대역 쌍 판별 연구에 기초하여 문제를 해결한다.

따라서, 본 연구에서는 기존의 음차 표기 모델과 음차 표기 대역 쌍 추출 모델에서 특히 문제가 되는 난도가 높은 음차 표기 후보를 정확히 판별하기 위한 음차 표기 대역 쌍 판별 모델을 구축하는 것을 목적으로, 다양한 딥러닝 모델을 구축하고 딥러닝 모델 간의 비교, 검토를 통하여 최종적으로 음차 표기 대역 쌍 판별에 특화된 모델인 Distance LSTM-CNN with Layer Normalization 모델을 제안한다.

평가를 위해 정보 검색에서 사용자들이 자주 사용하는 음차 표기 후보를 웹 문서에서 수집하되, "AOA - 초아"와 같이 판별 난도가 낮은 음차 표기 대역 쌍 후보는 판별 모델의 차별성을 증명할 수 없으므로 제외하고, 주로 난도가 높은 음차 표기 후보들을 추출하여 데이터 셋을 구축함으로써, 제안하는 방법의 실용성을 검증한다.

본 논문에서 제안하는 음차 표기 대역 쌍 판별에 특화된 모델인 Distance LSTM-CNN with Layer Normalization은 한국어-영어간 음차판별을 위해 변형된 KODEX 방법과의 비교 결과, 약 35%의 품질 향상을 보였으며 [4]에서 높은 품질을 보이는 음차 생성 모델을 음차 판별 모델로 변형한 Sequence-to-Sequence with Attention 모델과의 비교에서도 약 3.5%의 품질 향상을 보였다. 또한, 두 질의의 관련성을 판별하는 연구에서 높은 품질을 보이는 딥러닝 모델인 Distance LSTM 모델보다 약 3% 정도의 품질 향상을 보여 최종적으로 89.70%의 품질을 보였다. 마지막으로 Layer Normalization을 적용한 모델이 적용하지 않은 모델과 비교해 약간의 품질 향상과 더불어 수렴 속도가 약 3배 빨라짐으로써 Layer Normalization 효과를 확인할 수 있었다.

본 논문은 다음과 같이 구성되어 있다. 1장의 서론에 이어 2장에서는 관련 연구에 대해 살펴보고, 3장에서는 본 논문에서 제안하는 음차 표기 대역 쌍 판별 모델에 대해 설명한다. 4장에서는 실험 결과를 비교, 검토하며 마지막 5장에서는 결론에 대해 살펴 본다.

2. 관련 연구

2.1 음차 표기 모델(Transliteration Model)

음차 표기 모델은 외국어 표기를 입력으로 자국어 표기를 생성하는 연구로써, 주로 확률 및 기계학습을 이용하여 연구되어왔다[1][4][5][6]. [1]에서는 확률 기반 음차 표기 모델을 제안했으며, 발음 단위를 음소에 매핑할 수 있도록 매핑 테이블을 정의하고, 확률 모델을 이용하여 주어진 영어 단어에 대한 가장 높은 확률을 가진 한국어 음차 표기를 생성하였다. [5]는 최대 엔트로피를 이용하여 음차 표기를 생성하는 방법을 제안하였으며, [6]은 음성적 정보와 자소/음소의 문맥정보를 이용, 결정 트리 및 메모리 기반 학습 모델에 활용하여 한국어 음차 표기를 생성하는 방법을 제안했다. 최근 [4]에서는 기계번역에 주로 사용되는 Sequence-to-Sequence 모델을 활용하여 영어를 기반으로 다양한 언어의 음차 표기를 생성하는 모델을 제안하였다.

2.2 음차 표기 대역 쌍 추출(Transliteration Pairs Extraction)

음차 표기 대역 쌍 추출은 이중 언어 문서에서 외국어와 그에 대응되는 음차 표기된 용어를 자동으로 추출하는 방법이다[3][7]. 주로 패턴 및 기계학습을 활용하여 음차 표기 대역 쌍 후보를 추출하고, 추출된 외국어 및 자국어 표기가 정확한 음차 관계인지 여부를 계산하여 음차 표기 대역 쌍을 추출하였다. [3]은 패턴을 이용하여 이중 언어 문서에서 음차 표기 대역 쌍 후보를 추출하고, 음차 표기 모델(Transliteration Model)을 활용하여 외국어를 입력으로 한국어 음차 표기를 생성한 후, 생성된 음차 표기와 대역 쌍 후보에서 추출된 한국어 음차 표기와의 음성적 유사도를 계산하여 음차 표기 대역 쌍을 추출하는 방법을 제안하였다. 그리고, [7]은 영어-일본어 음차 표기 대역 쌍 추출을 위해 잡음 채널 오류 모델과 학습 가능한 편집거리 함수를 이용하여 음차 표기 대역 쌍을 추출하였다.

2.3 음차 표기 대역 쌍 판별(Transliteration Pairs Verification)

음차 표기 대역 쌍 판별은 외국어 표기와 자국어 음차 표기로 구성된 음차 표기 대역 쌍 후보가 정확한 음차 관계인지 판별하는 것이며, 주로 기계번역과 교차언어 정보 검색과 같이 다국어 자연어 처리 작업에 활용되었다[9]. [8]에서는 동일한 언어 간 다양한 외국어 음차 표기 대역 쌍(디지털-디지탈) 비교를 위해 두 입력을 특정 기호로 인코딩하여 비교하는 KODEX 알고리즘을 제안하여, 한국어간 다양하게 표현되는 음차가 동일한지 판별하였다. [9]는 음차 판별이 어려운 인명의 음차 표기 대역 쌍을 판별하기 위해 Discrete Variant Hidden Markov Model (HMM) Alignment 기법을 제안하였으며, 국가간 발음이 달라 음차 판별 난도가 높은 인명 음차 표기 대역 쌍 판별에서 좋은 결과를 얻었다.

3. 음차 표기 대역 쌍 판별 모델

이 장에서는 정보 검색에 적합한 음차 표기 언어 자원을 구축하는 것을 목적으로, 웹 문서에서 추출되는 다양한 음차 표기 대역 쌍 후보가 정확한 음차 관계인지 판별하는 딥러닝 기반 음차 표기 대역 쌍 판별 모델을 구축하는 방법을 보인다.

3.1 Sequence-to-Sequence with Attention 모델

Sequence-to-Sequence 모델은 입력 문장을 $x = x_1, \ldots, x_T$ 로 인코딩 한 후, 디코더를 통해 $P(y|x)$ 를 최대

화하는 출력 문장 $y = y_1, \ldots, y_k$ 을 생성하는 모델이며, 주로 기계 번역 및 챗봇에 적용되어 만족할 만한 결과를 보였다[10]. [4]에서는 영어 표기를 기반으로 다양한 언어(아랍어, 일본어, 중국어 등)의 음차 표기를 생성하기 위해 Sequence-to-Sequence 모델을 적용하여 성공적인 결과를 얻었다. 이와 같은 연구에 기초하여 본 연구에서는 음차 표기 대역 쌍 판별을 위한 Sequence-to-Sequence 모델을 구축하였다. 그리고 LSTM Cell을 추가하여 RNN 내부에 3개의 게이트(Input, Output, Forget)와 1개의 메모리 공간으로 정보를 갱신 혹은 제거를 통해 멀리 있는 정보가 희미해지는 RNN의 그래디언트 소멸 문제(Gradient Vanishing Problem)를 해결하도록 하였다[11]. 또한 단계별 중요도가 고려 될 수 있도록 주의(Attention) 기법을 이용하여 모델을 구성했다.
[그림 1]은 음차 표기 대역 쌍 판별을 위해 구축한 Sequence-to-Sequence with Attention 모델의 구성도 이다.

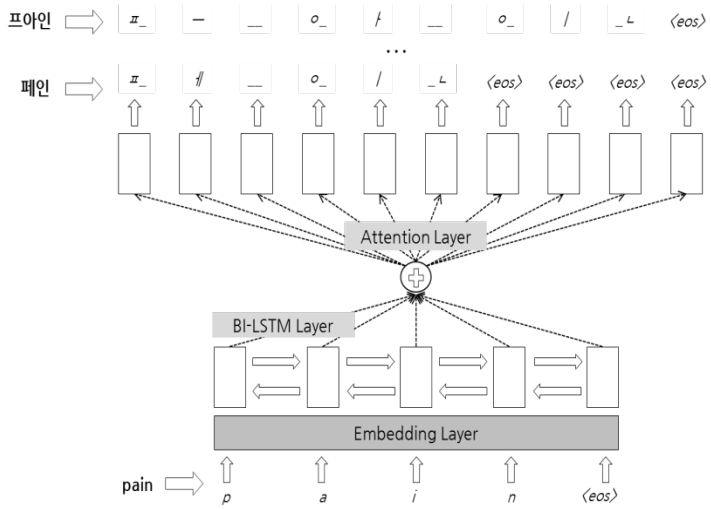


그림 1. Sequence-to-Sequence with Attention 모델

본 연구에서의 목적은 웹 문서에서 추출된 다양한 음차 표기 대역 쌍 후보가 적합한 음차 관계인지 여부를 판별하는 것이므로, 상위 1개의 결과만 추출되는 기존 디코더를 상위 N개의 음차 표기 생성 결과를 추출할 수 있도록 변경했다. 평가를 위해 정답이 부착되어있는 음차 표기 대역 쌍 후보와 모델에서 추출한 음차 표기 추출 결과 상위 N개 중 동일한 음차 표기가 존재하는지 확인하는 방법으로 음차 표기 대역 쌍 판별 모델을 구축했다.
[그림 2]는 음차 표기 대역 쌍 후보를 판별하는 예이다.

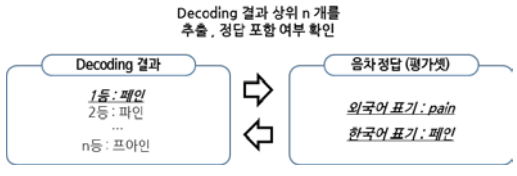


그림 2. 디코더를 변경한 음차 표기 대역 쌍 판별 방법

3.2 Distance LSTM 모델

정보 검색에 적합한 음차 표기 언어 자원을 구축하기 위해서는 음차 표기 대역 쌍 후보가 형태적, 의미적으로 동일한지 여부를 판별해야 한다. 이와 관련된 최신 연구들은, 두 문장의 관련성을 판별하기 위해 구축된 SNLI[12] 및 Quora Question Pairs[13] 데이터를 이용한 연구들이며, 주로 질의-응답 및 동의질의 판별 문제를 해결하기 위해 제안되었다[14][15][16][17]. [15][16][17]은 두 질의가 동일한 의미를 지니는지 판별하기 위해 다양한 정렬(Alignment) 및 주의(Attention) 기법을 적용했다. 하지만, 음차 표기의 경우 음차간 동일한 시퀀스로 매칭되어야 하는 특성이 있으며, 정렬 및 주의 기법을 적용하게 되면 "one way - 웨이 원"과 같이 동일한 단어로 구성되지만 다른 시퀀스를 가진 음차 표기를 동일하게 판별하는 문제가 있었다. 동일한 시퀀스를 유지해야 하는 음차 표기 특성을 반영하여 본 논문에서는 [14]의 Distance LSTM 모델을 이용하여 음차 표기 대역 쌍 판별 모델을 구축했다. 그리고 거리벡터(Distance vector) 연산에서 빼기(Subtract) 및 곱하기(Multiply) 연산이 가장 좋은 품질을 보인다는 [16]의 연구를 반영하여, Distance LSTM 모델의 품질 개선을 위해 거리벡터 연산(Distance vector)에 빼기(Subtract) 및 곱하기(Multiply) 연산이 적용되도록 변경하였다. 결과적으로 다른 거리 벡터 연산과 비교해 가장 좋은 결과를 얻을 수 있었다.
[그림 3]은 음차 표기 대역 쌍 판별을 위해 구축한 개선된 Distance LSTM 모델의 구성도이다.

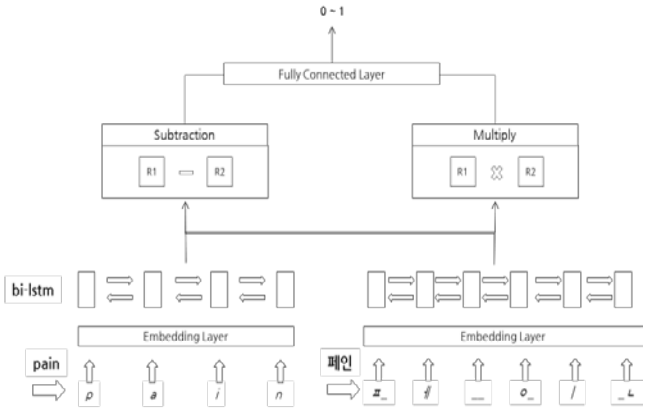


그림 3. Distance LSTM 모델

3.3 Distance LSTM-CNN with Layer Normalization 모델

Convolutional Neural Networks(CNN) 모델은 입력 문장의 주위 문맥 정보를 Convolutional Layer을 통해 추출하고, Pooling Layer를 거쳐 중요 자질만을 추상화하는 특징을 가지는 구조이다. [18]은 Convolutional Layer와 Max Pooling Layer로 구성된 단순한 구조의 CNN 모델을 제안하여 문장 및 문서 분류에서 좋은 결과를 얻었다.

음차 표기의 경우, 두 입력이 동일한 시퀀스로 매칭되어야 하는 특성이 있으므로, 순서를 유지하면서 주위 문맥(음차)의 중요 정보를 추상화하는 CNN 구조를 추가하는 것은 음차 표기 대역 쌍 판별을 위한 중요한 자질이 될 수 있으며, 음차 간 긍정적인 연관 관계를 부여할 수 있다고 가정했다. 이와 같은 가정을 기반으로 음차 판별에 적합한 CNN 구조를 추가, 적용했다.

음차 판별에 적합한 중요 주위 문맥 정보를 추가하기 위한 CNN 구조는 아래처럼 정의하여 적용하였다.

$$C_h = \text{MaxPooling}\left(\text{Conv1d}_{fsize}(\text{x}_1, \dots \text{x}_{n-h+1:n})\right)$$

x는 입력이며, n은 음차 길이, h는 현재 음차를 기준으로 몇 개의 인접한 음차를 조합을 생성할 것인지의 개수(윈도우 크기), Conv1d는 1차원 합성곱, fsize는 필터의 수, 마지막으로 MaxPooling은 벡터를 추상화하기 위해 최대 Pooling을 수행하는 함수이다.

정의된 CNN 구조를 이용, h(윈도우 크기)를 1, 2, 3으로 설정한 후, 거리벡터(Distance vector) 연산인 빼기(Subtract) 및 곱하기(Multiply) 벡터 연산에서 출력된 벡터 각각에 대해 [그림 4]와 같이 CNN 구조를 추가했다. 정의한 CNN 구조를 추가함으로써 인접한 음차 조합 자질을 생성하고, 중요 음차 자질만을 추상화하는 방법으로 음차 판별에 적합한 중요 인접 음차 정보가 모델에 반영되도록 하였다.

[그림 4]는 Distance LSTM 모델에 CNN 구조를 추가한 Distance LSTM-CNN 모델의 구성도이다. 또한, Batch Size에 의존하지 않으며, 학습 수렴 속도를 빠르게 하고, 일부 연구에서는 품질 향상이 있는 계층 정규화(Layer Normalization) 기법을 LSTM에 추가적으로 적용했다[19].

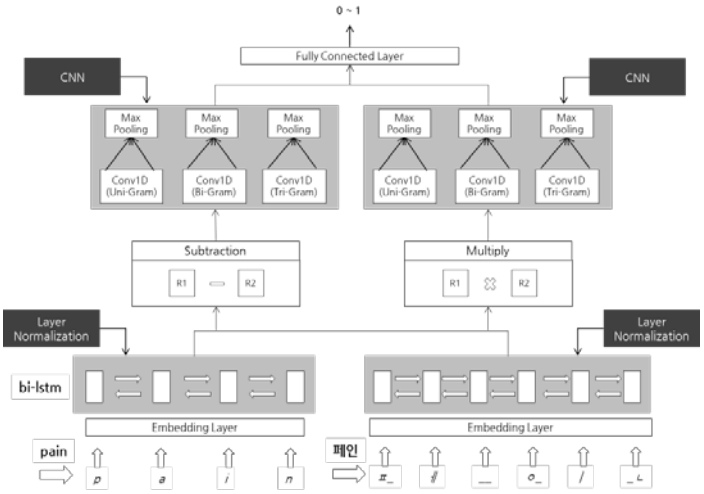


그림 4. Distance LSTM-CNN with Layer Normalization 모델

4. 실험

4.1 실험 환경

모든 모델은 공정한 비교를 위해 동일하게 파라미터를 설정했다. 일반적으로 딥러닝은 많은 파라미터를 가지고 있으며, 파라미터 값의 설정에 따라 조금씩 품질 차이가 나므로, 가장 좋은 품질을 보이는 파라미터를 실험을 통해 찾아 설정하였다.

입력 층은 외국어는 알파벳 단위, 한국어는 자소 단위로 입력을 수행했다. 입력 층은 서로 다른 언어로 이루어져 있으며 음차 표기를 위한 교차언어 데이터는 존재하지 않으므로 임베딩 벡터(Embedding vector)에 대한 전처리(Pre-Training)는 따로 수행하지 않고 학습 데이터에 의해 임베딩 벡터가 학습되도록 구성했다. 임베딩 벡터 차원과 은닉 계층 차원은 128 차원을 사용, 과적합 방지를 위한 Dropout 비율은 0.2, 그리고 활성(Activation) 함수는 마지막 계층에서만 Sigmoid를 사용했으며, 나머지 계층에서는 Relu를 사용했다. 또한, Fully Connected Layer는 1개의 계층만 쌓아 실험을 진행했다.

추가적으로, Distance LSTM-CNN에서 CNN을 위한 필터(Filter)차원은 128차원으로 설정했으며, Dropout 비율은 0.5를 사용했다. Sequence-to-Sequence with Attention 모델은 가장 좋은 품질을 보인 버킷 개수 1개로 고정했으며, 상위 N개는 30개를 추출했다.

마지막으로, 딥러닝 모델들은 오류 역전파 알고리즘에 의해 학습되는데 본 논문에서는 Adam 기법을 이용하여 파라미터를 최적화했다.

4.2 학습 및 평가 데이터

웹 문서에서 자주 사용되는 음차 표기 후보를 추출하기 위해 웹 문서 및 사용자 검색 질의를 이용, “AOA(초아)”, “memento(메멘터)” 등의 패턴을 적용하여 실험을 위한 음차 표기 데이터를 추출했다. 추출된 음차 후보들은 다양한 난도를 가질 수 있으며, “AOA - 초아” 같은 형태는 쉽게 음차 판별이 가능하므로 최대한 반영하지 않고 “memento - 메멘터”와 같이 모호하고 어려운 형태만 추출할 수 있도록 변형된 KODEX 알고리즘을 이용, 다음과 같은 절차를 거쳐 데이터를 추출했다.

1. 웹 문서에서 자주 사용되는 음차 표기 패턴(wikipedia(위키피디아), (concern(관심))을 찾아 음차 표기 대역 쌍 후보를 추출
2. 영어-한국어간 음차 판별을 위해 변형된 KODEX 알고리즘을 이용하여, 음차 표기 대역 쌍 후보간 유사도를 계산하고, 편집 거리가 0, 1인 난도가 높은 음차 데이터를 추출(예시 : concern - 관심, memento - 에멘토 등)
3. 3명의 검수자가 추출된 음차 표기 대역 쌍 후보를 검수, 정답과 오답을 판별하여 검수 데이터 구축

실험에 사용된 검수 데이터는 음차 15,093개, 비음차 15,093개이며 학습, 개발, 평가 셋을 각각 8:1:1 비율로 나누어 평가 셋을 통해 품질을 측정하였다.

실험에 사용된 평가 셋에 포함된 데이터 예시는 [표 1]과 같다.

표 1. 웹 문서로부터 추출된 난도가 높은 음차 후보

외국어	한국어	음차여부
ibm	아이비엠	1
bas	버스	0
glam	그램	0
you light up my life	유라이트업마이라이프	1
lcd tv	엘씨디티비	1
civil war	씨빌워	1

[표 1]을 보면 일반단어, 개체명, 약어성, 복합명사 등 난도가 비교적 높은 다양한 형태의 음차 표기 대역쌍이 추출된 것을 확인 할 수 있다.

4.3 품질 비교

음차 표기 대역 쌍 판별 모델의 품질 평가를 위해 변형된 KODEX 및 Sequence-to-Sequence with Attention 모델, 그리고 Distance LSTM 모델 및 Distance LSTM 모델에 CNN 구조를 추가한 Distance LSTM-CNN(+LN) 모델과의 결과를 비교하였다. 평가 척도는 정확률(Precision), 재현율(Recall)을 사용했으며, Distance LSTM 및 Distance LSTM-CNN(+LN) 결과는 0~1 사이의 확률 값으로 나오게 되며 0.5 이상의 값이 나올 경우 음차로 판별했다. 용도에 따라 임계 값을 변경하는 것에 의해 정확률과 재현율을 조절하여 결과를 유연하게 추출할 수 있다.

[표 2]는 음차 표기 대역 쌍 판별 모델의 평가 결과를 보여준다.

표 2. 모델별 음차 표기 대역 쌍 판별 결과

모델	정확률	재현율	F1 점수
변형된 KODEX	57.02	55.69	53.67
Seq2Seq with Attention	86.18	86.19	86.18
Distance LSTM	87.30	86.87	86.89
Distance LSTM-CNN	89.75	89.52	89.55
Distance LSTM-CNN(+LN)	89.69	89.70	89.70

[표 2]에서 볼 수 있듯이, 변형된 KODEX를 이용하여 편집거리가 작은 데이터를 추출하였기 때문에 난도가 높은 데이터가 대부분이며, 이러한 이유로 인해 변형된 KODEX의 품질이 53.67%로 비교적 낮은 것을 확인할 수 있다. 또한, 다국어 음차 생성에서 높은 품질을 보이는 음차 생성 모델을 음차 판별 모델로 변형한 Sequence-to-Sequence with Attention 모델은 86.18%의 품질을 보였으며, 두 질의의 관련성을 판별하는 연구에서 좋은 품질을 보이는 Distance LSTM 모델을 개선한 모델이 86.89%로 더 높은 품질을 보이는 것을 확인할 수 있었다. 그리고 음차의 시퀀스적인 특성을 반영하는 CNN 구조를 추가한, 음차 표기 대역 쌍 판별에 특화된 Distance LSTM-CNN 모델은 89.55%로 Distance LSTM 모델과 비교해 약 3%의 품질 향상을 보였다. 마지막으로 계층 정규화(Layer Normalization)를 추가한 Distance LSTM-CNN(+LN)모델과 Distance LSTM-CNN 모델과의 비교 결과 품질이 미미하게 향상(89.70%) 되었으며, epoch가 60 -> 21로 약 40번의 epoch 가 절약됨으로써 학습 시 수렴 속도를 개선하는 계층 정규화(Layer Normalization)의 효과를 확인할 수 있었다.

4.4 결과 검토

변형된 KODEX와 같이 음성적 유사도의 편집거리를 이용하는 기존 방법은 대부분 자음만을 이용하여 음차 표기 대역 쌍 여부를 판별한다. 예를 들어 “korea - 고려(X)”의 경우 ‘k’는 ‘ㄱ’과 매칭되며, ‘o’는 모음이므로 무시, ‘r’과 ‘ㄹ’이 매칭되고, ‘e’, ‘a’는 모음이므로 무시하여 최종적으로 음차 표기가 동일하다고 판단하는 문제가 있었으며, 이외에도 ‘ibm - 아이비엠(0)’ 등 약어성 단어를 판별하지 못하는 문제가 있었다.

반면, 딥러닝을 이용한 방법들은 모음 및 시퀀스 정보를 추가로 반영하기에 “youtube - 유트브(X)” 등의 난도가 높은 잘못된 음차 표기까지 정확히 판별할 수 있었다.

Sequence-to-Sequence with Attention 모델은 생성모델의 특성상 긴 단어나 복합명사인 ‘for the first time - 포더퍼스트타임(0)’, ‘you light up my life - 유라이트업마이라이프(0)’, ‘identification - 아이덴티피케이션(0)’와 같은 음차 표기에 대해서 낮은 품질을 보였으며, 편집거리 알고리즘을 이용하여 해당 문제를 일부 해결할 수 있지만 잘못된 음차 표기까지 추출될 수 있는 문제가 있었다.

마지막으로, Distance LSTM-CNN(+LN) 모델은 순서를 유지하면서 주위 문맥의 중요 정보를 추상화하는 자질을 추가, 적용함으로써 Distance LSTM 모델에서 잘못 판단하는 난도가 높은 “bubble love - 버블리러브(X)”, “vans - 반즈(X)”와 같은 음차 표기 대역 쌍 후보를 정확히 판별하는 것을 확인할 수 있었다.

5. 결론

본 논문에서는 검색 품질 향상을 위해 문서에서 자주 사용되는 다양한 음차 표현을 언어 자원으로 구축하기 위한 딥러닝 기반 음차 표기 대역 쌍 판별 모델을 제안했다. 제안하는 모델을 평가하기 위해 정보 검색에서 사람들이 자주 사용하는 음차 표기 데이터를 웹 문서에서 수집하고, 수집된 음차 표기 데이터 중 판별 난도가 높은 데이터 위주로 데이터 셋을 구축하여 모델의 실용성을 평가했다. 평가 결과, 최종적으로 제안하는 음차 표기 대역 쌍 판별에 특화된 모델인 Distance LSTM-CNN(+LN)은 한국어-영어간 음차판별을 위해 변형된 KODEX 방법과의 비교 결과, 약 35%의 품질 향상을 보였으며, 다국어 음차 생성에서 높은 품질을 보이는 음차 생성 모델을 음차 판별 모델로 변형한 Sequence-to-Sequence with Attention 모델과의 비교에서도 약 3.5%의 품질향상을 보였다. 또한 두 질의의 관련성을 판별하

는 연구에서 높은 품질을 보이는 딥러닝 모델인 Distance LSTM 모델을 개선한 모델과 비교해, 음차의 중요 자질을 부각시키는 CNN 구조를 추가함으로써 약 3% 정도의 품질 향상을 보여, 최종적으로 89.70%의 품질을 보였다. 마지막으로 제안하는 모델에 계층 정규화(Layer Normalization) 기법을 적용함으로써 미미한 품질 향상과 더불어 학습 시 약 40번의 epoch가 절약됨으로써 학습 시 수렴 속도를 개선하는 계층 정규화(Layer Normalization)의 효과를 확인할 수 있었다.

참고문헌

[1] 이재성, “다국어 정보검색을 위한 영-한 음차 표기 및 복원 모델”, 박사학위논문, 한국과학기술원 전산학과, 1999.

[2] 이희승, 안병희, 고친판 “한글 맞춤법 강의”, 신구문화사, 1994.

[3] 오종훈, 배선미, 최기선 “자동 음차 표기를 이용한 영-한 음차 표기 대역 쌍의 자동 추출”, 정보과학회논문지(B), 제31권, 제1호, pp. 928-930, 2004.

[4] Andrew Finch, Lemao Liu, Xiaolin Wang, and Eiichiro Sumita. Target-bidirectional neural models for machine transliteration. In Proc. of NEWS, pages 78-82, 2016.

[5] 김태일 ,“최대 엔트로피 모델을 이용한 다국어 정보검색에서의 영-한 음차 표기 모델” , 석사학위논문 , 서강대학교, 1999.

[6] 오종훈, 최기선, “자소 및 음소 정보를 이용한 영어-한국어 음차 표기 모델”, 정보과학회논문지(B), 제32권, 제4호, pp. 312-326, 2005

[7] Brill E., Gary Kacmarcik, Chris Brockett, Automatically Harvesting Katakana-English Term Pairs from Search Engine Query Logs. NLPRS 393-399, 2001.

[8] 강병주, 이재성, 최기선, “외국어 음차 표기의 음성적 유사도 비교 알고리즘”, 정보과학회논문지(B), 제26권, 제10호, pp. 1237-1246, 1999.

[9] Jan, EA-EE., Ge, Niyu, Lin., Shih-Hsiang., Roukos, salim., Sorensen, Jeffrey. A novel approaxh for proper name transliteration verification, ISCSLP, 89-94, 2010.

[10] Sutskever, I. Vinyals, O., Le. Q. V. Sequence to sequence learning with neural networks. , In Proc. Advances in Neural Information Processing Systems, 27, 3104-3112, 2014.

[11] Hochreiter, S. & Schmidhuber, J. Long short-term memory., Neural Comput. 9, 1735-1780, 1997.

[12] Samuel R Bowman, Gabor Angeli, Christopher Potts, and Christopher D Manning. A large annotated corpus for learning natural language inference. arXiv preprint arXiv:1508.05326, 2015.

[13] https://data.quora.com/First-Quora-Dataset Release-Question-Pairs

[14] Lili Jiang, Shuo Chang, and Nikhil Dandekar. "Semantic Question Matching with Deep Learning.", 2017.

[15] Ankur P Parikh, Oscar Tackstrom, Dipanjan Das, and Jakob Uszkoreit. A decomposable attention model for natural language inference. In Proceedings of EMNLP, 2016.

[16] Shuohang Wang and Jing Jiang. 2016. A Compare-Aggregate Model for Matching Text Sequences. CoRR abs/1611.01747, 2016.

[17] Wang, Z.; Hamza, W.; and Florian, R. Bilateral multiperspective matching for natural language sentences. In IJCAI, 2017

[18] Y. Kim, Convolutional Neural Networks for Sentence Classification, Conference on Empirical Methods in Natural Language Processing, 2014.

[19] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. Layer normalization. arXiv preprint arXiv:1607.06450, 2016.

LSTM을 이용한 한국어 이미지 캡션 생성

박성재○, 차정원
창원대학교
tjdwo1289@gmail.com, jcha@changwon.ac.kr

Generate Korean image captions using LSTM

Seong-Jae Park○, Jeong-Won Cha
Changwon National University

요 약

본 논문에서는 한국어 이미지 캡션을 학습하기 위한 데이터를 작성하고 딥러닝을 통해 예측하는 모델을 제안한다. 한국어 데이터 생성을 위해 MS COCO 영어 캡션을 번역하여 한국어로 변환하고 수정하였다. 이미지 캡션 생성을 위한 모델은 CNN을 이용하여 이미지를 512차원의 자질로 인코딩한다. 인코딩된 자질을 LSTM의 입력으로 사용하여 캡션을 생성하였다. 생성된 한국어 MS COCO 데이터에 대해 어절 단위, 형태소 단위, 의미형태소 단위 실험을 진행하였고 그 중 가장 높은 성능을 보인 형태소 단위 모델을 영어 모델과 비교하여 영어 모델과 비슷한 성능을 얻음을 증명하였다.

주제어: 이미지 캡션 생성, Deep Learning, LSTM, CNN

1. 서론

스마트폰과 각종 센서들의 상용화로 인해 이미지 데이터의 양이 증가함에 따라 이미지 데이터의 활용성이 증가하고 있다. 그 중 이미지 캡션 생성기술은 이미지를 설명하는 텍스트를 생성하는 기술로 이미지에서 객체 인식 및 자연어 처리 기술에서 문장 생성의 기술을 필요로 하는 기술이지만 영어권에서는 이미 MicroSoft에서 시각장애인들을 위해 주변 환경을 설명해주는 씨잉 AI 앱[1]을 출시하는 등 상용화를 앞두고 있다.

그러나 한국어 이미지 캡션 생성기술의 경우 기계학습이나 딥러닝을 적용할 만한 한국어 이미지 캡션 데이터가 부족하다.

본 논문에서는 영어 이미지 캡션 데이터를 번역하여 한국어 데이터를 생성하였다. 딥러닝을 사용할 때 한국어 특성에 맞는 캡션 모델을 찾기 위해서 어절별, 형태소별, 의미형태소별 실험을 통해서 최적의 생성 모델을 찾았다.

2. 관련 연구

이미지 캡션 생성에 대해서는 다음과 같은 연구가 있었다. Multi-modal RNN을 이용한 방법[2]에서는 이미지 모델과 언어모델 그리고 두 모델을 통합하는 통합모델 총 3가지 모델을 이용해 이미지 캡션 생성을 진행하였다. 그리고 2015 Image Captioning Challenge에서 구글이 제안한 방법[3]은 CNN과 LSTM을 이용해 이미지 캡션을 생성하였는데 CNN은 Inception V3를 이용하였다. 또한 최근 CNN과 SVM을 함께 사용해 이미지 처리에 높은 성능을 보인 R-CNN과 RNN을 이용한 방법[4] 등 다양한 방법이 연구되었다. 한국어 이미지 캡션 생성은 CNN과 LSTM RNN의 변형인 GRU를 이용하는 방법[5]이 제안되었는데 이 연구에서는 Flickr 8K 데이터를 대상으로 영어 이미지 캡션을 번역자가 번역해 사용하였다. 따라서 [5]에서는 번역자가 직접 영어 데이터를 번역해야하는 단점이 존재한다.

3. 제안 방법

한국어 캡션 데이터를 생성하기 위해서 기존 영어 캡션 데이터를 번역을 이용해 한국어로 변환하였다.

영어권에서는 단어 단위를 입력으로 사용하여 캡션을 생성하였다. 하지만 한국어 모델의 경우 어절 단위로 입력을 사용하게 되는 경우 조사에 따라 같은 단어도 다른 단어로 인식하는 문제가 존재한다. 따라서 기존 데이터에 비해 학습데이터의 양이 훨씬 많아야 하는 문제가 있다.

어절 단위 학습데이터와 형태소분석을 거친 형태소 단위 학습데이터를 생성하였고 추가적으로 문장의 의미를 생성하는 것은 기능형태소가 아닌 의미형태소라고 판단되어 의미형태소만 남긴 데이터를 구축하였다.

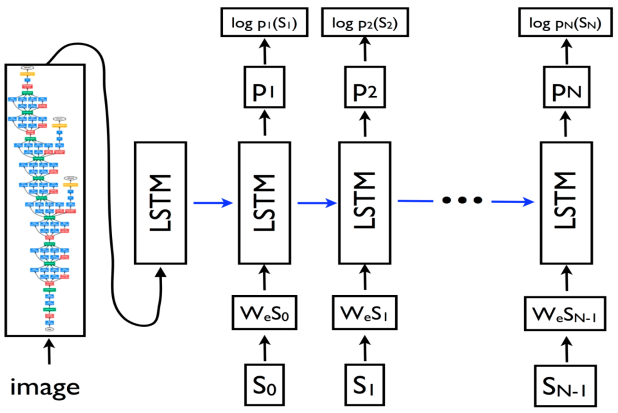


그림 1 이미지 캡션 생성 모델 구조도

본 논문에서는 그림 1과 같이 CNN과 LSTM을 사용한 이

미지 캡션 생성 모델[3]을 사용한다. 입력 이미지를 CNN을 이용하여 512차원의 자질로 인코딩하며 인코딩된 자질을 이용하여 문장을 생성하는 LSTM으로 구성되어있다. CNN은 이미지 구글의 Inception V3를 사용하여 마지막 히든 레이어의 값을 LSTM의 입력으로 사용하였다.

4. 실험 및 토의

본 논문에서는 한국어 이미지 캡션 생성을 위해 MS COCO데이터 셋을 사용하였다. MS COCO 데이터 셋은 123,287개의 이미지와 하나의 이미지당 5문장의 캡션으로 총 616,435문장으로 구성된다. 이 중 117,211개의 이미지를 학습에 사용하였고 2,025개의 이미지를 검증에 사용하였으며 4,051개의 이미지를 테스트에 사용하였다.

제안 방법과 같이 생성된 어절 단위, 형태소 단위, 의미형태소 단위 학습데이터를 이용해 step을 각 200,000번으로 설정하여 학습 후 실험을 진행하였다.

성능평가에는 기계번역에서 성능지표로 사용하는 BLEU score를 사용하였다.

표 1은 학습 데이터 유형별 실험 결과이다. 각 모델별 실험 결과를 비교했을 때 형태소 단위의 실험이 가장 성능이 높고 어절 단위 실험이 가장 성능이 낮은 것을 확인할 수 있다.

표 1 학습데이터 유형별 실험결과

Model	B-1	B-2	B-3	B-4
어절 단위	0.289	0.190	0.141	0.111
의미형태소 단위	0.597	0.392	0.286	0.225
형태소 단위	0.615	0.430	0.322	0.251

46. 이창수, 천주룡, 김주근, 김태일, 강인호, Naver Search, "Distance LSTM-CNN with Layer Normalization을 이용한 음차 표기 대역 쌍 판별"

가장 낮은 성능을 보인 어절 단위 모델의 경우 조사에 따라 동일 단어가 다른 단어로 인식하기 때문에 데이터의 부족으로 인한 오류가 많이 발생한 것으로 생각된다. 형태소 단위 모델이 의미형태소 단위 모델보다 성능이 좋은 것은 기능형태소가 제한적인 어휘를 가지기 때문에 예측 성능이 높게 나타나기 때문이라고 생각된다. 이를 증명하기 위한 추가 실험으로 형태소 단위 실험결과의 정답과 예측 데이터를 각각 의미형태소와 기능형태소만 남기고 성능을 측정하였다. 표 2의 결과를 보면 기능 형태소의 성능이 높은 것을 확인할 수 있다.

표 2 형태소 단위 모델의 세부 실험 결과

Model	B-1	B-2	B-3	B-4
의미형태소	0.547	0.366	0.280	0.230
기능형태소	0.686	0.457	0.346	0.279

따라서 이 중 가장 높은 성능을 보인 형태소 단위 모델을 step을 400,000번으로 증가시켜 학습을 진행하고 영어 모델은 step을 1,000,000번으로 증가 시켜 학습해 두 모델의 성능을 비교하였다.

표 3은 형태소단위, 영어 모델의 실험결과와 MS COCO Image Captioning Challenge에서 높은 성능을 보인 5개의 모델의 결과이다. 400,000번 학습하여 생성한 형태소 단위 모델의 성능이 학습량이 적음에도 불구하고 영어 모델의 성능보다 B-4 score로 0.100 더 높음을 확인하였으며 나머지 MS COCO Image Captioning Challenge에서 높은 성능을 보인 모델과 비교하더라도 크게 낮지 않은 성능을 보임을 확인하였다.

표 3 실험 결과

Model	B-1	B-2	B-3	B-4
형태소 단위	0.630	0.445	0.333	0.260
영어	0.611	0.441	0.323	0.250
NIC[3] (google)				0.309
MSR Captivator[6]				0.308
m-RNN(2) [2]				0.302
m-RNN [2]				0.299
MSR [7]				0.291

그림 2와 3은 출력결과의 예를 보여준다.

(가) (나) (다)

그림 3 MS COCO 평가 이미지 중 올바른 결과의 예

그림 2를 대상으로 형태소 모델과 영어 모델이 생성한 캡션은 가)의 경우 “야구선수가 공을 치려고 합니다.” 와 “a group of people on a field paying baseball.” 로 형태소 모델과 영어 모델 모두 이미지를 잘 설명하는 캡션을 생성한 것을 확인할 수 있다. 나)의 경우 “고양이는 나무벤치에 앉아있다.” 와 “a cat sitting on top of a wooden bench.” 를 캡션으로 생성하였고 다)의 경우 “바나나의 큰 무리가 시장에 있습니다.” 와 “a market with a bunch of bananas hanging from the ceilling” 을 생성하였다. 그림 2의 세 개의 이미지를 대상으로 형태소 모델과 영어 모델이 생성한 캡션이 이미지를 잘 설명하고 있는 것을 확인 할 수 있다.

(가) (나) (다)

그림 4 MS COCO 평가 이미지 중 오류 결과의 예

그림 3을 대상으로 형태소 모델과 영어 모델이 생성한 캡션은 가)의 경우 “케이크를 절단하는 칼을 들고 여자.” 와 “a man and woman cutting a cake with a knife” 를 캡션으로 생성하였다. 그러나 가)의 정답은 “요리사는 초밥을 만드는 레스토랑에 있습니다” 로 형태소 모델의 경우 “요리사” 를 “여자” 로 , “초밥” 을 “케이크” 로 기술하였다. 이는 “요리사” 와 “초밥” 이 학습 코퍼스에 나타난 빈도가 적어 생기는 문제로 예측되며 실제로 생성된 단어 사전을 보면 “여자” 의 경우 37,482번 , “요리사” 의 경우 671번 발생하였다. 마찬가지로 “케이크” 의 경우 8,894번 발생하였으나 “초밥” 의 경우 62번 발생하였다.

나)의 경우 “테이블에 앉아있는 샌드위치.” 와 “a sandwich sitting on top of white plate” 를 생성하였는데 위와 동일하게 117번 나타난 “콜라” 를 인식하지 못하는 문제가 발생하였다. 다)의 경우 “프리즈비를 들고 잔디에 서있는 어린 소년” 과 “a man in a field with a frisbee” 를 생성하는 등 학습 코퍼스에 저빈도로 나타난 사물을 오 인식하여 잘못된 캡션을 생성하는 문제가 발생하였다.

이렇듯 캡션이 부정확하게 생성되는 경우는 학습데이터의 부족으로 인한 오인식이며 이를 해결하기 위해서는 추가적인 학습데이터를 사용해야 할 것이라고 생각된다.

5. 결론 및 향후연구

본 논문에서는 한국어 이미지 캡션을 생성하기 위해 영어 캡션데이터를 번역을 이용해 한국어로 변환하는 방법을 제안하였다. 또한 어절단위보다 형태소단위로 학습을 진행하는 경우 성능이 더 높음을 보였다.

향후연구로는 LSTM의 입력으로 사용되는 Inception V3가 아닌 VGGNet[8] 또는 이미지 처리에 높은 성능을 보이고 있는 Fast R-CNN등을 사용하는 모델을 적용할 예정이다. 또한 의미형태소로 생성된 캡션을 seq2seq를 이용하여 실제 문장을 생성하는 연구를 진행할 계획이다.

감사의 글

본 연구는 미래창조과학부 및 정보통신기술연구진흥센터의 정보통신·방송 연구개발사업의 일환으로 수행하였음.[2015-0-00219, 개방형 미디어 생태계 구축을 위한 시맨틱 클러스터 기반 시청상황 적응형 스마트방송 기술 개발]

참고문헌

[1] https://www.microsoft.com/en-us/seeing-ai/
[2] MAO, Junhua, et al. Deep captioning with multimodal recurrent neural networks (m-rnn). arXivpreprint arXiv: 1412.6632, 2014.
[3] VINYALS, Oriol, et al. Show and tell : Lessons learned from the 2015 mscoco image captioning challenge. IEEE transactions on pattern analysis and machine intelligence, 2017, 39.4: 652-663
[4] KARPATHY, Andrej; FEI-FEI, Li. Deep visual-semantic aligments for generating image descriptions. In:Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2015. p. 3128-3137
[5] 배장성, 이창기. (2016). 딥러닝을 이용한 한국어 이미지 캡션 생성. 한국정보과학회 학술발표 논문집. , 488-490
[6] J. Devlin, H. Cheng, H. Fang, S. Gupta, L. Deng, X. He, G. Zweig, and M. Mitchell. “Language models for image captioning: The quirks and what works,” in ACL, 2015.
[7] H. Fang, S. Gupta, F. Iandola, R. Srivastava, L. Deng, P. Dollar, J. Gao, X. He, M. Mitchell, J.Platt, C. L. Zitnick, and G. Zweig, “From captions to visual concepts and back,” in CVPR, 2015.
[8] SIMONYAN, Karen; ZISSERMAN, Andrew. Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556, 2014.

● 구두발표 5: 형태소/구문분석

동적 오라클을 이용한 한국어 의존 구문분석*

이경호○, 이공주
충남대학교 전자전파정보통신공학과, 전파정보통신공학과
gyholee@gmail.com, kjoolee@cnu.ac.kr

Korean Dependency Parsing using Dynamic Oracle

Gyoung Ho Lee○, Kong Joo Lee
Chungnam National University

요 약

구문분석은 자연언어처리의 오랜 관심 분야로 다양한 접근방법과 알고리즘이 시도되어 계속 발전하고 있다. 하지만 기존의 접근방법은, 학습단계에서는 정답으로부터 추출된 이전 정보를 사용하고 평가 단계에서는 예측으로 이루어진 정보를 활용한다는 근본적인 차이가 있다. 이러한 차이를 극복하기 위한 다양한 시도가 있었고 그 중 동적 오라클 기법이 합리적인 시간 증가와 성능향상을 보였다. 본 연구에서는 이러한 동적 오라클 기법을 한국어 구문분석에 적용하였다. 동적 오라클 기법을 한국어에 적용할 때 고려해야하는 부분에 대해 탐구하고 실험을 통해 동적 오라클 기법을 한국어 구문분석에 적용하여 결과를 살펴보았다.

주제어: 구문분석, 동적 오라클, 인공신경망, 의존문법

1. 서론

구문분석은 자연언어 처리 분야의 오랜 관심 분야로 여러 접근 방법과 알고리즘이 도입되어 계속 발전해나가고 있다. 최근에는 포인터 네트워크를 활용한 구문분석 모델이나 순환신경망(Recurrent Neural Network)을 이용한 구문분석기 등, 인공신경망을 활용한 다양한 모델이 개발되고 있고 좋은 성능을 보이고 있다[1][2][3]. 한국어는 어순 배열의 자유도가 높고 문장 성분 생략이 빈번한 특성이 있다. 그렇기 때문에 이런 특성에 적합하다고 알려진 의존문법 구문분석이 한국어 구문분석의 주된 연구 대상이 되어 왔다[3, 4].

기존의 의존문법 구문분석 알고리즘은 주로 의존 트리로부터 구문분석 진행 상태(state)와 이 상태에서 어떤 행동(action)을 해야 하는지를 추출하고 상태로부터 행동을 결정하는 모델을 학습한다.

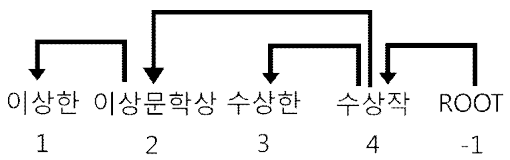


그림 1. 구문트리 예

그림 1과 같은 구문트리의 구문분석은 표 1과 같이 진행된다. 본 연구에서는 한국어 구문분석의 특성인 지배소 후위 원칙과 투사성 원칙을 활용하여 backward 방식의 Arc-Eager 알고리즘[5]으로 구문분석을 진행하였다[1].

표 1. 구문분석 예

Step	Stack	Buffer	Action
0	-1	4, 3, 2, 1	Right-Arc
1	-1, 4	3, 2, 1	Right-Arc
2	-1, 4, 3	2, 1	Reduce
3	-1, 4	2, 1	Right-Arc
4	-1, 4, 2	1	Right-Arc
5	-1, 4, 2, 1		

구문분석은 Stack과 Buffer를 초기화 하는 것으로 시작된다. Stack에는 이미 자신의 지배소를 찾은 단어들이 들어있고 Buffer에는 지배소를 찾을 단어들이 위치한다. Stack의 최상단 단어(표 1의 Stack 열에서 가장 오른쪽)는 Buffer의 최하단 단어(표 1의 Buffer 열의 가장 왼쪽)의 지배소 후보이다. 이 두 단어와 Stack, Buffer의 상태를 이용해 다음에 취할 행동을 결정한다(Action 열). 학습과정에서는 정답트리를 이용하여 두 단어의 관계가 지배소-피지배소 관계인지 알 수 있기 때문에 이 둘을 연결할지(Right-Arc), 현재의 지배소 후보를 Stack에서 버리고 다음 단어로 넘어갈지(Reduce)를 결정할 수 있다. 이러한 행동을 정적 오라클(Static oracle)이라 한다.

학습단계에서는 오라클을 통한 전이를 수행하면서 상태와 행동을 수집하고, 수집된 상태와 행동을 이용해 상태가 주어지면 올바른 행동을 예측하도록 기계학습 모델을 학습시킨다. 실제 구문분석을 수행할 때는 오라클을 알 수 없으므로 학습된 모델에서 예측한 행동에 따라 구문분석을 진행한다.

* 이 논문은 2016년도 정부(미래창조과학부)의 재원으로 한국연구재단의 지원을 받아 수행된 기초연구사업임(No. NRF-2015R1C1A2A01051685)

하지만, 학습과정에서 도달하는 특정 상태 c와 실제 적용 단계의 상태c'은 근본적으로 차이가 있다. 학습과정의 상태 c는 정답으로부터 추출된 올바른 상태와 행동을 통해 도달한 결과이다. 하지만 c'의 경우 모델로부터 예측된 행동을 따라온 상태이기 때문에 올바른 경로를 통해 도달한 상태라는 것을 보장할 수 없다.

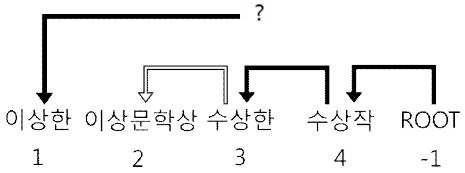


그림 2 모델을 이용한 구문분석 예

그림 2는 모델을 이용한 구문분석을 가정한 예이다. 표 1과 같이 정답 트리를 알고 있다면 '이상문학상'과 '수상한'의 관계가 피지배소-지배소관계가 될 수 없는 것을 알고 $Reduce$를 수행했을 것이다. 하지만 모델의 예측으로 구문분석이 진행된다면, 그림 2와 같이 잘못된 관계를 예측 할 수 있다. 이러한 경우 그 다음 단어인 '이상한'과 '이상문학상'의 관계에도 변화가 생긴다. 만약 학습 단계에서 이런 문장을 보았다면, '수상작'을 지배소로 가지고 있는 '이상문학상'과 '이상한'의 관계가 지배소-피지배소 관계인 것을 학습하였을 것이다. 하지만 예측과정에서는, '이상문학상'의 지배소를 잘못 예측한 결과로, '수상한'을 지배소로 가지는 '이상문학상'과 '이상한'의 관계를 추론해야 하지만 학습단계에서 이러한 관계를 학습하지 않았기 때문에 문제가 될 수 있다.

이러한 문제는 학습단계에서 정답트리로부터 추출된 정적 오라클 경로뿐만 아니라 다양한 경로를 탐색해보고 그것에 대해 학습하도록 함으로써 완화할 수 있다. 그림 2와 같은 경우, 비록 3-2번 단어의 관계는 틀렸지만, 이 상태에서 2-1번 단어의 관계를 학습 할 수 있다면 3-2 단어에서 발생한 에러가 다음단계로 전파되는 것을 최소화 할 수 있다. 하지만 기존의 정적 오라클 방법으로는 이러한 작업을 수행하기 어렵기 때문에 다른 접근법이 필요하다.

본 연구에서는 이러한 문제점의 해결 방법으로 개발된 동적 오라클(Dynamic Oracle)을 한국어 구문분석에 적용해보고자 한다. 동적 오라클을 이용하면 특정 상태에서 적용할 행동을 정답트리를 통해 미리 결정해 놓지 않고 해당 상태에서 적용 가능 하면서 적용하였을 때 남아 있는 정답 트리의 훼손을 최소화하는 행동을 구할 수 있다. 이러한 행동을 통해 지금까지 잘못된 경로를 왔다고 하더라도 다음에 선택하는 행동은 에러의 전파를 막고 현재 상태에서 가능한 최선의 선택을 할 수 있도록 해준다.

이러한 시도 중 하나로 강화학습을 이용한 구문분석연구가 진행된 바 있다[6, 7]. 이들 연구에서는 상태에 따른 행동을 결정하는 정책(policy)을 설정한다. 이 정책에 따라서 행동의 확률을 구하고 이 확률에 따라 트리를 완성해 나간다. 트리가 완성되면, 트리의 완성도에 따라 보상(reward)을 받게 된다. 강화학습은 이 보상이 커지는 방향으로 정책을 발전시켜나간다. 이때 확률적으로 새로운 경로가 개척될 수 있는데 이러한 경로의 보상값을 통해 이전의 경로 외에 새로운 경로가 더 좋은지 나쁜지 탐색하면서 가장 높은 보상을 받을 수 있는 경로를 찾아간다. 하지만 강화학습의 경우, 여러 상태와 행동에 대한 다양한 탐색을 수행해야 하므로 학습 속도가 느린 단점이 있다.

동적 오라클은 학습단계에서의 오라클 설정에만 영향을 미치며 실제 적용에는 기존과 같이 탐욕적 알고리즘(Greedy parsing)으로 적용가능하다. 그렇기 때문에 기존의 구문분석 알고리즘과 학습 모델에 대한 적은 변화로도 적용가능하다. 또한 구문분석에서 발생 가능한 상태와 오류들에 대한 탐색을 수행하기 때문에 강화학습을 통한 학습보다 더 빠른 학습을 수행 할 수 있다.

본 논문의 구성은 다음과 같다. 2장에서는 동적 오라클과 관련된 구문분석 관련 연구에 대해 소개하고 3장에서는 이를 한국어 구문분석에 적용하는 방법과 예측 모델에 대해 설명한다. 그리고 4장에서는 한국어 구문분석에 적용한 실험 결과와 이전의 구문분석 연구들과 비교를 하고 마지막 5장에서 이 연구의 결론을 맺는다.

2 관련 연구

본 연구에서는 동적 오라클을 이용한 구문분석을 한국어 구문분석에 적용하였다. [5]에서 동적 오라클 방법에 대해 처음 제안하였다. 이 논문에서는 동적 오라클의 필요성과 동적 오라클의 정의 방법에 대해 제안하였다. 이를 학습 할 수 있는 Online 학습 절차에 대해 제안하고 이를 Arc-eager 구문분석 방법에 적용하였다. CoNLL 2007데이터에 이를 적용한 실험 결과, 영어의 경우 정적 오라클을 사용 할 때와 동적 오라클을 사용할 때 UAS(Unlabeled Attachment Scores) 점수가 86.24에서 88.81 증가함으로써 동적 오라클이 성능향상에 도움이 되는 것을 증명하였다. [8]의 연구에서는 앞선 [5]연구를 발전시켜 Arc-Eager, Arc-hybrid, Easy-First등의 구문분석 알고리즘에 동적 오라클을 적용하였다. 같은 데이터에 대해 ARC-Eager: 88.69, arc-hybrid: 88.69, easy-first: 89.41의 성능향상을 보였다.

또한 이 논문에서는 arc-standard 알고리즘이 아크 분해(Arc Decomposition)되지 않음을 증명하여 다른 알고리즘에 비해 동적 오라클 적용의 효율이 낮음을 증명하였다. [9] 논문에서는 딥러닝 기법을 이용한 Stack LSTM[10] 알고리즘에 동적 오라클을 적용하였다. 기존의 정적 오라클을 통해 학습한 Stack LSTM보다 동적 오라클을 적용하였을 때 최대 93.56의 성능을 나타냄으로 써 최근의 딥러닝을 이용한 파서와 동적 오라클의 사용이 성능향상을 이끌어 낼 수 있음을 나타내었다.

본 연구에서는 앞서 연구들이 영어나 다른 언어에 적용한 바와 같이 동적 오라클 방법을 한국어 구문분석에 적용해보고자 한다. [9]의 논문과 같이 arc-eager를 사용하며 이 연구에서 적용한 Stack LSTM을 단순화하여 한국어 구문분석을 수행하였다.

3. 본 론

3.1 동적 오라클

본 연구에서는 구문분석의 학습단계와 적용단계에서 발생하는 차이를 줄이기 위한 방법으로 동적 오라클을 적용하였다. 구문분석의 현재 상태(c)에서 전이를 통해 도달 가능한 완성된 구문트리(G) 중, 정답 트리와의 손실($loss$)이 가장 적은 트리를 식 1과 같이 나타낸다[5].

$$\min_{G:c\rightarrow G} Loss(G, G_{gold}) \quad \cdots (1)$$

이때 손실은 정답 트리(G_{gold})의 arc들과 임의의 트리 G의 arc의 차집합 개수를 의미한다.

[5]의 연구에서 상태 c에서 전이 행동 t를 취했을 때 발생하는 비용함수를 다음과 같이 정의한다.

$$COST(t;\ c,\ G_{gold}) = [\min_{G:t(c)\rightarrow G} Loss(G, G_{gold})] - [\min_{G:c\rightarrow G} Loss(G, G_{gold})] \quad \ldots (2)$$

즉, c에서 t를 취할 때의 비용은 현재 상태에서 도달 가능한 가장 좋은 트리(정답 트리와의 차이가 크지 않은 트리)와 상태 t를 취한 새로운 상태에서 선택한 가장 좋은 트리와의 차이를 의미한다. [5] 연구에서 이 c에서 t를 수행할 때 비용(cost)이 0이 되는 t가 존재함을 증명하였으며 이러한 t들을 동적 오라클로 정의하였다. 구문분석의 어느 단계에서든, 만약 정답 트리를 만드는 길에서 벗어났더라도 동적 오라클을 따라간다면 현재 상태에서 도달 가능한 가장 적절한 트리를 생성할 수 있게 된다. [5]와 [8] 연구에서는 c와 정답트리를 이용하여 동적 오라클을 구하는 규칙에 대해 정의하였다. 이들 연구에서 Arc-eager의 4 종류 행동에 대한 규칙을 정의했지만 한국어 구문분석 알고리즘에서는 2 종류(Right－Arc$_{label}$, Reduce) 행동에 대한 규칙만 필요하다.

현재 상태 c를 $S=\sigma|s$, $B=b|\beta$와 같이 표시할 수 있다. 이때 S는 Stack, s는 Stack 최상단, σ는 Sack의 나머지 단어들을 의미하고 B는 Buffer, b는 Buffer의 최하단, β는 Buffer 나머지 부분을 의미한다. 전이 행동 Reduce는 s를 S에서 제거하는 것이다. c에서 이 행동을 취하면 B의 단어 중 s를 지배소로 하는 단어들은 지배소를 찾을 수 없게 된다. 그렇기 때문에 Reduce의 비용은 정답 트리에서 (s,k) 아크의 개수(k: B의 단어 중 s를 지배소로 가지는 단어)이다. Right－Arc$_{label}$는, b의 지배소를 s로 결정하는 것이다. 이렇게 되면 b는 더 이상 σ에서 자신의 지배소를 찾을 수 없다. 그렇기 때문에 정답트리에서 b의 지배소 k가 σ에 있다면 cost 값은 1이 된다. 만약, 앞서 단계에서 잘못된 예측으로 k가 이미 S에서 제거 되었다면, b의 지배소로 어떤 단어로 선택해도 손실은 달라지지 않는다. 그렇기 때문에 이러한 경우 비용은 0이다. 또한 b의 k가 s인 경우 정답트리에 있는 행동이기 때문에 비용에 영향을 주지 않아 비용은 0이다. 본 연구에서는 지배소 후위 원칙과 투사성 원칙에 따르기 때문에 b의 지배소를 β에서 찾는 규칙은 적용하지 않았다.

3.2 모델과 자질

상태로부터 행동을 결정하는 분류기로 [9]와 [10]에서 사용한 Stack LSTM을 단순화한 인공신경망 모델을 사용하였다. 본 연구에서 사용한 모델은 그림 3과 같다.

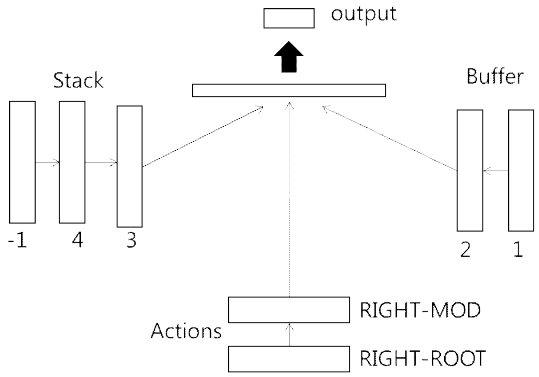


그림 3 분류기 모델

이 모델은 3개의 순환신경망(RNN) 레이어와 1개의 전방향 인공신경망(FNN) 레이어로 구성된다. 3개의 RNN 레이어는 Stack과 Buffer, 그리고 지금까지 예측했던 전이 행동들을 표현하는 벡터를 각각 생성한다. RNN을 통해 각각의 벡터로 표현된 Stack 정보와 Buffer정보, 그리고 행동들의 정보는 하나의 벡터로 연결되어 FNN의 입력으로 사용된다. Stack과 Buffer의 RNN 입력은 각각의 어절을 표현하는 벡터들이다.

하나의 어절은 1) 어절의 첫 번째 형태소와 태그 2)어절의 두 번째 형태소와 태그 3) 어절의 끝에서 두 번째 형태소와 태그 4) 어절의 마지막 형태소와 태그를 이용하여 표현된다. 각 형태소와 태그는 50차원의 벡터로 표현되고 어절은 이들 벡터를 연결하여 400차원의 벡터로 표현되다. 전이 행동은 50차원의 벡터로 표현한다. 이들은 Stack, Buffer의 입력 순서와 모델이 예측한 순서대로 RNN 레이어에 입력된다. 각각의 어절과 전이 행동을 입력받는 RNN은 400차원의 출력을 가지며 2개의 layer를 가진 GRU를 사용하였다. 이들 RNN의 출력은 하나의 벡터로 연결되어 1200 차원의 히든 레이어를 가진 FNN에 입력된다. 한국어 구문분석의 경우, Arc-Eager 알고리즘을 사용하면서 backward로 구문분석을 수행할 경우 Right-Arc와 Reduce만을 이용하여 구문분석을 수행할 수 있는 장점이 있다. 그렇기 때문에 이 모델의 출력은 $Reduce$와, Right－Arc$_{label}$들을 출력으로 가진다.

3.3 학습 알고리즘

본 연구에서 사용한 학습 알고리즘은 표 2와 같다. 이 알고리즘은 학습데이터에 있는 문장 W과 그 문장의 정답트리 T를 이용하여 학습을 진행한다. 먼저 W를 이용하여 구문분석의 상태를 초기화 한다(c). c에는 현재 진행 중인 Stack과 Buffer 그리고 지금까지 채결된 arc정보를 담고 있다. c의 상태에 따라 구문분석의 진행 여부를

판단한다. 구문분석은 먼저 해당 상태에서 선택 가능한 행동들(lt)의 수집한다(line 5). 그리고 현재 상태와 lt를 이용하여 3.1절에서 설명한 동적 오라클과 현재 모델의 각 전이 행동에 대한 예측 확률을 수집한다(line 6, 7). 알고리즘은 학습단계에서 구문분석이 동적 오라클의 경로를 따라가다가 이따금씩 새로운 경로를 시도해보도록 설계되었다. 다음번 전이를 위한 행동 nt는 line 9와 같이 정의된다. 구문분석은 80%의 확률로 동적 오라클의 경로를 따라간다. 나머지 경우, lt중 하나를 선택하여 진행한다. 학습이 진행됨에 따라 모델의 예측 정확도가 높아지면서 모델로부터 가장 높은 점수를 받는 행동이 수렴된다. 이런 경우 새로운 경로를 개척할 수 있는 기회가 줄어든다. 이를 방지하기 위해 일정한 확률로 lt 중 임의의 하나의 행동을 선택해 구문분석을 진행하도록 하였다.

구문분석을 수행하는 동안 상태와 그 상태에 대한 동적 오라클을 수집한다(line 10). 수집된 상태와 오라클을 이용하여 일정 주기마다 모델을 학습시켜 준다(line 12). 상황에 따라 동적 오라클은 1개 이상의 오라클을 가질 수 있다. 이를 학습하기 위해 multilabel soft-margin loss[12]를 이용하였다.

표 2. 학습 알고리즘

1: $FeatureList = \{\varnothing\}$
   $\mathrm{model} = ModelInitial()$
2: $for\ (W,\ T) \in TrainingData:$
3: $\quad c \leftarrow Initial(W)$
4: $\quad while$ not Terminal(c):
5: $\quad\quad lt \leftarrow LegalTransition(c)$
6: $\quad\quad dynamic \leftarrow Dynamic(c,\ lt)$
7: $\quad\quad actionProb \leftarrow \mathrm{model}(c)$
8: $\quad\quad \alpha = random(),\ \beta = random()$
9: $\quad\quad nt = \begin{cases} choice(dynamic) & \text{if } \alpha < 0.8 \\ \max(lt,\ actionProb) & \text{if } \alpha \geq 0.8 \text{ and } \beta < 0.8 \\ choice(lt) & else \end{cases}$
10: $\quad\quad FeatureList \leftarrow \{\mathrm{c,dynamic}\} \cup \mathrm{FeatureList}$
11: $\quad\quad Next(c,\ nt)$
12: $\quad\quad for\ each\ every\ N\ step\ do:$
$\quad\quad\quad Train(\mathrm{model},\ FeatureList)$

4. 실험

동적 오라클을 한국어 구문분석에 적용하였을 때의 효과를 알아보기 위한 실험을 수행하였다. 실험에는 21세기 세종[13]의 구구조 구문분석 트리를 의존 구조 구문분석 트리로 변환한 결과를 사용하였다. 그리고 용언들 사이의 관계를 [14]의 참조하여 재조정하였다. 변환과정 후 2개 이상의 노드를 가진 트리를 수집하여 총 57,912개의 트리를 얻었고 이를 9:1로 나누어 평가와 학습과 평가에 각각 52,120, 5,792개의 트리를 사용하였다.

표 3은 본 연구의 베이스라인과 이전 연구, 그리고 동적 오라클을 사용한 모델의 구문분석 성능 표이다.

표 3. 성능 비교

Algorithms	UAS	LAS
Baseline	90.65	88.97
Dynamic Oracle	90.97	89.30
Pointer Network[2]	91.65	89.34
Stack LSTM[3]	90.83	88.49

표 3은 평가 데이터의 전체 [피지배소, 전이 행동, 지배소] 목록 중, 피지배소의 지배소를 맞게 찾은 비율(unlabeled attachment score, UAS)와 전이 행동과 지배소까지 맞게 찾은 경우(labeled attachment score, LAS)비율을 백분율로 나타낸 결과이다. 표 3의 Baseline은 동적 오라클을 적용한 모델과 같은 모델에서 정적 오라클을 사용한 결과이다. Pointer Network와 Stack LSTM은 각각 구문분석에 딥러닝 알고리즘을 적용한 모델로 후자는 전이기반 알고리즘을 기반으로 한다. 평가 결과, 동적 오라클을 사용하였을 때, 정적 오라클을 사용한 결과보다 좀 더 높은 결과를 나타내었다.

표 4는 문장의 어절 수가 2~10개, 10~15개, 15~20개인 문장들을 수집하고 수집된 문장들에 대한 정적 오라클과 동적 오라클의 UAS 차이를 비교한 표이다.

표 4. 어절 길이별 UAS 점수

어절 길이	문장 수	정적오라클	동적 오라클	차이
2~10	2,725	95.10	95.21	0.11
10~15	1,363	92.17	92.34	0.17
15~20	759	89.71	90.11	0.40

표 4에서 문장의 길이가 길어질수록 정적오라클과 동적오라클의 성능 차이가 벌어지는 것을 볼 수 있다. 앞에서 발생한 오류의 영향을 줄여주는 동적오라클의 효과가 긴 문장일수록 발휘되기 유리하기 때문에, 문장의 길이가 길어질수록 두 방법의 성능차이가 벌어지는 것으로 생각된다.

5. 결론

본 연구에서는 구문분석의 학습단계와 적용단계의 차이로부터 발생할 수 있는 문제점을 줄여 줄 수 있는 동적 오라클에 대해 탐구하고 이를 한국어 구문분석에 적용하기 위한 방법을 연구하였다. 연구 결과 동적 오라클을 사용했을 때 정적 오라클 사용보다 더 나은 성능을 나타냄을 알 수 있다. 이러한 결과는 기존의 정적 오라클이 탐색하지 못한 영역을 동적 오라클을 통해 탐색 할 수 있기 때문에 적용단계에서 발생하는 에러 상황에 대한 강건한 대처가 가능했기 때문으로 생각된다. 향후연구로 동적 오라클을 이용하면서 더 다양한 경로를 효율적으로 탐색 할 수 있도록 강화학습이나 모방학습 등을

한국어 구문분석에 적용해볼 계획이다.

참고 문헌

[1] 이창기, 김준석, and 김정희. "딥 러닝을 이용한 한국어 의존 구문분석." 제 26 회 한글 및 한국어 정보처리 학술대회 (2014): 87-91.

[2] 박천음, and 이창기. "멀티 태스크 학습 기반 포인터 네트워크를 이용한 한국어 의존 구문분석." 한국정보과학회 학술발표논문집 (2016): 440-442.

[3] 나승훈, et al. "순환 컨트롤러를 이용한 Stack LSTM 기반 한국어 의존 파싱." 한국정보과학회 학술발표논문집 (2016): 446-448.

[4] 이건일, and 이종혁. "인공 신경망을 이용한 형태소 기반 한국 H 의존 구문분석.", 동계학술발표회논문집 (2014)

[5] Goldberg, Yoav, and Joakim Nivre. "A Dynamic Oracle for Arc-Eager Dependency Parsing." COLING. 2012.

[6] Zhang, Lidan, and Kwok Ping Chan. "Dependency parsing with energy-based reinforcement learning." Proceedings of the 11th International Conference on Parsing Technologies. Association for Computational Linguistics, 2009.

[7] Lê, Minh, and Antske Fokkens. "Tackling Error Propagation through Reinforcement Learning: A Case of Greedy Dependency Parsing." arXiv preprint arXiv:1702.06794 (2017).

[8] Goldberg, Yoav, and Joakim Nivre. "Training deterministic parsers with non-deterministic oracles." Transactions of the association for Computational Linguistics 1 (2013): 403-414.

[9] Dyer, Chris, et al. "Transition-based dependency parsing with Stack long short-term memory." arXiv preprint arXiv:1505.08075 (2015).

[10] Dyer, Chris, et al. "Transition-based dependency parsing with Stack long short-term memory." arXiv preprint arXiv:1505.08075 (2015).

[11] Chen, Danqi, and Christopher Manning. "A fast and accurate dependency parser using neural networks." Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP). 2014.

[12] Zanoci, C., & Andress, J. Exploring CNN-RNN Architectures for Multilabel Classification of the Amazon.

[13] 국립국어원. 21세기세종계획. 2012.

[14] 의존 구문분석 말뭉치 구축을 위한 의존 관계 태그 세트 및 의존 관계 설정 방법, 정보통신단체표준, TTA, 2015

멀티 레이어 포인터 네트워크를 이용한 한국어 의존 구문 분석

박천음*O, 황현선*, 이창기*, 김현기**
강원대학교*, 한국전자통신연구원**
{parkce, hhs4322, leeck}@kangwon.ac.kr, hkk@etri.re.kr

Korean Dependency Parsing with Multi-layer Pointer Networks

Cheoneum Park*O, Hyunsun Hwang*, Changki Lee*, Hyunki Kim**
Kangwon National University*, Electronics and Telecommunications Research Institute**

요 약

딥 러닝 모델은 여러 히든 레이어로 구성되며, 히든 레이어의 깊이가 깊어질수록 레이어의 벡터는 높은 수준으로 추상화된다. 본 논문에서는 Encoder RNN의 레이어를 여러 층 쌓은 멀티 레이어 포인터 네트워크를 제안하고, 멀티 태스크 학습 기반인 멀티 레이어 포인터 네트워크를 이용한 한국어 의존 구문 분석 모델을 제안한다. 멀티 태스크 학습 모델은 어절 간의 의존 관계와 의존 레이블 정보를 동시에 구하여 의존 구문 분석을 수행한다. 실험 결과, 본 논문에서 제안한 모델이 기존 한국어 의존 구문 분석 연구들 보다 좋은 UAS 92.16%, LAS 89.88%의 성능을 보였다.

주제어: 의존 구문 분석, 멀티 레이어 포인터 네트워크, Multi-layer Pointer Networks, 딥 러닝

1. 서론

구문 분석은 문장성분 사이의 관계를 분석하고 문장의 구조적, 의미적 중의성을 해결하는 자연어처리 문제이며, 의존 구문 분석(Dependency parsing)과 구구조 구문 분석(Phrase structure parsing) 등이 있다. 의존 구문 분석은 문장구조를 중심어(head)와 수식어(modifier)로 구성된 의존 관계로 표현하는 방법이며[1], 어순이 자유롭고 문장성분의 생략이 빈번한 한국어와 같은 언어에서 주로 연구되었다. 최근 딥 러닝 기반 의존 구문 분석은 RNN (Recurrent neural network)를 이용한 전이 기반 방법[2-4]과 그래프 기반 방법[5-9]이 활발하게 연구되고 있다. 의존 구문 분석은 의미분석(의미역 결정 등)과 담화분석(상호참조해결), 질의응답 등에 응용될 수 있다.

포인터 네트워크(Pointer networks)[10]는 어텐션 메커니즘(Attention mechanism)[11]을 이용하여 입력열에 대응되는 위치를 출력 결과로 하는 RNN의 확장된 모델이며, 입력된 문장에서 특정 단어를 가리켜 찾는 문제에 적합하다. 어텐션 메커니즘은 주어진 입력 열 중에서 출력 결과에 영향을 미치는 위치를 더욱 집중하여 계산하는 어텐션 가중치(attention weight)를 학습하는 방법이다.

본 논문에서는 인코딩 단계에서 인코더(Encoder RNN)를 여러 층으로 쌓아 더 높은 수준의 추상화를 시도하고, 디코딩 단계에서 멀티 태스크 학습 기반 포인터 네트워크를 이용하여 각 어절의 중심어를 찾고 의존 구문 분석의 레이블(label) 정보를 출력하는 멀티 레이어 포인터 네트워크(Multi-layer Pointer Networks)를 이용한 한국어 의존 구문 분석을 제안한다.

2. 관련 연구

최근 딥 러닝을 이용한 전이 기반과 그래프 기반 의존 구문 분석이 활발히 연구되고 있다. 전이 기반 의존 구문 분석은 입력(버퍼)과 스택으로부터 구문 분석 상태 표현을 얻고, 신경망을 이용하여 다음 전이 액션을 결정하는 방법이다[2-4]. 그래프 기반 방법은 입력 단어들에 대한 의존 관계의 점수(score)를 딥 러닝 모델로 계산하여 의존 구문 분석을 수행한다[5-9].

한국어에서 딥러닝을 이용한 의존 구문 분석은 [12]의 FFNN (Feed-forward Neural Network)를 이용한 의존 구문 분석을 시작으로, RNN을 이용하여 전이 기반 의존 구문 분석을 수행하는 [2-4]와 그래프 기반 의존 구문 분석을 수행하는 [5-9]가 있다. 그 외로 네트워크 출력 결과만으로 의존 관계를 파악할 수 있는 포인터 네트워크를 이용한 의존 구문 분석[13]이 있다. 포인터 네트워크를 이용한 방법은 멀티 태스크 학습 기반으로 중심어의 위치와 레이블 정보를 동시에 학습할 수 있다.

본 논문에서 제안하는 멀티 레이어 포인터 네트워크를 이용한 한국어 의존 구문 분석은 기존 포인터 네트워크를 이용한 의존 구문 분석 모델의 인코더 부분을 여러 층으로 쌓아 인코딩을 수행하여 인코딩 레벨에서 각 단어들 간의 관계표현을 더욱 추상화하여 학습 및 예측을 수행하는 확장된 방법이다.

3. 멀티 레이어 포인터 네트워크

포인터 네트워크는 RNN encoder-decoder에서 확장된 모델로서 어텐션 메커니즘을 기반으로 한다. 인코더는 RNN을 이용하여 입력 열에 대한 hidden state를 인코딩하여 인코딩 벡터로 만든다. 디코더(Decoder RNN)는 현

재까지 생성된 디코더의 hidden state와 인코딩 벡터를 어텐션 함수로 계산하여 입력열에 대한 얼라인먼트 점수(alignment score)를 구하고, 가장 높은 점수를 가지는 위치를 결과로 출력한다. 멀티 레이어 포인터 네트워크는 인코더 단계에서 레이어를 여러 층 쌓아 구성한 포인터 네트워크의 확장된 모델이다. 인코더에서 인코딩을 수행할 때 레이어를 여러 층 쌓으면 각 hidden state의 표현들이 더욱 추상화된다.

본 논문에서는 인코더로 bidirectional Gated Recurrent Unit(BiGRU)[14]을 사용하며, 각 히든 레이어의 BiGRU 수식은 아래와 같다.

$$e_s = W_e x_s$$
$$h_s^1 = BiGRU(h_{s-1}^1, e_s)$$
$$h_s^2 = BiGRU(h_{s-1}^2, h_s^1)$$
$$h_s^3 = BiGRU(h_{s-1}^3, h_s^2)$$

x_s는 입력열의 s 번째 단어이고, e_s는 x_s에 단어표현(word embedding)을 적용한 것이다. h_s^1은 BiGRU로 인코딩한 결과로써 GRU의 forward pass와 backward pass의 hidden state를 연결(concatenate)한 것이며, 다음 히든 레이어 h_s^2의 입력으로 주어진다. h_s^2는 h_s^1을 입력으로 받아 BiGRU로 인코딩을 수행한 결과이며, h_s^3도 h_s^2를 입력으로 받아 BiGRU로 인코딩을 수행한 결과이다. 이와 같이 레이어를 여러 층 쌓게 되면 주어진 입력열의 표현을 더욱 추상화된다.

디코더에서는 입력열의 특정 위치($yi_t \in Y_{input}$)를 입력받아 yi_t의 중심어 위치($y_t \in Y_{output}$)와 의존 구문 레이블($z_t \in Z_{output}$)을 예측하며 다음과 같이 정의된다.

$$h_t = GRU(h_{yi_t}^l, h_{t-1}),\ l \in \{1,2,3\}$$
$$a_t^s = \frac{\exp(score_a(h_t, h_s^l))}{\sum_{s'} \exp(score_a(h_t, h_{s'}^l))}$$
$$y_t = argmax_s(a_t^s)$$

디코더에서는 forward GRU를 사용하며, 디코더의 각 스텝의 hidden state를 h_t로 나타낸다. h_t는 인코더의 hidden state $h_{yi_t}^l$과 디코더의 이전 hidden state를 입력으로 받아 GRU를 거쳐 계산한다. 인코더의 hidden state h_s^l에서 인덱스 l은 멀티 레이어의 수를 의미한다. a_t^s는 yi_t의 중심어 위치 확률로 $score_a$함수의 결과 벡터를 정규화한 값(attention weight)이다. a_t^s에서 가장 높은 확률을 갖는 위치가 yi_t의 중심어 위치인 출력 결과 y_t가 된다.

$$c_t^D = h_{y_t}^l$$
$$score_a(h_t, h_s^l) = \begin{cases} v_t^T \tanh(W_a[h_t; h_s^l]), & concat \\ v_t^T \tanh(W_a[h_t; h_{yi_t}^l; h_s^l]), & concat2 \\ v_t^T \tanh(W_a[h_t; c_{t-1}^D; h_s^l]), & concat3 \\ v_t^T \tanh(W_a[h_t; h_{yi_t}^l; c_{t-1}^D; h_s^l]), & concat4 \end{cases}$$

$score_a$함수는 얼라인먼트 점수를 계산하며, $concat$, $concat2$와 $concat3$, $concat4$로 나뉜다. 먼저 $concat$은 h_t와 h_s^l을 연결한(concat) 후 가중치 행렬을 곱하여 점수를 계산하며, $concat2$는 h_t와 $h_{yi_t}^l$, h_s^l을 이용하여 점수를 계산한다. $concat3$는 h_t와 c_{t-1}^D, h_s^l을 이용하여 점수를 계산하고, $concat4$는 h_t와 $h_{yi_t}^l$, c_{t-1}^D, h_s^l 모두를 이용하여 점수를 계산한다. 여기서 c_t^D는 a_t에서 가장 높은 얼라인먼트 점수를 가지는 y_t를 이용(hard attention)하여 구한 문맥 벡터이다.

$$score_z(h_t, c_t^D) = \begin{cases} u_t^T ReLU(W_z[c_t^D; h_t]), & concat_z \\ u_t^T ReLU(W_z[c_t^D; h_t; h_{yi_t}^l]), & concat2_z \end{cases}$$
$$z_t = argmax(softmax(score_z(h_t, h_{y_t}^l))$$

z_t는 의존 관계 레이블의 출력 결과로, h_t와 c_t^D를 입력으로 한 $score_z$함수를 이용하여 구한다. $score_z$함수는 $concat_z$와 $concat_z2$ 방법으로 나뉜다. $concat_z$는 $score_a$의 $concat$을 기반으로 하며, c_t^D와 h_t를 이용하여 점수를 계산한다. $concat_z2$ 방법은 $score_a$의 $concat2$와 $concat3$, $concat4$를 기반으로 하며, c_t^D와 h_t, $h_{yi_t}^l$을 이용하여 점수를 계산한다.

4. 멀티 레이어 포인터 네트워크를 이용한 한국어 의존 구문 분석

본 논문에서 제안한 멀티 레이어 포인터 네트워크를 이용한 의존 구문 분석은 멀티 태스크 학습 방법을 이용하며, 디코더의 입력으로 주어진 임의의 어절(형태소 단위로 나눠 입력)에 대하여 의존 관계를 갖는 중심어의 위치를 학습하고, 이에 해당하는 레이블 정보도 함께 학습한다. 멀티 레이어 포인터 네트워크는 인코더를 여러 층 쌓아 일반 포인터 네트워크보다 더 높은 추상화를 시도하는 모델이다. 이에 따른 멀티 레이어 포인터 네트워크의 모델 구조는 [그림 1]과 같다.

[그림 1]은 인코더와 디코더로 구성되며, 인코더는 입력열(Input layer)과 임베딩 레이어(Projection layer), l개($l \in \{1,2,3\}$)의 히든 레이어 (Hidden layer)로 구성된다. 입력열은 $X = \{A, B, C, D, </s>\}$와 같으며, 각 알파벳은 입력되는 형태소를 의미하고, </s>는 문장의 끝을 알리는 종료 기호이다. 이때 인코더의 입력인 형태소들은 문장의 어절들을 형태소 단위로 바꿔 사용하고, 어절 정보를 표현하기 위하여 어절 사이에 띄어쓰기 심볼(<sp>)을 추가한다. 인코더의 입력열 X는 임베딩 레이어에서 단어 표현(word embedding)이 적용되어 히든 레이어로 보내진다. 첫 번째 히든 레이어에서 BiGRU를 수행하여 인코딩 벡터 h^1을 만들고, 그 다음 두 번째 히든 레이어에서 앞서 생성한 h^1을 입력으로 인코딩을 수행한다. 두 번째 히든 레이어의 인코딩 결과 h^2는 세 번째 히든 레이어의 입력으로 주어 인코딩을 수행하여 h^3를 만들며, 각 레이어 마다 드랍아웃(dropout)을 적용하여 과적합

(over fitting)을 방지한다.

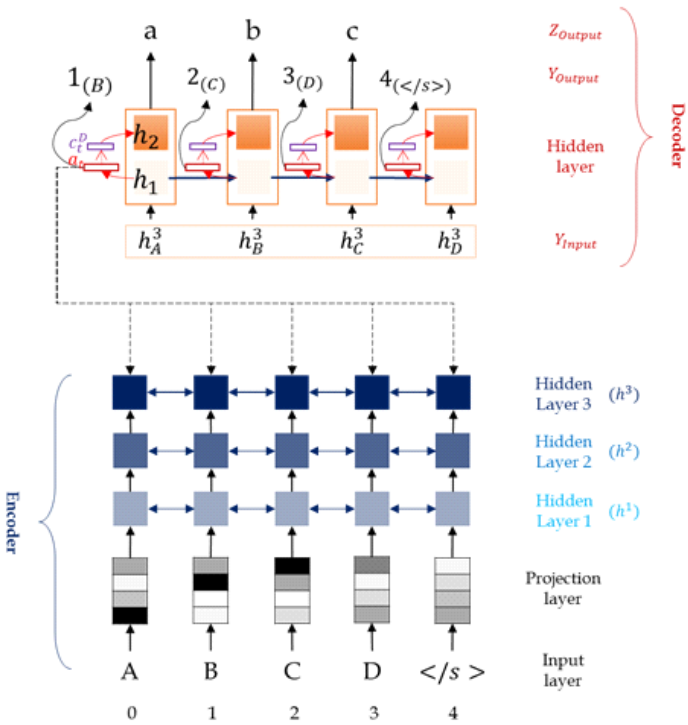


그림 1. 멀티 레이어 포인터 네트워크를 이용한 의존 구문 분석 모델

디코더는 입력열(Y_{input})과 히든 레이어, 출력 결과(Y_{output}, Z_{output})로 구성되며, Y_{input}={$0_{(A)}$, $1_{(B)}$, $2_{(C)}$, $3_{(D)}$}와 같이 주어진 입력열에 대응되는 출력열들(의존 관계를 나타내는 Y_{output}={$1_{(B)}$,$2_{(C)}$,$3_{(D)}$, $4_{(</s>)}$}, 의존 관계 레이블 정보를 출력하는 Z_{output}={a, b, c, d})을 생성한다. 디코더의 입력 Y_{input}은 각 어절의 위치 정보를 형태소 레벨의 인덱스로 표현한 것이며, 출력 Y_{output}은 각 어절의 중심어 위치를 형태소 레벨의 인덱스로 표현한 것이다. 이와 같이 입력과 출력의 단위는 형태소이기 때문에 어절의 대표 형태소 위치를 정의하여 학습데이터를 만들어 사용하며, [13]에서 가장 좋은 성능을 보인 *Dp0*를 입력기준으로 한다.

5. 실험

본 논문에서 제안한 멀티 레이어 포인터 네트워크를 이용한 한국어 의존 구문 분석의 실험 데이터는 의존 구조로 변환된 세종 데이터셋[13]을 사용하였으며, 데이터셋은 총 59,659 문장이다. 학습에 사용한 문장은 전체 문장 중 90%인 53,842 문장이고, 나머지 10%인 5,817 문장을 평가에 사용하였다. 입력 형태소에 대한 단어표현은 10만 단어에 대한 2년치 뉴스기사를 Neural Network Language Model (NNLM)으로 학습한 것을 사용하였다. 의존 구문 분석 결과에 대한 평가 척도는 Unlabeled Attachment Score (UAS), Labeled Attachment Score (LAS)를 사용하였다.

인코더와 디코더, 어텐션 레이어의 활성함수는 모두 tanh를 사용하였고, 의존 관계 레이블 Z_{output}의 활성함수는 relu를 사용하였다. 임베딩 레이어에서 적용되는 단어 표현은 형태소 단위 100차원을 사용하였고, 히든 레이어의 차원 수와 드랍아웃은 [14]에서 가장 좋은 성능을 보인 차원 수 600과, 드랍아웃 0.2로 설정하였으다. 학습은 모멘텀(momentum)을 이용하였고, 학습율(learning rate) 0.1을 시작으로 성능 개선이 없으면 3 에포크(epoch)마다 50%씩 감소시켰다.

표 1. 한국어 의존 구문 분석 성능 비교 (자동 분석 형태소 결과 이용)

의존 구문 분석	UAS	LAS
임수종[15]: 세종코퍼스	88.15	-
이창기[12] with MI	90.37	88.17
박천음[13]: Pointer Networks	91.79	89.48
나승훈[9]: Deep biaffine	91.78	89.76
2-layer stack concat	**92.16**	**89.88**
2-layer stack concat2	92.11	89.82
2-layer stack concat3	92.12	89.85
2-layer stack concat4	92.08	89.78
3-layer stack concat	92.11	89.70
3-layer stack concat2	92.13	89.79
3-layer stack concat3	91.98	89.63

[표 1]은 본 논문에서 제안한 멀티 레이어 포인터 네트워크를 이용한 한국어 의존 구문 분석 성능을 기존의 연구들과 비교한 표이다. 사용한 데이터는 모두 세종코퍼스로 같으며 자동 형태소분석 결과를 사용하였다. 실험 결과, 본 논문에서 제안한 멀티 레이어 포인터 네트워크를 이용하여 의존 구문 분석을 수행하는 방법이 일반 포인터 네트워크를 사용한 박천음[14]과 Deep biaffine 모델을 사용한 나승훈[9] 연구에 비하여 전반적으로 높은 성능을 보였으며, 2-layer stack cancat 모델이 UAS 92.16%, LAS 89.88%로 박천음[14] 연구에 비하여 UAS 0.37%, LAS 0.4%, 나승훈[9] 연구에 비하여 UAS 0.38%, LAS 0.12%의 성능 향상을 보였다. 그러나 3-layer stack을 하였을 때 오히려 성능이 떨어지는 경향을 보였다. 이는 stack을 쌓는 것이 항상 좋은 성능을 보이지 않으며 과적합 되기 쉬움을 보여준다.

[그림 2, 3]은 가장 좋은 성능을 보인 2-layer stack cancat 모델에 대한 의존 관계 포인팅 얼라인먼트 스코어와 의존 레이블 얼라인먼트 스코어에 대한 결과를 보인다. 인코더의 입력은 "덕분/NNG[0] 에/JKB[1] <SP>[2] 나/NP[3] 는/JX[4] <SP>[5] 수석/NNG[6] 으로/JKB[7] <SP>[8] 졸업/NNG[9] 하/XSV[10] 는/ETM[11] <SP>[12] 기쁨/NNG[13] 을/JKO[14] <SP>[15] 가지/VV[16] 었/EP[17] 다/EF[18] ./SF[19] </S>[20]"과 같다. 디코더는 입력 기준 *Dp0*를 따르며, 디코더의 입력은 "덕분/NNG, 나/NP, 수석/NNG, 졸업/NNG, 기쁨/NNG, 가지/VV"와 같고, 디코더의 정답은 y_{output}=[16, 16, 9, 13, 16, 20], z_{output}=[DP, NP_OBJ, DP, NP, VP_MOD, NP_AJT]와 같다. 디코더의 출력은 [그림 2]와

같이 $\hat{y}_{output}$=[16, 16, 9, 13, 16, 20], [그림 3]과 같이 $\hat{z}_{output}$=[DP, NP_OBJ, DP, NP, VP_MOD, NP_AJT]의 결과를 보였다.

6. 결론

본 논문에서는 한국어 의존 구문 분석을 수행하기 위하여 인코딩 단계에서 더욱 높은 추상화를 시도하기 위한 멀티 레이어 포인터 네트워크 모델을 이용하였다. 실험 결과, 본 논문에서 제안한 방법이 2-layer stack과 concat 얼라인먼트 스코어 방법, 단어 표현 100차원, 히든 레이어 600 차원, 드랍아웃 0.2와 같은 하이퍼 파라미터일 때 UAS 92.16%, LAS 89.88%로 한국어 의존 구문 분석에 관한 선행연구들 보다 더욱 높은 성능을 보였다.

향후 연구로는 의존 구문 분석을 수행하는 포인터 네트워크에 음절과 어절 표현을 추가로 적용하여 인코더에서의 표현력을 더욱 높인 네트워크 구조를 모델링 할 예정이다.

감사의 글

본 연구는 과학기술정보통신부 및 정보통신기술진흥센터의 정보통신·방송 연구개발 사업의 일환으로 하였음. [2013-0-00131, (엑소브레인-1세부) 휴먼 지식증강 서비스를 위한 지능진화형 WiseQA 플랫폼 기술 개발]

참고문헌

[1] D. Hays. Dependency theory: a formalism and some observations. Language, pp. 511-525, 1964.

[2] M. Ballesteros, C. Dyer, N. A. Smith. Improved Transition-Based Parsing by Modeling Characters instead of Words with LSTMs, *EMNLP 2015*, 2015.

[3] L. Kong, C. Alberti, D. Andor, I. Bogatyy, D. Weiss. DRAGNN: A Transition-based Framework for Dynamically Connected Neural Networks, https://arxiv.org/abs/1703.04474

[4] 나승훈, 김강길, 김영길. Stack LSTM을 이용한 전이 기반 한국어 의존 파싱, *KCC 2016*, 2016.

[5] 나승훈, 이건일, 신종훈, 김강일. 순환 컨트롤러를 이용한 Stack LSTM기반 한국어 의존 파싱, *KCC 2016*, 2016.

[6] E. Kiperwasser and Y. Goldberg. Simple and accurate dependency parsing using bidirectional LSTM feature representations. *TACL*, 2016.

[7] T. Dozat, C. D. Manning. Deep Biaffine Attention for Neural Dependency Parsing, *ICLR 2017*, 2017.

[8] X. Ma and E. Hovy. Neural Probabilistic Model for Non-projective MST Parsing, https://arxiv.org/abs/1701.00874

[9] 나승훈, 이건일, 신종훈, 김강일. Deep Biaffine Attention을 이용한 한국어 의존 파싱, *KCC 2017*, 2017.

[10] O. Vinyals, M. Fortunato and N. Jaitly. Pointer Networks. *Advances in Neural Information Processing Systems*, pp. 2674-2682, 2015.

[11] D. Bahdanau, et al. Neural machine translation by jointly learning to align and translate. *Proc. of ICLR' 15*, arXiv:1409.0473, 2015.

[12] 이창기, 김준석, 김정희. 딥 러닝을 이용한 한국어 의존 구문 분석. 제26회 한글 및 한국어 정보처리 학술대회, pp. 87-91, 2014.

[13] 박천음, 이창기. 포인터 네트워크를 이용한 한국어 의존 구문 분석. *정보과학회논문지 44.8*, pp. 822-831, 2017.

[14] K. Cho, et al. Learning phrase representation using RNN encoder-decoder for statistical machine translation. *Proc. of EMNLP' 14*, 2014.

[15] 임수종, 김영태, 나동열. 자질 가중치의 기계학습에 기반한 한국어 의존파싱. 정보과학회논문지, 소프트웨어 및 응용 제38권 제4호, 2011.

의존관계 어텐션	덕분/NNG	나/NP	수석/NNG	졸업/NNG	기쁨/NNG	가지/VV
덕분/NNG [0]						
에/JKB [1]						
<SP> [2]						
나/NP [3]						
는/JX [4]						
<SP> [5]						
수석/NNG [6]						
으로/JKB [7]						
<SP> [8]						
졸업/NNG [9]			0.9966			
하/XSV [10]						
는/ETM [11]						
<SP> [12]						
기쁨/NNG [13]				0.9997		
을/JKO [14]						
<SP> [15]						
가지/VV [16]	1.0000	1.0000			1.0000	
었/EP [17]						
다/EF [18]						
./SF [19]						
</S> [20]						1.0000

그림 2. 한국어 의존 구문 분석 결과의 포인팅 얼라인먼트 점수

레이블 어텐션	덕분/NNG	나/NP	수석/NNG	졸업/NNG	기쁨/NNG	가지/VV
NP_CNJ						
NP_SBJ						
VP						
NP_OBJ		0.9994				
NP_AJT						0.9999
VP_MOD					1.0000	
DP	0.9998		0.9999			
NP				0.9999		
AP						
VNP						
NP_MOD						
VP_SBJ						
VP_OBJ						
VNP_MOD						
VP_CMP						
NP_CMP						
VNP_CMP						
VNP_OBJ						
VNP_AJT						
AP_AJT						
VP_AJT						
IP						
AP_MOD						
NP_INT						
VNP_SBJ						
L						
R						
X_CMP						
VP_CNJ						
VNP_CNJ						
X_CMP						
X_MOD						
X_CNJ						
X_AJT						
X_SBJ						
X_OBJ						
<S>						
</S>						

그림 3. 한국어 의존 구문 분석 결과의 레이블 분류에 대한 얼라인먼트 점수

딥러닝을 이용한 전이 기반 한국어 품사 태깅 & 의존 파싱 통합 모델

민진우 ○†, 나승훈†, 신종훈††

전북대학교†, ETRI††
Jinwoomin4488@gmail.com, nash@jbnu.ac.kr, jhsin82@etri.re.kr

A Transition based Joint Model for Korean POS Tagging & Dependency Parsing using Deep Learning

Jin-Woo Min○†, Seung-Hoon Na†, Jong-Hoon Sin††
Chonbuk National University†, ETRI††

요 약

형태소 분석과 의존 파싱은 자연어 처리 분야에서 핵심적인 역할을 수행하고 있다. 이러한 핵심적인 역할을 수행하는 형태소 분석과 의존 파싱에 대해 일괄적으로 학습하는 통합 모델에 대한 필요성이 대두 되었고 이에 대한 많은 연구들이 수행되었다. 기존의 형태소 분석 & 의존 파싱 통합 모델은 먼저 형태소 분석 및 품사 태깅에 대한 학습을 수행한 후 이어서 의존 파싱 모델을 학습하는 파이프라인 방식으로 진행되었다. 이러한 방식의 학습을 두 번 연이어 진행하기 때문에 시간이 오래 걸리고 또한 형태소 분석과 파싱이 서로 영향을 주지 못하는 단점이 존재하였다. 본 논문에서는 의존 파싱에서 형태소 분석에 대한 전이 액션을 포함하도록 전이 액션을 확장하여 한국어 형태소 분석 & 의존파싱에 대한 통합모델을 제안하였고 성능 측정 결과 세종 형태소 분석 데이터 셋에서 F1 97.63%, SPMRL '14 한국어 의존 파싱 데이터 셋에서 UAS 90.48%, LAS 88.87%의 성능을 보여주어 기존의 의존 파싱 성능을 더욱 향상시켰다.

주제어: 형태소 분석, 품사 태깅, 의존 파싱, LSTM, 통합모델

1. 서론

형태소 분석과 의존 파싱은 다양한 자연언어처리 분야에서 핵심적인 역할을 수행하고 있다. 한국어 형태소 분석은 CRF, SVM 등 기존의 기계학습 방법이 주를 이루었다[1-3]. 최근 다양한 딥러닝 모델들[4,5]을 이용한 형태소 분석 연구들이 진행되고 있는데 딥러닝 방식은 별도의 자질 추출 단계를 필요로 하지 않고 최소의 자질로부터 복잡한 자질까지 스스로 학습하는 방식이다.

형태소 분석은 음절 단위 형태소 분석 방식과 단어 단위 형태소 분석 방식으로 나뉠 수 있으며 음절 단위 형태소 분석은 입력 문장을 음절 단위로 하여 형태소 시작과 해당 형태소로 이어짐을 나타내는 [B,I]태그가 포함된 품사 태그를 부착한다. 형태소 기반 방식은 분할 된 형태소에 직접 바로 태그를 부여하는 방식이다[6].

또한, 의존 파싱 분야에서도 딥러닝을 이용한 연구들이 많이 이루어지고 있다. 딥러닝을 이용한 의존 파싱 연구는 크게 전이 기반 방식[7-11,14]과 그래프 기반 방식[12,13]으로 나뉘어 연구되고 있다. 전이 기반 방식은 입력에 대한 버퍼와 스택으로부터 자질 벡터들을 얻은 후 딥러닝 신경망을 통해 전이 액션을 결정하는 방식으로 해당 딥러닝 모델은 초기에는 FNN[7]에 기반을 두었으나 최근 들어 LSTM과 같은 RNN 계열의 모델들[8-11]이 주를 이루고 있다. 그래프 기반 방식은 전역적 탐색 방식으로 지배소와 의존소에 대한 점수를 뉴럴 신경망을 통해 계산한다.

본 논문에서는 전이 기반 의존 파싱 방식에서 형태소 분석 액션도 처리 할 수 있도록 전이 액션을 확장하여 형태소 분석 & 의존 파싱 통합 모델을 제안하고 세종 형태소 분석 데이터 셋과 SPMRL' 14 의존 파싱 데이터 셋에 적용하여 각각 형태소 분석 성능 F1 97.80%, 의존 파싱 성능 UAS 90.48%, LAS 88.87%를 보여주어 기존의 의존 파싱 성능을 더욱 향상시켰다.

2. 관련 연구

한국어 품사 태깅에 대한 다양한 연구가 진행되었다. 음절 기반 한국어 형태소 분석은 주로 순차 태깅 기반으로 연구가 진행되었는데 [1,3]에서는 각각 기계학습 모델인 CRF, Structural SVM을 적용하였고 [4]에서는 품사태깅, 개체명 인식 등 순차 태깅 문제에서 최고의 성능을 보이고 있는 딥러닝 모델인 Bi-LSTM CRF를 적용하였다. Sequence-to-Sequence 모델은 임의 길이의 한 종류의 시퀀스를 다른 한 종류의 시퀀스로 변환하는 딥러닝 모델로 기계번역 분야에서 탁월한 성능을 보여주고 있다. [5]에서는 입력문장을 해당 형태소와 품사 태그로 번역하는 모델로

표 1. 전이 액션 별 스택 및 버퍼 정보의 갱신 과정

M_t	C_t	S_t	B_t	Action	M_{t+1}	C_{t+1}	S_{t+1}	B_{t+1}
M	c, C	S	B	$Split(t)$	$(t, c), M$	C	S	B
M	c, C	S	B	$Merge$	M	C	S	B
W_{st}, M	<SP>, C	S	$(\_, u), B$	$Word$	M	C	S	$(\mathrm{wt}(W_{st}), u), B$
M	C	S	$(\mathrm{wt}(W_{st}), u), \mathrm{B}$	$Shift$	M	C	$(\mathrm{wt}(W_{st}), \mathrm{u}), S$	B
M	C	$\mathrm{u}, \mathrm{v}, S$	B	$LeftArc(l)$	M	C	$g_l(v, u), S$	B
M	C	$\mathrm{u}, \mathrm{v}, S$	B	$RightArc(l)$	M	C	$g_l(u, v), S$	B

$[W_{st} = \{(t_1, c_1), \cdots, (t_k, c_k)\}$, k: 해당 어절 내 형태소의 개수, c:형태소, t:품사태그]

보고 Sequence-to-Sequence 모델을 한국어 형태소 분석 및 품사 태깅 문제에 적용하는 연구가 진행되었다.

[6]에서는 중국어 형태소 분할 문제를 전이 기반 방식으로 적용하여 현재 음절을 현재 형태소에 부착하는 액션, 현재 형태소를 결정짓고 해당 음절을 새로운 형태소의 시작음절로 분할하는 2가지 액션으로 적용하여 기존의 성능을 향상시켰다. 또한 [18]에서는 전이 기반으로 중국어 단어 분할 및 의존 파싱에의 통합 모델에 대한 연구를 진행하였는데 [6]에서의 단어 분할을 위한 전이 액션과 의존 파싱을 위한 전이 액션을 적절하게 조합하는 방식을 사용하였다.

딥러닝을 이용한 의존 파싱 연구에는 Biaffine Attention을 적용한 그래프 기반 방식이 있고 이를 이용하여 [12]에서는 영어 데이터셋에서 적용하여 현재 시스템에서 최고 성능을 보여주었다. [13]에서는 한국어와 같은 형태학적으로 복잡한 언어에 적용할 수 있도록 문자기반 Biaffine Attention 모델로 확장하여 한국어 의존 파싱 문제에서 기존 최고 성능을 개선시켰다.

전이 기반 신경망 모델은 다양한 자연어 처리 분야에 적용되어 왔고 의존 파싱 문제에서도 탁월한 성능을 보여주고 있다. 그중에서 Stack LSTM[9-11]은 스택과 버퍼 내의 정보를 Stack LSTM을 통해 인코딩 한 후 다음 전이 액션을 결정하는 방식이다. [10]에서는 한국어와 형태학적으로 복잡하여 같이 복잡하여 단어 표상을 직접적으로 얻어내기 힘든 언어에 대해서 음절 혹은 형태소를 LSTM혹은 CNN을 통해 합성하여 단어의 표상을 얻어내는 방법들이 연구되었다. 음절 태그의 표상과 형태소의 표상들을 LSTM을 통해 얻은 후 결합하여 단어 표상을 얻어내는 하이브리드 합성 방법을 제안하였다. [11]에서는 순환 컨트롤러를 사용하여 전이액션을 결정하는 방식으로 Stack LSTM을 확장한 연구도 있었다.

Sequence-to-Sequence 모델을 이용하여 구문 구조를 [15]에서와 같이 선형화하여 시퀀스로 변환한 후 모델의 출력으로 하여 입력 문장으로부터 출력을 생성하는 방식으로 학습한다. 또한, 한국어 의존 파싱에서 어텐션 메커니즘을 이용하여 입력열에 대한 위치를 출력하는 포인터 네크워크를 이용하여 입력 어절에 대한 중심어를 찾고 해당 전이 액션을 결정하는 연구도 있었다[16].

형태소 분석 및 의존 파싱에 대한 통합 모델에 대한 연구도 진행되었는데 [17]에서는 Stack Propagation 방식을 사용하여 형태소 분석에 대한 학습을 수행한 후 파싱에 대한 학습을 수행 할 때 형태소 분석 결과에 대한 품사의 Explicit 표상을 사용하는 것 아닌 형태소 분석 단계에서 딥러닝 모델을 거친 히든 상태인 Implicit 표상을 사용하는 방식이다. 위의 방식은 형태소 분석이 잘못되었을 때 발생하는 형태소 분석 오류를 최소화 할 수 있다.

본 논문에서는 형태소 분석을 처리 할 수 있도록 전이 액션을 추가하고 [14]의 전이 기반 의존 파싱 모델을 확장하여 한국어 형태소 분석 및 의존 파싱 통합모델을 정의하고 제안 모델에 대한 실험을 진행하였다.

3. 통합 모델

3.1. 전이액션의 확장

[14]에서의 $Shift$, $Left\_Arc$, $Right_Arc$의 기존 파싱 액션 세 가지에 액션은 $Split$, $Merge$, $word$의 세 가지 품사 태깅 액션을 추가하여 총 6가지의 액션으로 구분되며 파싱 액션을 위한 버퍼와 스택인 B, S이외에 형태소 분석을 위한 버퍼와 스택 C, M을 별도로 두며 전이 액션 별 스택 및 버퍼 정보의 갱신 과정은 다음 표 1로 설명한다. 표 1의 c, <SP>, t는 각각 음절, 공백 음절, 품사 태그를 의미하고 u, v는 단어 혹은 부분 트리의 ROOT 노드를 나타내고 해당 어절의 시작 형태소와 형태소의 품사로부터 마지막 k번째 형태소와 품사까지의 집합을 $W_{st} = \{(s_1, t_1), \cdots, (s_n, t_k)\}$로 정의하며 M에서 W_{st}로부터 어절의 시작 형태소와 끝 형태소의 품사를 결합인 어절 태그를 생성하는 함수는 $\mathrm{wt}(W_{st})$, 어절 태그가 부여되지 않은 버퍼의 TOP노드는 튜플 (_, u)로 표현한다.

먼저 형태소 분석 액션인 $Split$ Action은 현재 버퍼가 가리키고 있는 음절을 새로운 형태소를 시작으로 하여 해당 음절(형태소)를 스택 M에 PUSH하는 액션으로 새로운 형태소에 품사 t를 부여하는 역할도 수행한다. $Merge$ Action은 단순히 현재 스택이 가리키고 있는 형태소에 해당 음절을 추가하는 액션으로 실제로 하는 동작은 버퍼의 Focus를 다음 음절로 이동하는 역할만을 하게 된다. $Word$ Action은 공백 음절을 만나면 자동적으로 수행되며 어절의 경계를 의미하므로 M내의 해당 어절에 대한 모든 형태소들에 대한 품사가 결정되어지기 때문에 wt함수로 M에서 어절에 대한 어절 태그를 생성하는 액션이다. 또한, 현재 버퍼의 Focus를 공백 음절에서 다음 음절로 이동하는 역할도 수행한다.

다음으로 의존 파싱에 대한 전이 액션은 [8]의 Arc-standard 방식의 $Shift, Left_Arc, Right_Arc$로 구성된다. $Shift$ Action

표 2. 통합 전이 액션의 실행 예

Action	M	C	S	B
$init$	[]	[고,향,은,<SP>,서,울,이,다,<SP>]	[]	[고향은,서울이다,root]
$Split(NNG)$	[고]	[향,은,<SP>,서,울,이,다,<SP>]	[]	[고향은,서울이다,root]
$Merge$	[고]	[은,<SP>,서,울,이,다,<SP>]	[]	[고향은,서울이다,root]
$Split(JX)$	[고,은]	[<SP>,서,울,이,다,<SP>]	[]	[고향은,서울이다,root]
$Word$	[고,은]	[서,울,이,다,<SP>]	[]	[고향은,서울이다,root]
$Shift$	[고,은]	[서,울,이,다,<SP>]	[고향은]	[서울이다,root]
$Split(NNP)$	[고,은,서]	[울,이,다,<SP>]	[고향은]	[서울이다,root]
$Merge$	[고,은,서]	[이,다,<SP>]	[고향은]	[서울이다,root]
$Split(VCP)$	[고,은,서,이]	[다,<SP>]	[고향은]	[서울이다,root]
$Split(EF)$	[고,은,서,이,다]	[<SP>]	[고향은]	[서울이다,root]
$Word$	[고,은,서,이,다]	[]	[고향은]	[서울이다,root]
$Shift$	[고,은,서,이,다]	[]	[고향은,서울이다]	[root]
$Left\_Arc$(tpc)	[고,은,서,이,다]	[]	[서울이다]	[root]
$Shift$	[고,은,서,이,다]	[]	[서울이다, root]	[]
$Right\_Arc(root)$	[고,은,서,이,다]	[]	[root]	[]

는 현재 파서 버퍼가 가리키고 있는 단어를 스택에 추가하는 액션이다. $Left\_Arc\ Action$, $Right_Arc\ Action$은 스택 내의 두 Top 노드들 간에 의존성을 형성하는 액션으로 두 노드 사이의 지배소(head)와 의존소(dep)를 결정하여 하나로 병합된 파스 트리를 다시 스택에 추가하는 액션이다. 파스 트리를 생성하는 함수는 $g_l(v,u)$로 여기서 l은 의존 관계 레이블을 나타내며 괄호의 앞 요소(v)가 지배소를 나타내고 뒷 요소(u)는 의존소를 표현한다.

위의 표 2는 전이 액션의 실행 과정을 보여준다. 표 2에서 보듯이 입력은 공백을 포함한 한국어 원문이고 기본적으로 음절 단위로 전이 액션을 결정하게 된다. 만약 버퍼 C가 가리키고 있는 음절이 "공백"이 아닌 경우에는 파싱 액션이 아닌 품사 태깅 액션을 결정하게 되는 반면 "공백"인 경우에는 어절의 경계가 결정되기 때문에 $Word$액션이 자동으로 수행되어 어절태그가 생성된 후 해당 시점에 파싱 액션을 결정하도록 하여 형태소 분석 액션과 파싱 액션이 특정 상황에서만 발생하도록 제한하였다.

3.2. 버퍼와 스택의 TOP 노드 표상

두 버퍼 C,B에 대한 표상은 다음 두 입력 임베팅 벡터열 $\boldsymbol{x}=\{\boldsymbol{x}_1,\cdots,\boldsymbol{x}_n\}$, $\boldsymbol{z}=\{\boldsymbol{z}_1,\cdots,\boldsymbol{z}_n\}$로부터 각각 LSTM을 통해 은닉벡터로 얻어지게 되는데 형태소 분석을 위한 음절 임베딩 벡터 $\boldsymbol{x}_i$를 얻을 때 i번째 음절을 c_i라 하고 다음 표 3과 같은 자질 유형을 사용한다.

표 3. 입력으로 사용되는 자질 유형

음절 자질	Explanation
Unigram	c_{i-1}, c_i, c_{i+1}
Bigram	$c_i c_{i+1}$
Trigram	$c_i c_{i+1} c_{i+2}$

반면, 파싱을 위해 사용되는 음절 벡터열 $\boldsymbol{z}$는 별도의 자질 추출 없이 해당 음절의 임베딩으로 구성되며 임베딩을 얻는 과정은 수식 (1),(2)로 정리한다.

$$\boldsymbol{x}_i = lookup([uni(i); bi(i); tri(i)]) \quad (1)$$
$$\boldsymbol{z}_i = lookup(c_i) \quad (2)$$

위의 수식에서 uni, bi, tri함수는 각각 표 3에 대응되어 n-gram 자질을 추출하는 함수이고 $lookup$함수는 임베딩 lookup Table을 보고 해당 임베딩 벡터를 얻어내는 함수이다. 입력열 $\boldsymbol{x}$ 와 $\boldsymbol{z}$로부터 은닉열 $\boldsymbol{h}=\{\boldsymbol{h}_1,\cdots,\boldsymbol{h}_n\}$, $\boldsymbol{k}=\{\boldsymbol{k}_1,\cdots,\boldsymbol{k}_n\}$를 얻는 과정은 수식 (3)과(4)로 보여준다.

$$\{\boldsymbol{h}_1,\cdots,\boldsymbol{h}_n\} = LSTM(\{\boldsymbol{x}_1,\cdots,\boldsymbol{x}_n\}) \quad (3)$$
$$\{\boldsymbol{k}_1,\cdots,\boldsymbol{k}_n\} = LSTM(\{\boldsymbol{z}_1,\cdots,\boldsymbol{z}_n\}) \quad (4)$$

형태소 분석 버퍼와 스택 C, M과 파싱 버퍼와 스택 B, S의 해당 상태 벡터들을 각각 $\mathbf{r}(C)$, $\mathbf{r}(M)$, $\mathbf{r}(B)$, $\mathbf{r}(S)$라 정의한다. S_0, S_1을 각각 버퍼 혹은 스택의 TOP, 2번째 TOP 노드로 표현하고 해당 노드의 입력 음절에 대한 위치를 얻어내는 함수는 $cpos$이며 각 상태 백터는 다음 식 (5)~(8)을 통해 얻어진다[1].

$$\mathbf{r}(C) = \boldsymbol{h}_{cpos(C_0)} \quad (5)$$
$$\mathbf{r}(M) = \boldsymbol{h}_{cpos(M_0)} \quad (6)$$
$$\mathbf{r}(B) = [\boldsymbol{k}_{cpos(B_0)}; lookup(wt(B_0))] \quad (7)$$
$$\mathbf{r}(S) = [\boldsymbol{k}_{cpos(S_0)}; \boldsymbol{k}_{cpos(S_1)}; feats(S_0); feats(S_1)] \quad (8)$$

[1] 엄밀하게 정의하면 각 노드는 음절을 포함한 튜플의 형태로 존재하여 튜플의 음절을 얻어내는 함수인 char를 이용하여 $\boldsymbol{h}_{cpos(char(C_0))}$가 정확한 수식이지만 편의상 위의 형태를 사용한다. 식 (5)~(8) 동일.

식 (5),(6)에서 보듯이 형태소 버퍼와 스택의 상태 백터 $\mathbf{r}(C)$, $\mathbf{r}(M)$는 단순히 입력 열 $\boldsymbol{x}$로부터 LSTM을 통해 얻어진 은닉열 $\boldsymbol{h}$에서 TOP노드에 해당하는 위치의 은닉 벡터의 값을 취하게 된다. 식 (5)가 수행되는 과정을 표 2로 예를 들면 처음 $Merge$ Action 이 수행되는 3번 째 줄 버퍼 C 의 TOP노드는 "은" 을 나타내고 문장 내 음절 열에서 해당 음절은 세 번째에 위치하고 있으므로 LSTM을 통해 얻어진 은닉열 $\boldsymbol{h}$에서 세 번째 은닉 상태인 $\boldsymbol{h}_3$를 취하게 되는 것이다.

파서 버퍼와 스택의 상태 백터인 $\mathbf{r}(B)$, $\mathbf{r}(S)$로 넘어가서 식 (7)로 얻어지는 $\mathbf{r}(B)$는 TOP노드에 해당하는 은닉 열 뿐만 아니라 함수 $\mathrm{wt}(a)$를 통해 a노드의 시작 형태소와 끝 형태소의 품사인 어절태그를 생성한 후 $lookup$함수를 통해 변환된 임베딩 벡터를 결합하여 상태 벡터를 얻는다. 마지막으로 식 (8)으로 얻어지는 $\mathbf{r}(S)$는 $\mathbf{r}(B)$와 동일한 과정을 통해 얻어지는 스택 S의 TOP에 위치하는 두 파스 트리의 루트의 어절 S_0, S_1의 은닉 표상 뿐 아니라 다음 표 4와 같이 노드 S_0, S_1에 대한 각각 6개의 자질 $feats(S_0)$, $feats(S_1)$을 합한 총 12 개의 자질들을 포함한다.

표 4. 파서 스택의 파스트리에 대한 자질 벡터

$feats(S_i)$
$S_i.child_1.label$
$S_i.child_1.sibling_{-1}.label$
$S_i.child_{-1}.label$
$S_i.child_{-1}.sibling_1.label$
$S_i.child_2.$label
$S_i.child_{-2}.$label

여기서 $child$는 해당 노드의 자식 노 드로 $child_{-i}, child_j$는 각각 해당 노드의 왼쪽 i번째 자식 오른쪽 j번째 자식을 의미하며 $sibling$은 해당 노드의 형제 노드로 $child$와 동일한 표기법을 사용하여 label은 상위 노드와의 의존 관계 레이블을 의미한다.

3.3. 전이 기반 품사태깅 & 의존 파싱 통합모델

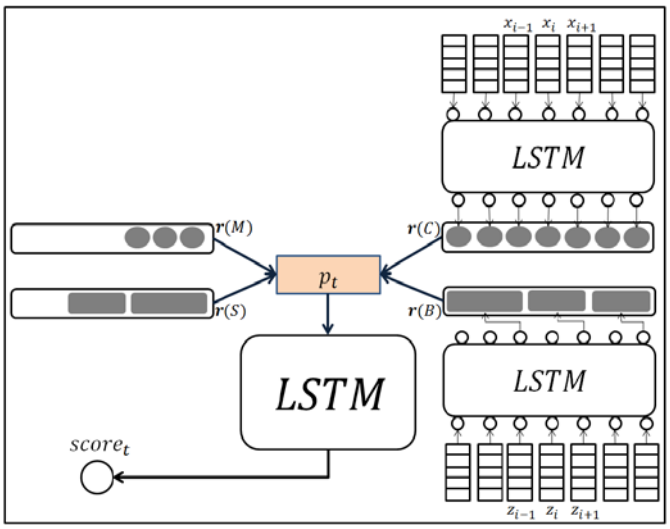


그림 1. 형태소 분석 & 의존 파싱 통합모델 구조

본 모델은 형태소 분석 및 품사 태깅에 대한 학습을 수행한 후 이어서 의존 파싱 모델을 학습하는 파이프 라인 방식과 달리 품사 태깅과 의존 파싱을 동시에 학습하는 Joint 방식이며 모델의 전체적인 도식도는 위의 그림 1와 같다. 그림 1에서 보듯이 첫번째 단계로 음절 단위로 LSTM을 통해 얻어진 은닉벡터들이 버퍼 C, B에 채워지게 된다. 그 후, 전이 액션은 버퍼와 스택의 상태를 결합하여 딥러닝 신경망을 통해 결정하게 되는 구조로 3.2절에서 정의한 $\mathbf{r}(C)$, $\mathbf{r}(M)$, $\mathbf{r}(B)$, $\mathbf{r}(S)$의 상태 벡터들을 각각 아래의 식 (9)와 같이 연결하여 파서 상태 표상을 얻는다.

$$\boldsymbol{p}_t = [\mathbf{r}(C); \mathbf{r}(M); \mathbf{r}(B); \mathbf{r}(S)] \quad (9)$$

식 (9)를 통해 얻어지게 되는 파서 상태 표상 집합을 $\boldsymbol{p} = \{\boldsymbol{p}_1, \cdots, \boldsymbol{p}_t\}$로 표현하며 식 (10)에서와 같이 LSTM 신경망의 입력으로 하여 LSTM을 거쳐 다음 전이 액션으로의 점수 $\boldsymbol{score}_t$를 계산한다. $LSTM_t$ 함수는 LSTM을 통해 얻어진 은닉열로부터 t번째 은닉 벡터를 취하도록 하는 함수이다.

$$\boldsymbol{score}_t = \boldsymbol{W} \cdot \left(LSTM_t(\{\boldsymbol{p}_1, \cdots, \boldsymbol{p}_t\})\right) + \boldsymbol{b} \quad (10)$$

얻어진 $\boldsymbol{score}_t$는 softmax층으로 연결되어 확률로 변환한 후 전이 확률 중에 최대가 액션을 다음 전이 액션으로 하여 버퍼와 스택의 다음 상태를 결정한다. 파이프라인 방식과 제안 모델의 차이점은 해당 Action을 결정하기 위해 형태소 분석과 의존 파싱의 버퍼와 스택의 상태를 모두 고려하게 된다. 다시 말해서 품사 태깅 액션을 결정할 때 파싱 결과를 이용하고 파싱 액션을 결정할 때 품사 태깅 결과를 이용하여 서로가 작용하게 되어 보다 나은 결과를 예측하게 하는 방식이다.

3.4. 기학습 파라미터의 사용

3.2 절에서 설명한 바와 같이 형태소 분석을 위한 음절 표상은 LSTM을 통해 얻어지게 된다. 본 논문에서 사용한 SPMRL'14 데이터 셋은 정답 형태소가 부착된 2만 7천여 문장으로 형태소 분석은 의존 파싱에 비해 보다 많은 학습 데이터를 필요로 한다. 파싱 데이터 셋은 일반적으로 형태소 분석 데이터에 비해 부족하기 때문에 형태소 분석에 대한 학습이 완전하게 이루어지지 않는 문제가 발생한다.

본 논문에 대한 연구를 수행하면서 전이 기반으로 형태소 분석 만을 하는 모델에 대한 연구도 수행하였는데 모델의 구조는 음절 열로부터 LSTM을 통해 은닉 상태를 얻어내는 과정과 버퍼와 스택으로부터 상태벡터를 얻고 상태벡터를 LSTM을 통해 전이 액션을 결정하는 두 단계로 구분하며 [1]과 동일한 약 25만여 문장의 세종 형태소 분석 데이터부터 학습을 진행하였다. 위의 형태소 분석 모델의 학습을 통해 얻어진 첫번째 단계의 LSTM의 모든 파라미터 집합을 $\mathbf{mW}$라 하고 본 모델에서의 형태소 분석 데이터 부족 문제를 해결하기 위해 파라미터 집합 $\mathbf{mW}$를 식 (1)의 LSTM의 파라미터의 초기 값으로 사용하여 데이터 부족 문제를 보완하였다.

4. 실험

4.1. 실험 셋팅

본 논문에서는 학습 집합으로 SPMRL'14 한국어 의존 파싱 데이터 셋의 약 2만 7천 문장을 이용하였으며 또한 형태소 분석 실험을 위한 평가 집합으로는 [1]과 동일한 세종 형태소 분석 데이터 셋의 약 25만 여 문장을 사용하였으며 25만 문장 중 5만 문장을 테스트 문장 집합으로 하여 평가를 실시하였다. 제안 모델의 학습률은 0.0005, 모든 히든 레이어의 Dropout 비율은 0.8로 설정하였다.

4.2. 실험 결과

표 5와 6에는 각각 형태소 분석 실험 결과와 의존파싱 실험 결과가 제시되어 있다.

표 5. 세종 형태소 분석 실험 결과

	F1(morph)
CRF * [2]	97.61%
CRF(BIES) *	97.75%
Bi-LSTM CRF *	96.96%
SVM [3]	98.03%
Seq2Seq [5]	97.15%
전이기반 *	97.77%
Segment&Tag&Parsing Joint Model *	97.63%

(*는 평가셋이 동일)

표 6. SPMRL'14 의존 파싱 실험 결과

	UAS	LAS
나승훈[9] Stack LSTM	89.10	87.34
나승훈[11] Stack LSTM + 컨트롤러	89.94	88.36
민진우[14] SynTaxNet	90.33	88.69
POSTag & Parsing Joint Model	90.48	88.87

표 5에서 보듯이 동일 데이터 셋에서 기존의 CRF의 성능보다 미약하게 높은 성능을 보여주고 있으나 BIES 표기법을 사용한 CRF나 전이기반 형태소 분석 모델에 비해서는 다소 떨어지는 결과를 보여준다. 3.4절에서 기학습 파라미터를 사용하여 보완하였으나 좀 더 효과적인 방안이 필요하다.

표 6에는 SPMRL'14 한국어 의존 파싱 데이터셋에 대한기존의 방식과 본 모델에 대한 실험 결과가 제시 되어 있다. 본 모델의 베이스 라인 모델인 [14]의 SynTaxNet은 F1 measure 97.61%의 CRF로 자동 형태소 분석결과를 이용한 모델로 Joint모델의 성능이 더 높은 것을 확인 할 수 있다.

5. 결론

본 논문은 전이 기반 의존 파싱 모델에 형태소 분석을 처리할 수 있도록 전이액션을 추가 확장하여 한국어 형태소 분석 및 의존 파싱 통합모델에 대한 실험을 진행하였고 의존 파싱에서 기존의 성능을 향상시켰다. 하지만 이 방식은 형태소 분석 데이터의 부족으로 별도의 데이터로부터 미리 형태소 분석에 대한 학습 파라미터를 End-to-End 방식이 아닌 방식으로 학습이 진행되었다. 차후 End-to-End 방식으로 학습할 수 있는 방안에 대해 모색할 예정이다.

또한, 형태소 분석에서는 전이 기반 형태소 분석 모델에 비해서는 다소 떨어지는 결과를 보여주고 있는데 이러한 문제점을 보완하는 연구에 대해 진행할 예정이다.

감사의 글

본 연구는 미래창조과학부 및 정보통신기술진흥센터의 정보통신·방송 연구개발사업의 일환으로 수행하였음. [R7119-16-1001, 지식증강형 실시간 동시통역 원천기술 개발]

참고문헌

[1] 나승훈, 양성일, 김창현, 권오욱, 김영길. "CRF 에 기반한 한국어 형태소 분할 및 품사 태깅." HCLT 2012.

[2] Seung-Hoon Na . Conditional Random Fields for Korean Morpheme Segmentation and POS Tagging. ACM Transactions on Asian and Low-Resource Language Information Processing , 14(3), 2015

[3] 2015이창기. "Structural SVM 을 이용한 한국어 띄어쓰기 및 품사 태깅 결합 모델." 정보과학회논문지: 소프트웨어 및 응용 40.12 (2013): 826-832.

[4] 김혜민, 윤정민, 안재현, 배경만, 고영중. 품사 분포와 Bidirectional LSTM-CRFs를 이용한 음절 단위 형태소 분석기, HCL 2016

[5] 이건일, 이의현, 이종혁. "Sequence-to-sequence 모델을 이용한 한국어 형태소 분석 및 품사 태깅." 한국정보과학회 학술발표논문집 (2016)

[6] Zhang, Meishan, Yue Zhang, and Guohong Fu. "Transition-Based Neural Word Segmentation." ACL (1). 2016.

[7] 이창기, 김준석, 김정희. "딥 러닝을 이용한 한국어 의존 구문 분석." 제 26 회 한글 및 한국어 정보처리 학술대회 (2014): 87-91.

[8] 이건일, 이종혁. "순환 신경망을 이용한 전이 기반 한국어 의존 구문 분석." 정보과학회 컴퓨팅의 실제 논문지 21.8 (2015): 567-571.

[9] 나승훈, 김강일, 김영길, "Stack LSTM을 이용한 전이 기반 한국어 의존 파싱," KCC 2016

[10] 나승훈, 신종훈, 김강일, "Stack LSTM 기반 한국어 의존 파싱을 위한 음절과 형태소의 결합 단어 표상 방법", HCLT 2016

[11] 나승훈, 이건일, 신종훈, 김강일, "순환 컨트롤러를 이용한 Stack LSTM기반 한국어 의존 파싱," KCC 2016

[12] T. Dozat, C. D. Manning, “Deep Biaffine Attention for Neural Dependency Parsing,” ICLR 2017
[13] 나승훈, 이건일, 신종훈, 김강일, "Deep Biaffine Attention을 이용한 한국어 의존 파싱", 한국정보과학회 학술발표논문집, 2017.6
[14] 민진우, 나승훈, "전이기반 순환유닛을 이용한 SyntaxNet 기반 한국어 의존 파싱", 한국정보과학회 학술발표논문집, 2017.6
[15] 황현선, 이창기, Sequence-to-Sequence 모델을 이용한 한국어 구구조 구문 분석, HCLT 2016
[16] 박천음, 이창기, “멀티 태스크 학습 기반 포인터 네트워크를 이용한 한국어 의존 구문 분석,” KCC 2016
[17] Zhang, Yuan, and David Weiss. "Stack-propagation: Improved representation learning for syntax." arXiv preprint arXiv:1603.06598 (2016).
[18] Kurita, Kawahara, Kurohashi, Neural Joint Model for Transition-based Chinese Syntactic Analysis, 2017

Multi-task sequence-to-sequence learning을 이용한 한국어 형태소 분석과 구구조 구문 분석

황현선○, 이창기

강원대학교

{hhs4322, leeck}@kangwon.ac.kr

Korean morphological analysis and phrase structure parsing using multi-task sequence-to-sequence learning

Hyunsun Hwang ○, Changki Lee

Kangwon National University

요 약

한국어 형태소 분석 및 구구조 구문 분석은 한국어 자연어처리에서 난이도가 높은 작업들로서 최근에는 해당 문제들을 출력열 생성 문제로 바꾸어 sequence-to-sequence 모델을 이용한 end-to-end 방식의 접근법들이 연구되었다. 한국어 형태소 분석 및 구구조 구문 분석을 출력열 생성 문제로 바꿀 시 해당 출력 결과는 하나의 열로서 합쳐질 수가 있다. 본 논문에서는 sequence-to-sequence 모델을 이용하여 한국어 형태소 분석 및 구구조 구문 분석을 동시에 처리하는 모델을 제안한다. 실험 결과 한국어 형태소 분석과 구구조 구문 분석을 동시에 처리할 시 형태소 분석이 구구조 구문 분석에 영향을 주는 것을 확인 하였으며, 구구조 구문 분석 또한 형태소 분석에 영향을 주어 서로 영향을 줄 수 있음을 확인하였다.

주제어: 형태소 분석, 구구조 구문 분석, multi-task learning, sequence-to-sequence learning

1. 서론

형태소 분석은 한국어 자연어처리 중 하나로 형태소 분리, 품사 태깅, 원형 복원 등의 여러 단계를 거쳐 난이도가 높은 작업에 속한다. 구문 분석은 문장의 구조를 분석하는 방법으로 구구조 구문 분석과 의존 구문 분석이 사용된다. 그러나 한국어 특성상 구구조 구문 분석의 난이도가 높고 시간 복잡도가 $O(n^3)$으로 높아 한국어 자연어처리에서는 주로 의존 구문 분석이 사용되었다.

최근 기계학습 알고리즘 중 하나인 딥 러닝(Deep Learning)을 자연어처리에 적용하는 연구가 많이 진행되었다[1,2]. 그 중 sequence-to-sequence 모델은 입력열을 길이가 다른 출력열로 변환하는 모델로, end-to-end 방식의 신경망 구조를 사용한다. 이러한 방식의 sequence-to-sequence 모델은 복잡한 문제를 출력열 생성 문제로 바꾸어 기존의 복잡한 작업을 단순화 시키는 장점이 있다. Sequence-to-sequence 모델은 Neural Machine Translation(NMT) 모델에 처음 적용이 되어 기계번역 문제를 end-to-end 방식의 모델로 처리하였다[3,4]. 이후 다른 자연어처리 문제들에 적용이 되었는데, 특히 복잡한 한국어 형태소 분석과 구구조 구문 분석을 출력열 생성 문제로 바꾸어 end-to-end 방식의 접근을 시도한 연구들이 진행 되었다[5,6,7].

한국어 형태소 분석 및 구구조 구문 분석을 출력열 생성 문제로 바꾸었을 때, 이 두가지 문제는 하나의 열로서 표현이 가능하며 sequence-to-sequence 모델을 사용하여 동시에 분석이 가능하다. 본 논문에서는 sequence-to-sequence 모델을 이용하여 한국어 형태소 분석과 구구조 구문 분석을 동시에 처리하는 모델을 소개하며 이러한 모델의 단점을 설명하고 이를 극복하는 새로운 sequence-to-sequence 모델을 제안한다.

2. 관련 연구

한국어 형태소 분석은 형태소 분리, 품사 태깅, 원형 복원 등의 여러 단계를 거치며 특히 품사 태깅의 경우 CRFs나 Structural SVM등의 기계학습 알고리즘을 주로 사용하였다[8]. [5]에서는 이러한 여러 단계로 이루어진 한국어 형태소 분석을 sequence-to-sequence 모델을 이용하여 end-to-end 방식의 한국어 형태소 분석을 제안하였다. [6]에서는 sequence-to-sequence 모델에 입력열의 단어를 출력열에 복사하는 copying mechanism을 적용하여 학습데이터에서 상태적으로 적게 등장하는 고유 명사와 같은 단어들에 내성을 지닌 end-to-end 방식의 한국어 형태소 분석을 제안 하였다.

한국어 구구조 구문 분석은 주로 확률을 이용한 방법들이 연구되었으며[9], 최근에는 문장을 트리 구조 형태로 분석하는 구구조 구문 분석 결과를 구문 분석 태그가 포함된 괄호를 이용하여 하나의 열로서 표현하여 sequence-to-sequence 모델을 이용한 한국어 구구조 구문 분석을 시도한 연구가 진행되었다[7]. [7]에서는 sequence-to-sequence 모델의 성능을 높이기 위한 attention mechanism[10]과 input-feeding[11]의 기술을 적용하여 높은 한국어 구구조 구문 분석 성능을 보였다.

3. Multi-task sequence-to-sequence learning을 이용한 한국어 형태소 분석과 구구조 구문 분석

한국어 형태소 분석과 구구조 구문 분석은 하나의 출력열로서 표현될 수 있다. 본 논문에서는 sequence-to-sequence 모델을 이용하여 한국어 형태소 분석과 구구조 구문 분석을 동시에 처리하는 multi-task sequence-to-sequence 모델을 제안한다.

3.1 출력 결과를 합친 sequence-to-sequence learning

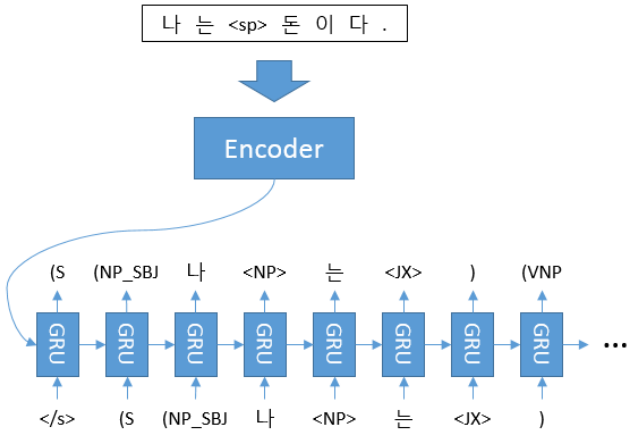


그림 1. 출력 결과를 합친 sequence-to-sequence 모델을 이용한 한국어 형태소 분석 및 구구조 구문 분석의 예시

[7]에서는 형태소 분석된 결과를 이용한 구구조 구문 분석을 시도하여 출력 결과의 대상 어절을 'XX'로 치환 하였다. 이는 이미 형태소 분석이 되어있다는 가정하에 구구조 구문 분석을 시도하기 때문이며, 이를 다시 형태소 분석 결과로 치환하면 sequence-to-sequence 모델을 이용하여 한국어 형태소 분석 및 구구조 구문 분석을 동시에 시도할 수 있다. 그림 1은 출력 결과를 합친 sequence-to-sequence 모델을 이용한 한국어 형태소 분석 및 구구조 구문 분석 예시 이다. Sequence-to-sequence 모델의 인코더 입력은 형태소 분석이 되지 않은 문장을 음절 단위로 넣게 되며 이때 어절을 구분하는 태그인 '<sp>'(띄어쓰기)태그를 같이 넣게 된다. 출력은 구문 분석 태그를 포함하여 형태소 분석된 결과를 [6]과 동일하게 음절 단위로 출력을 하게 된다. 이때 [6]과 마찬가지로 입력열의 단어가 출력열에도 등장 하게 됨으로 copying mechanism을 적용할 수 있다.

그러나 한국어 특성상 한국어 자연어처리는 형태소 분석이 상당히 중요하며, 이러한 모델은 구구조 구문 분석 시 형태소 분석 결과를 이용할 수 없다는 문제가 발생하게 된다.

3.2 Hidden state를 공유하는 Multi-task sequence-to-sequence learning

3.1절에서 설명되었듯이 한국어 구구조 구문 분석은 한국어 형태소 분석에 영향을 받기 때문에 출력 결과를 단순히 합친 sequence-to-sequence 모델로는 낮은 구구조 구문 분석 성능을 보이게 된다. 이에 따라 본 논문에서는 구구조 구문 분석이 형태소 분석 결과를 이용할 수 있도록 hidden state를 공유하는 multi-task sequence-to-sequence 모델을 제안한다.

기존의 multi-task learning은 하나의 hidden state에서 서로 다른 task의 결과를 출력하는 모델로서 하나의 신경망으로 서로 다른 문제를 동시에 해결 할 수 있다는 장점이 있다[12]. 그러나 한국어 형태소 분석과 구구조 구문 분석의 출력은 서로 다른 출력열의 형태로 길이가 다를 수 있다. 본 논문에서는 서로 다른 디코더를 이어 hidden state를 공유하는 multi-task sequence-to-sequence 모델을 설계하였다.

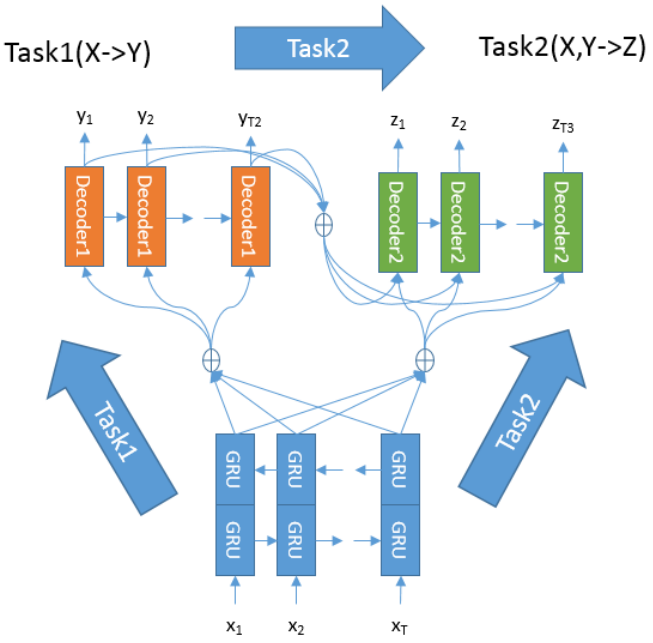


그림 2. Hidden state를 공유하는 multi-task sequence-to-sequence 모델

그림 2는 서로 다른 두 task의 디코더를 이어 hidden state를 공유하는 multi-task sequence-to-sequence 모델의 그림이다. $X(x_1,x_2,\cdots,x_T)$는 입력열이며, $Y(y_1,y_2,\cdots,y_{T2})$는 task1(형태소 분석)의 출력열이고, $Z(z_1,z_2,\cdots,z_{T3})$은 task2(구구조 구문 분석)의 출력열이다. Task1의 경우 입력열 X를 인코딩하여 출력열 Y를 디코딩하게된다. Task2의 경우 입력열 X를 인코딩한 hidden state들을 이용하여 출력열 Z를 디코딩 하나, 이때 task1의 디코더 hidden state들을 추가 정보로 보게 된다.

본 논문에서는 그림 2의 task1을 한국어 형태소 분석으로 설계하여 [6]과 동일한 attention mechanism, input-feeding, copying mechanism을 적용한 sequence-

to-sequence 모델로 설계 하였다. Task2는 한국어 구구조 구문 분석으로 설계하여 [7]과 동일한 attention mechanism, input-feeding을 적용한 sequence-to-sequence 모델을 사용하였으며, 추가적으로 task1의 정보를 사용하기 위해 다음과 같이 디코더를 재설계하였다.

$$ {}^{1}e_i^t = f_{ATT1}(E_{tgt}(z_{t-1}), h1_{t-1}, h2_{t-1}, {}^{1}h_i) $$

$$ {}^{1}a_i^t = \frac{{}^{1}e_i^t}{\sum_{ii=1}^{T} \exp({}^{1}e_{ii}^t)} $$

$$ {}^{1}c^t = \sum_{i=1}^{T} {}^{1}a_i^t \, {}^{1}h_i $$

$$ {}^{2}e_j^t = f_{ATT2}(E_{tgt}(z_{t-1}), h1_{t-1}, h2_{t-1}, {}^{2}h2_j) $$

$$ {}^{2}a_j^t = \frac{{}^{2}e_j^t}{\sum_{jj=1}^{T2} \exp({}^{2}e_{jj}^t)} $$

$$ {}^{2}c^t = \sum_{j=1}^{T2} {}^{2}a_j^t \, {}^{2}h2_j $$

$$ \mathrm{z} = \sigma(W_z E_{tgt}(z_{t-1}) + U1_z h1_{t-1} + U2_z h2_{t-1} + W_{zc1} \, {}^{1}c^t + W_{zc2} \, {}^{2}c^t + b_z) $$

$$ \mathrm{r} = \sigma(W_r E_{tgt}(z_{t-1}) + U1_r h1_{t-1} + U2_r h2_{t-1} + W_{rc1} \, {}^{1}c^t + W_{rc2} \, {}^{2}c^t + b_r) $$

$$ \mathrm{m} = \mathrm{f}(W_m E_{tgt}(z_{t-1}) + U2_m h2_{t-1} + U1_m(h1_{t-1} \odot \mathrm{r}) + W_{mc1} \, {}^{1}c^t + W_{mc2} \, {}^{2}c^t + b_m) $$

$$ h1_t = (1 - z) \odot h1_{t-1} + z \odot \mathrm{m} $$

$$ h2_t = f_2(W_{h2} h1_t + b_{h2}) $$

$$ z_t = argmax(softmax(W_{zh} h1_t + W_{zh2} h2_t + W_{zz} E_{tgt}(z_{t-1}) + W_{zc1} \, {}^{1}c^t + W_{zc2} \, {}^{2}c^t + b_z)) $$

$E_{tgt}(z_{t-1})$은 task2의 이전 시간의 디코딩 결과로 생성된 단어 z_{t-1}의 출력 언어 word embedding이며, $h1_{t-1}$과 $h2_{t-1}$은 task2 디코더의 이전 hidden state이다. f_{ATT1}는 task2의 디코딩 시간 t에서 입력열 X의 인코더 hidden state vector(${}^{1}h_i$)에 대한 attention weight를 결정하기 위한 신경망이다. 생성된 attention weight는 입력열 X의 인코더 hidden state vector들을 이용하여 task2의 디코딩 시간 t에서 입력열 X에 대한 context vector ${}^{1}c^t$를 생성한다. 마찬가지로 task1에 대한 정보를 task2에 적용 시키기 위해 task1의 디코더 hidden state vector(${}^{2}h2_j$)를 또 다른 신경망인 f_{ATT2}를 이용하여 task2의 디코딩 시간 t에서 task1의 디코더 hidden state vector(${}^{2}h2_j$)에 대한 attention weight를 결정하고 마찬가지로 context vector ${}^{2}c^t$를 생성한다. 이후 [7]과 동일한 변형된 GRU 디코더를 사용하며 위에서 입력열 X에 대한 context vector ${}^{1}c^t$외 추가적으로 task1에 대한 context vector ${}^{2}c^t$를 또 다른 가중치를 두어 추가정보로 넣었다.

4. 실험 및 결과

본 논문에서 제안한 multi-task sequence-to-sequence 모델의 성능을 평가하기 위해 [7]과 동일한 세종말뭉치의 구구조 구문 분석 데이터를 사용하였다. 모든 source, target word embedding은 200차원을 사용하였고 형태소 분석(task1)의 디코더 히든레이어의 크기는 1000, 구구조 구문 분석(task2)의 디코더 히든레이어의 크기는 500으로 설계 하였다. 추가적으로 해당 데이터에 대한 형태소 분석만의 성능과 형태소 분석이 되지 않은 문장을 입력으로 할 때의 구구조 구문 분석의 성능도 측정하였다.

표 1. 한국어 형태소 분석(Task1) 및 구구조 구문 분석 (Task2) 성능 평가 결과

모델	Task1 F1	Task2 F1
RNN-search + input-feeding + copying[6]	92.48 (baseline)	-
RNN-search + input-feeding[7](raw corpus)	-	81.78 (baseline)
RNN-search + input-feeding[7](정답 형태소 분석 이용)	-	89.03(+7.25)
Model 1	94.10(+1.62)	78.56(-3.22)
Model 2	**94.78(+2.30)**	**85.61(+3.83)**

표 1은 각각의 sequence-to-sequence 모델 별 한국어 형태소 분석과 구구조 구문 분석의 성능 평가 결과 이다. Task1은 형태소 분석을 나타내며, task2는 구구조 구문 분석을 나타낸다. 먼저 [6]에서 제안된 RNN-search + input-feeding + copying 모델로 형태소 분석만을 시도할 시 F1 92.48의 성능을 보여 해당 데이터가 [6]의 데이터보다 크기가 작고, 형태소 분석이 어려운 데이터임을 알 수 있다([6]의 학습데이터는 9만 문장, 본 논문에서 사용한 학습데이터는 3만9천 문장). 마찬가지로 [7]에서 제안된 RNN-search + input-feeding 모델로 형태소 분석이 되지 않은 문장을 입력으로 받았을 시 F1 81.78의 낮은 구구조 구문 분석 성능을 보여 정답 형태소 분석을 사용한 [7]의 F1 89.03의 성능과 비교해 한국어 구구조 구문 분석에서 형태소 분석이 중요함을 알 수 있다. 표 1에서 model 1은 3.1절에서 제안한 단순히 출력 결과를 합친 sequence-to-sequence 모델이며 model 2는 3.2절에서 제안한 hidden state를 공유하는 multi-task sequence-to-sequence 모델이다. 실험 결과 출력 결과를 합친 model 1의 경우 형태소 분석의 성능이 F1 94.10이었으나 구구조 구문 분석 성능은 F1 78.56으로 형태소 분석이 되지 않은 문장을 입력으로 받았을 시의 F1 81.78의 성능보다 낮게 나왔다. 그러나 구구조 구문 분석 시 형태소 분석(task1)의 정보를 보게 설계한 model 2의 경우 구구조 구문 분석의 성능이 F1 85.61으로 형태소 분석이 되지 않은 문장을 입력으로 받았을 때보다 높은 성능을 보였다. Model 1에서의 구구조 구문 분석은 형태소 분석 정보를 사용하지 못함과 동시에 sequence-to-sequence 모델의 디코더 출력이 구구조 구문 분석 태그뿐만 아니라 형태소 분석 결과도 출력을 해야 하기 때문에 구문 분석의 난이도가 올라 것으로 분석된다. Model 2에서의 구구조 구문 분석은 형태소 분석이 되지 않은 문장을 입력으로 받았을 시의 구구조 구문 분석의 성능보다 높게 나와 task1의 형태소 분석 정보를 효과적

으로 사용하였음을 확인할 수 있다. 또한 model 1과 model 2 모두 [6]의 모델을 이용하여 순수하게 형태소 분석만을 시도한 경우보다 높은 성능을 보여 한국어 구구조 구문 분석에 한국어 형태소 분석이 영향이 끼치는 것은 물론 한국어 형태소 분석에 한국어 구구조 구문 분석이 영향을 미칠 수 있음을 보여준다. 그림 3은 hidden state를 공유하는 multi-task sequence-to-sequence 모델의 구구조 구문 분석 attention weight 예시이다. 구구조 구문 분석 시 입력열에 대한 정보뿐만 아니라 형태소 분석의 디코더 hidden state에 대한 정보를 [7]과 유사하게 어절의 구구조 구문 분석 태그 생성시 해당하는 어절 정보를 보려고 하는 것을 볼 수 있다.

감사의 글

이 논문은 2016년도 정부(미래창조과학부)의 재원으로 한국연구재단의 지원을 받아 수행된 기초연구사업임 (No. NRF-2016R1C1B1014124)

5. 결론

본 논문에서는 한국어 형태소 분석 및 구구조 구문 분석을 동시에 시도하는 multi-task sequence-to-sequence 모델을 제안하였다. 실험 결과 한국어 구구조 구문 분석에 형태소 분석이 영향을 미치게 설계하여 성능이 향상되는 것을 확인하였으며, 한국어 형태소 분석에 구구조 구문 분석 또한 영향을 미칠 수 있음을 확인하였다. 향후 연구로 단일 task 데이터 및 대규모 raw corpus 활용 등을 이용한 성능 향상 방안을 모색할 예정이다.

참고문헌

[1] Collobert, Ronan, et al. "Natural language processing (almost) from scratch." Journal of Machine Learning Research, 12, 2011.

[2] 이창기, 김준석, 김정희, "딥 러닝을 이용한 한국어 의존 구문 분석", 제26회 한글 및 한국어 정보처리 학술대회, pp. 87-91, 2014.

[3] Sutskever, Ilya, Oriol Vinyals, and Quoc V. Le. "Sequence to sequence learning with neural networks." Advances in neural information processing systems, 2014.

[4] Cho, Kyunghyun, et al. "Learning phrase representations using RNN encoder-decoder for statistical machine translation." EMNLP 2014

[5] 이건일, 이의현, 이종혁. "Sequence-to-sequence 기반 한국어 형태소 분석 및 품사 태깅." 정보과학회논문지 44.1 (2017): 57-62.

[6] 황현선, 이창기. "Copying mechanism 을 이용한 Sequence-to-Sequence 모델기반 한국어 형태소 분석." 한국정보과학회 학술발표논문집 (2016): 443-445.

[7] 황현선, 이창기, Sequence-to-Sequence 모델을 이용한 한국어 구구조 구문 분석, HCLT 2016

[8] 이창기, “Structural SVM을 이용한 한국어 띄어쓰기 및 품사 태깅 결합 모델”, 정보과학회논문지 : 소프트웨어 및 응용, 40(12), pp826-832, 2013.

[9] 이공주, 김재훈, 김길창. "한국어 구구조 문법을 기반으로 하는 확률적 구문 분석." 한국정보과학회 1996 년도 가을 학술발표논문집, 제23권, 제2호(A), pp557-560, 1996.

[10] Bahdanau, Dzmitry, Kyunghyun Cho, and Yoshua Bengio. "Neural machine translation by jointly learning to align and translate." International Confernece on Learing Represnetations, 2015.

[11] Luong, Minh-Thang, Hieu Pham, and Christopher D. Manning. "Effective approaches to attention-based neural machine translation." EMNLP 2015

[12] 박천음, 이창기. "포인터 네트워크를 이용한 한국어 의존 구문 분석." 정보과학회논문지 44.8 (2017): 822-831.

그림 3. Hidden state를 공유하는 multi-task sequence-to-sequence 모델의 구구조 구문 분석 attention weight 예시

● 구두발표 6: 대화/질의응답 2

도메인 특정 지식을 결합한 End-to-End Learning 방식의 한국어 식당 예약 대화 시스템 모델 개발

이동엽, 김경민, 임희석

고려대학교 컴퓨터학과
judelee93@korea.ac.kr, totoro4007@gmail.com, limhseok@korea.ac.kr

Development of a Dialogue System Model for Korean Restaurant Reservation with End-to-End Learning Method Combining Domain Specific Knowledge

Dong-Yub Lee, Gyeong-Min Kim, Heui-Seok Lim
Dept. of Computer Science and Engineering, Korea University

요 약

목적 지향적 대화 시스템(Goal-oriented dialogue system) 은 텍스트나 음성을 통해 특정한 목적을 수행할 수 있는 시스템이다. 최근 RNN(recurrent neural networks)을 기반으로 대화 데이터를 end-to-end learning 방식으로 학습하여 대화 시스템을 구축하는데에 활용한 연구가 있다. End-to-end 방식의 학습은 도메인에 대한 지식 없이 학습 데이터 자체만으로 대화 시스템 구축을 위한 학습이 가능하다는 장점이 있지만 도메인 지식을 학습하기 위해서는 많은 양의 데이터가 필요하다는 단점이 존재한다. 이에 본 논문에서는 도메인 특정 지식을 결합하여 end-to-end learning 방식의 학습이 가능한 Hybrid Code Network 구조를 기반으로 한국어로 구성된 식당 예약에 관련한 대화 데이터셋을 이용하여 식당 예약을 목적으로하는 대화 시스템을 구축하는 방법을 제안한다. 실험 결과 본 시스템은 응답 별 정확도 95%와 대화 별 정확도 63%의 성능을 나타냈다.

주제어: 대화 시스템, 딥러닝, 도메인 지식

1. 서론

목적 지향적 대화 시스템(Goal-oriented dialogue system) 은 텍스트나 음성을 통해 특정한 목적을 수행할 수 있는 시스템이다. 목적 지향적 대화 시스템을 구현하기 위한 이전 연구로는 slot-filling 의 방식과[1,2,3] language understanding, action selection과 같이 대화를 이해하고 그에대한 응답을 선택하는 모듈들을 이용하는 연구가 있다. 하지만 이러한 이전 연구들은 도메인에 대해 많은 지식을 포함하는 hand-craft 된 자질(feature)을 필요로 하고 이로인해 새로운 도매인에 대한 확장이 어렵다는 단점이 있다.

최근 RNN(recurrent neural networks)을 기반으로 대화 데이터를 end-to-end learning 방식으로 학습하여 대화 시스템을 구축하는데에 활용한 연구가 있다[4,5]. End-to-end 방식의 학습은 도메인에 대한 지식 없이 학습 데이터 자체만으로 대화 시스템 구축을 위한 학습이 가능하다는 장점이 있다. 하지만 end-to-end learning 방식을 이용하여 도메인 지식을 학습하기 위해서는 많은 양의 데이터가 필요하다는 단점이 존재한다. 식당 예약 시스템의 경우, 식당의 위치나 가격과 같은 특정 도메인에대한 지식 정보를 학습하기 위해서는 해당 정보들을 표현하는 많은 양의 학습 대화 데이터셋이 필요하다.

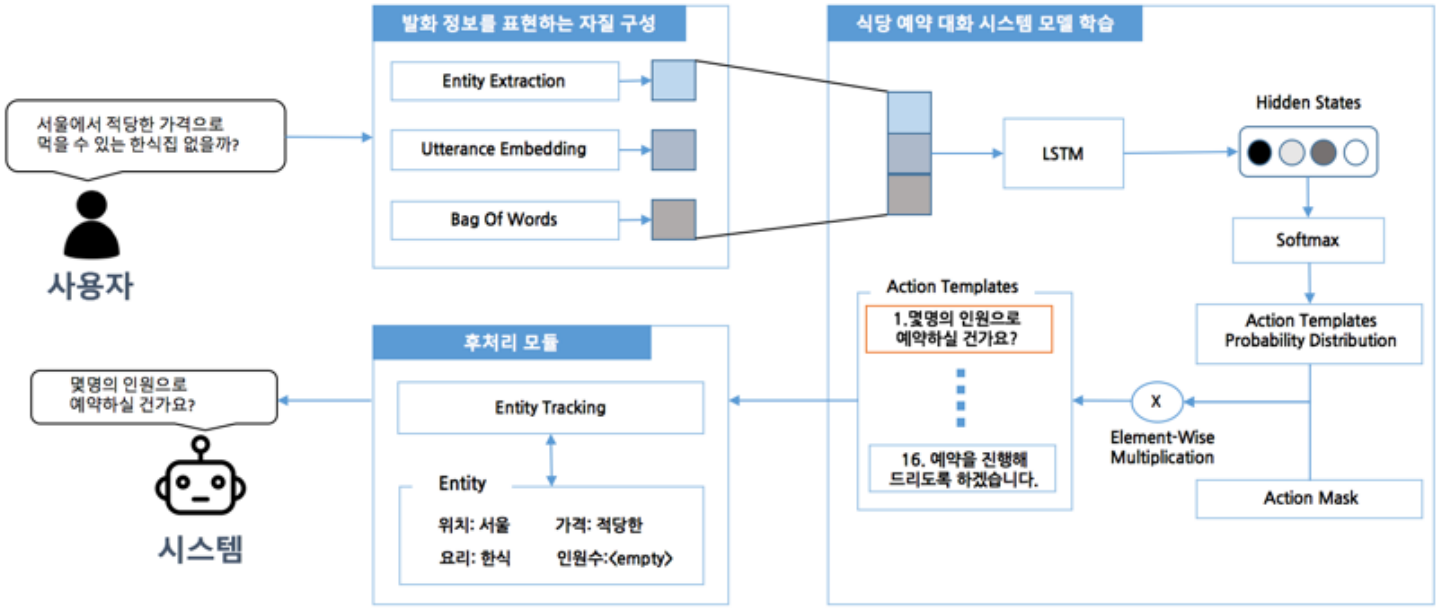
많은 양의 학습 대화 데이터셋이 필요하다는 일반적인 end-to-end learning 기반의 단점을 극복하기 위해 도메인 특정 지식을 결합하며 end-to-end learning 방식의 학습을 가능하게 한 Hybrid Code Network 구조를 제안한 연구가 있다[6]. Hybrid Code Network를 이용하여 구축한 대화 시스템은 도메인에 해당하는 특정 지식을 액션 템플릿(action template)의 정의를 통해 표현함으로써 보다 적은양의 학습 데이터 양으로 도메인 지식을 표현할 수 있다.

목적 지향적 대화 시스템의 학습을 위해 페이스북은 식당 예약에 관련한 학습 데이터셋(The 6 bAbI tasks)을 공개하였다[7]. 영어로 구축된 식당 예약에 관련한 학습 대화 데이터셋을 이용하여 한국어로 이루어진 식당 예약 관련 대화 데이터셋을 구축한 연구가 있다.[8, 9]

본 논문에서는 Hybrid Code Network를 기반으로 한국어로 구성된 식당 예약에 관련한 대화 데이터셋을 이용하여 식당 예약을 목적으로하는 대화 시스템을 구축하는 방법을 제안한다.

2. 제안하는 과정

[그림 1]은 본 논문에서 제안하는 도메인 특정 지식을



[그림 1] 도메인 특정 지식을 결합한 한국어 식당 예약 대화 시스템 모델의 전체 구조도

결합하고 end-to-end learning 방식의 학습이 가능한 한국어 식당 예약 대화 시스템 모델의 전체 구조도를 나타낸다. 본 논문에서는 한국어 식당 예약 대화 시스템 모델의 학습을 위해 학습 대화 데이터 759개와 테스트 대화 데이터 190개를 이용하였다. 사용자의 발화(utterance)로부터 식당 예약 대화 시스템 모델 학습에 필요한 자질(feature)을 구성하기 위해 개체 추출(Entity Extraction), 발화 임베딩(Utterance Embedding), 단어 주머니(Bag of Words) 와 같은 3가지 방법으로 발화 정보를 표현하는 자질을 구성한다. 각각의 방법으로 표현된 발화 정보를 표현하는 자질들은 연결(concatenation)되어 발화 정보에 대한 최종 자질을 형성한다. 형성된 발화 정보를 표현하는 최종 자질은 LSTM(long short term memory)의 입력으로 사용되고 LSTM은 은닉 상태(hidden states)를 계산하여 출력한다. 출력된 은닉 상태는 softmax의 입력으로 사용된다. 본 실험에서는 사용자에게 몇명의 인원으로 식당 예약을 할 것인지, 어떤 요리를 원하는지와 같은 발화를 식당 예약에 관련된 도메인 지식을 표현하기 위한 액션 템플릿(Action Template)로 정의하는데, softmax는 주어진 사용자의 발화에 대한 액션 템플릿들의 확률 분포(probability distribution)를 계산한다. 액션 템플릿들에 대한 확률 분포 값은 사용자의 발화에 대해 어떠한 액션 목록들이 필요한지 표시하는 액션 마스크(Action Mask) 값들과 성분끼리의 곱(element-wise multiplication)을 통해 최종적으로 시스템의 발화를 선택하게 된다. 후처리 모듈에서는 사용자의 발화에서 식당의 위치에 해당하는 서울과 같은 개체(entity)들을 추적(tracking)하여 최종적으로 선택된 시스템의 발화에서 추적중인 개체명이 필요할 시 추적중인 개체를 사용할 수 있도록 한다.

2.1 발화 정보를 표현하는 자질 구성

2.1.1 개체 추출(Entity Extraction)

본 논문에서는 식당의 위치, 음식의 가격, 음식의 종류, 인원 수 와 같이 식당 예약에 필요한 속성이 될 수 있는 개체들을 정의한다. 개체 추출을 진행하기 위해 각 속성들에 해당하는 개체 사전을 정의하고 발화로부터 문자열 매칭 알고리즘을 이용하여 발화 속에 존재하는 개체들을 추출할 수 있다.

2.1.2 발화 임베딩(Utterance Embedding)

발화 임베딩 모듈은 사용자의 발화로부터 의미론(semantics)적 특성을 반영하기 위해 word2vec 모델을 이용하여 사용자의 발화를 임베딩한 후 이를 자질로 구성한다. [식 1]은 발화를 구성하고 있는 전체 단어의 개수가 N, 문서의 i번째 단어를 워드 임베딩 공간에서 벡터 값으로 표현한 것을 v(i)라 할 때, 발화를 구성하고 있는 각 단어들의 벡터 평균을 나타낸다.

$$\frac{1}{N}\sum_{i=1}^{N} v(\mathrm{i}) \qquad (1)$$

2.1.3 발화 주머니(Bag of Words)

사용자의 발화를 발화 주머니로 구성하기 위해 학습 대화 데이터셋에 있는 단어 들의 집합을 기반으로 사전을 형성하였다. 이후 단어 사전을 기반으로 각 발화를 구성하는 단어들의 등장 여부를 표시하여 사용자의 발화를 표현하는 발화 주머니 자질을 구성하였다.

2.2 LSTM을 이용한 대화 시스템 모델 학습

시스템의 응답에 도메인 특정 지식을 반영하기 위해 [그림 2]와 같이 액션 템플릿을 정의한다. 본 논문에서 제안하는 식당 예약 대화 시스템 모델의 경우 총 16개의 액션 템플릿으로 구성되어 있다. 2.1장에서 구성한 발화 정보를 표현하는 자질들은 연결되어 발화 정보에 대한

최종 자질을 형성하고 이를 입력으로 이용하여 LSTM 은닉 상태를 계산하게 된다. 은닉 상태를 입력으로

Action Templates

0. api_call <음식종류> <위치> <인원수> <가격>
1. 가격의 범위는 어느정도로 생각하세요
2. 감사합니다
3. 네 또 변경하실게 있나요
4. 다른 리스트를 보여드릴게요
5. 또 도와드릴게 있나요
6. 몇명의 인원으로 예약하실 건가요
7. 안녕하세요 어떻게 도와드릴까요
8. 알겠습니다
9. 어떤 종류의 요리를 좋아하나요
10. 예약을 진행해드리도록 하겠습니다
11. 위치는 <info_address> 입니다
12. 위치는 어디에 있어야 하나요
13. 이 리스트는 어떤가요: <restaurant>
14. 전화번호는 <info_phone> 입니다
15. 좋아요 몇 가지 리스트를 보여드릴게요

[그림 2] 도메인 특정 지식을 반영하기 위한 시스템 응답 액션 템플릿 정의

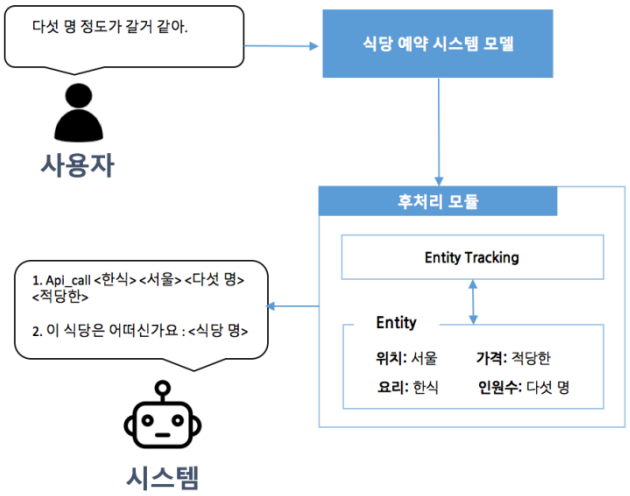
softmax는 각 액션 템플릿들에 대한 확률 분포 값을 계산한다. 이후 각 확률 분포 값들은 액션 마스크와 성분끼리의 곱을 통해 정규화(normalization)되고 이 값들과 실제 학습 데이터셋에서 레이블링 된 정답 시스템 응답과의 cross-entropy 값을 최소화 하도록 모델의 학습이 진행된다.

2.3 후처리 모듈을 통한 개체명 추적

후처리 모듈에서는 [그림 3]과 같이 사용자 발화에 대한 식당 예약 대화 시스템 모델의 예측 결과 선택된 액션 템플릿의 발화에서 개체명이 필요할 시, 해당 개체명을 같이 응답할 수 있도록 이전 발화로부터 추출된 개체들을 추적한다.

2.4 학습 모델 하이퍼 파라미터 설정

[표 1]은 한국어 식당 예약 대화 시스템 학습에 이용된 하이퍼 파라미터에 대한 정보를 나타낸다.



[그림 3] 이전 발화에서 추출한 개체들을 이용한 시스템 응답

[표 1] 학습에 이용된 모델의 하이퍼 파라미터 값

	Hyper-parameter	Value
word2vec	window size	5
	dimension	300
Bag of words	vocab size	1051
LSTM	initial state	0.0
	state size	128
	forget bias	1.0
	training epoch	20
	action template size	16
	initial learning rate	0.1
	decay rate	0.0

3. 실험 결과

한국어 식당 예약 대화 시스템 모델의 성능을 평가하기 위해 영어로 구성된 식당 예약 대화 데이터셋을 이용하여 실험한 식당 예약 대화 시스템 모델의 연구[6,7]와 같이 응답 별 정확도(per-response accuracy)와 대화 별 정확도(per-dialog accuracy) 을 측정하였다. 응답 별 정확도와 대화 별 정확도는 [식 2], [식 3]와 같이 나타낼 수 있다.

$$\text{per} - \textit{response}\ \text{accuracy} = \frac{\textit{시스템이 올바르게 응답한 발화의 개수}}{\textit{모든 대화를 구성하는 발화의 개수}} \quad (2)$$

$$\text{per} - \text{dialogue accuracy} = \frac{\text{시스템이 모두 올바르게 응답한 대화의 개수}}{\textit{모든 대화의 개수}} \quad (3)$$

[표 2]는 본 실험에서 진행한 한국어 식당 예약 대화 시스템의 응답 별 정확도와 대화 별 정확도를 나타낸다.

[표 2] 한국어 식당 예약 시스템의 응답 별 정확도와 대화 별 정확도

Methods	Accuracy
Per-response Accuracy	0.95
Per-dialogue Accuracy	0.64

[그림 4]는 학습된 모델을 이용하여 식당 예약을 진행하는 대화 시스템 모델의 실행 결과를 나타낸다. 사용자의 첫 발화에서 식당 예약을 진행하는데에 필요한 속성 중 인원 수와 위치가 부재되어 있어, 시스템은 이후의 대화에서 사용자에게 인원 수와 예약하고자 하는 식당의 위치를 물어본다. 필요한 속성들이 충족되면 시스템은 해당 속성들의 조건을 만족하는 api를 호출한다. 만약 사용자가 속성 값을 변경하면 시스템은 다른 변경사항이 또 있는지 사용자에게 물어보고 변경 사항이 없을 경우 변경된 조건을 만족하는 api를 호출하고 호출 결과 반환되는 식당 리스트를 사용자에게 제안한다. 사용자가 시스템이 제안한 식상 리스트를 만족할 경우, 시스템이 예약을 진행하며 대화가 종료된다.

[그림 4] 식당 예약을 진행하는 대화 시스템 모델

4. 결론

본 논문에서는 Hybrid Code Network를 기반으로 한국어로 구성된 식당 예약에 관련한 대화 데이터셋을 이용하여 식당 예약을 목적으로하는 대화 시스템을 구축하는 방법을 제안하였다. 실험 결과 구축한 대화 시스템은 도메인에 해당하는 특정 지식을 액션 템플릿의 정의를 통해 표현함으로써 보다 적은양의 학습 데이터 양으로 도메인 지식을 표현할 수 있었다. 또한 대화 도중 사용자가 식당 예약에 관련된 속성 값을 변경하고자 할 때에도 시스템이 해당 변경사항을 반영하여 식당 예약을 진행할 수 있었다. 실험 결과 본 시스템은 응답 별 정확도 95%와 대화 별 정확도 63%의 성능을 나타냈다.

본 실험에서는 사용자의 발화에서 개체를 추출하기 위해 각 속성에 해당하는 개체 사전을 정의하여 이를 기반으로 개체 속성을 추출하였는데, 향후 연구에서는 개체 사전에 나타나지 않는 OOV(out of vocabulary) 개체에 대해서도 올바른 속성 추출이 가능할 수 있도록 연구할 필요가 있다.

Acknowledgement

본 연구는 문화체육관광부 및 한국콘텐츠 진흥원의 2017년도 문화기술 연구개발 지원 사업으로 수행되었 음. [2017. 전통문화 융복합 지원을 위한 지능형 검색 플랫폼 구축]

참고문헌

[1] Lemon, O , Georgila, K, Henderson, J, and Stuttle, M. “An isu dialogue system exhibiting reinforcement

learning of dialogue policies: generic slot-filling in the talk in-car system,” In Proceedings of the 11th. Conference of the European Chapter of the ACL: Posters & Demonstrations, pages 119-122. 2006

[2] Wang, Z. and Lemon, O. “A simple and generic belief tracking mechanism for the dialog state tracking challenge: On the believability of observed information,” In Proceedings of the SIGDIAL Conference. 2013

[3] Young, S, Gasic, M, Thomson, B , and Williams, J. D. “Pomdp-based statistical spoken dialog systems: A review,” Proceedings of the IEEE, 101(5), 1160-1179. 2013

[4] Dzmitry, B, Kyunghyun, C, and Yoshua, B. “Neural machine translation by jointly learning to align and translate,” In ICLR. 2015

[5] Oriol, V, Ł ukasz, K, Terry, K, Slav P, Ilya, S, and Geoffrey, H. “Grammar as a foreign language,” In Advances in Neural Information Processing Systems, pages 2755- 2763. 2015

[6] Jason D Williams, Kavosh Asadi, and Geoffrey

Zweig. 2017. Hybrid code networks: practical and efficient end-to-end dialog control with supervised and reinforcement learning. In ACL.

[7] Bordes,A, and Weston,J, “Learning End-to-End Goal-Oriented Dialog,” ArXiv e-prints. 2016

[8] 이동엽, 허윤아, 임희석, “Hybrid Code Network를 이용한 한국어 식당 예약 시스템 모델,”컴퓨터교육학회. 2017.08.

[9] 이동엽, 김경민, “Korean Restaurant Reservation” (2017), GitHub repository, https://github.com/JudeLee19/korean_restaurant_reservation

심층적 의미 매칭을 이용한 cQA 시스템 질문 검색

김선훈○, 장헌석, 강인호

네이버

{seonhoon.kim, heonseok.jang, once.ihkang}@navercorp.com

Question Retrieval using Deep Semantic Matching for Community Question Answering

Seon-Hoon Kim○, Heon-Seok Jang, In-Ho Kang

Naver Corporation

요 약

cQA(Community-based Question Answering) 시스템은 온라인 커뮤니티를 통해 사용자들이 질문을 남기고 답변을 작성할 수 있도록 만들어진 시스템이다. 신규 질문이 인입되면, 기존에 축적된 cQA 저장소에서 해당 질문과 가장 유사한 질문을 검색하고, 그 질문에 대한 답변을 신규 질문에 대한 답변으로 대체할 수 있다. 하지만, 키워드 매칭을 사용하는 전통적인 검색 방식으로는 문장에 내재된 의미들을 이용할 수 없다는 한계가 있다. 이를 극복하기 위해서는 의미적으로 동일한 문장들로 학습이 되어야 하지만, 이러한 데이터를 대량으로 확보하기에는 어려움이 있다. 본 논문에서는 질문이 제목과 내용으로 분리되어 있는 대량의 cQA 셋에서, 질문 제목과 내용을 의미 벡터 공간으로 사상하고 두 벡터의 상대적 거리가 가깝게 되도록 학습함으로써 의사(pseudo) 유사 의미의 성질을 내재화 하였다. 또한, 질문 제목과 내용의 의미 벡터 표현(representation)을 위하여, semi-training word embedding 과 CNN(Convolutional Neural Network) 을 이용한 딥러닝 기법을 제안하였다. 유사 질문 검색 실험 결과, 제안 모델을 이용한 검색이 키워드 매칭 기반 검색보다 좋은 성능을 보였다.

주제어: cQA, 질문 검색, 의미 매칭, 딥러닝

1. 서론

cQA(Community-based Question Answering) 시스템은 사용자들이 특정한 정보를 얻기 위해 질문을 등록하고, 해당 질문 분야에 대해 잘 알고 있는 사용자들이나 전문가들이 답변을 할 수 있는 시스템이다. 대표적인 cQA 시스템으로는 네이버 지식iN[1], Yahoo! Answers[2], Quora[3] 등이 있다.

웹 환경의 발전으로 많은 사람들이 cQA 시스템을 통해 정보 커뮤니케이션 활동을 하게 되었고, 새로운 질문에도 비교적 짧은 시간 내에 답변을 받을 수 있게 되었다. 또한, 이렇게 축적된 질문-답변 쌍들은 Database 화 되어 동일한 정보 요구가 있는 또 다른 사용자들이 검색을 통해 정보 획득을 할 수 있도록 도움을 준다.

하지만, 시스템 사용 빈도가 높아지면서 새로운 질문의 인입량이 늘어나게 되고, 이에 따라 우선순위에 밀려 답변을 받지 못한 채 뒤로 밀려나는 질문들이 생기기도 한다. 또한, 검색을 통해 관련 정보를 얻으려 해도 키워드 매칭 기반의 검색으로는 단어는 다르지만 의미적으로 동일한 질문을 검색 하기가 쉽지 않다.

특정 질문이 작성되었을 때, 기존에 저장된 질문-답변 DB 에서 유사한 질문을 찾아 그에 해당하는 답변을 바로 줄 수 있다면, 답변 받지 못한 채로 남아있는 질문 비율은 많이 줄어들 것이다. 또한, 유사 질문 검색에 있어서, 단순 키워드 매칭을 넘어 의미적인 유사도도 판단할 수 있게 된다면 더 많은 질문에 대해 답변을 찾아 줄 수 있게 된다. 그리고, 특정 정보를 얻기 위해 단순 검색을 하는 사용자들도 키워드의 나열이 아닌 자연스러운 질문을 통해 원하는 답변을 얻을 수 있게 된다.

의미적으로 유사한 성질을 학습하기 위해서는 의미적으로 동일한 문장 쌍의 형태로 학습 데이터가 구성되어야 하지만, 이러한 데이터를 대량으로 확보하기는 쉽지 않다. 본 논문에서는 네이버의 대표적인 cQA 서비스인 지식iN에서 질문 제목과 질문 내용 데이터를 이용함으로써 이러한 한계를 극복하였다. 사용자들이 질문을 작성할 때, 질문 내용을 대표할 수 있는 문장으로 질문 제목을 작성한다. 질문 제목과 내용은 완벽하게 동일한 의미는 아니지만 제목이 내용에 대한 대표성을 띠고 있기 때문에 의사(pseudo) 유사 문장으로 간주하고, 이들의 유사도를 높게 학습함으로써 의미적인 성질을 부여하였다. 예를 들어, 그림 1과 같이 질문 제목에는 “하복부 통증”이라는 문구가 질문 내용에서 “아랫배가 아픈데” 라는 문구와 연결이 되어 있는 것을 볼 수 있다. 제목과 내용을 semi-training word embedding 과 CNN(Convolutional Neural Network) [1, 2] 을 이용하여 의미적 벡터 공간

[1] http://kin.naver.com

[2] https://answers.yahoo.com

[3] https://www.quora.com

으로 사상하고, 이들을 벡터 공간상의 가까운 곳에 위치하게 함으로써 이러한 의미적 연결고리를 부여하였다. 새로운 질문이 주어지면 이를 DB에 저장된 질문들과 함께 의미적 벡터 공간으로 사상하여 벡터 유사도를 계산함으로써 질문 검색을 수행하였다. 본 논문의 실험 결과 키워드 매칭 기반 검색보다 딥러닝을 통해 의미 벡터 공간으로 사상 후 검색하는 방법이 더 좋은 결과를 얻을 수 있었다.

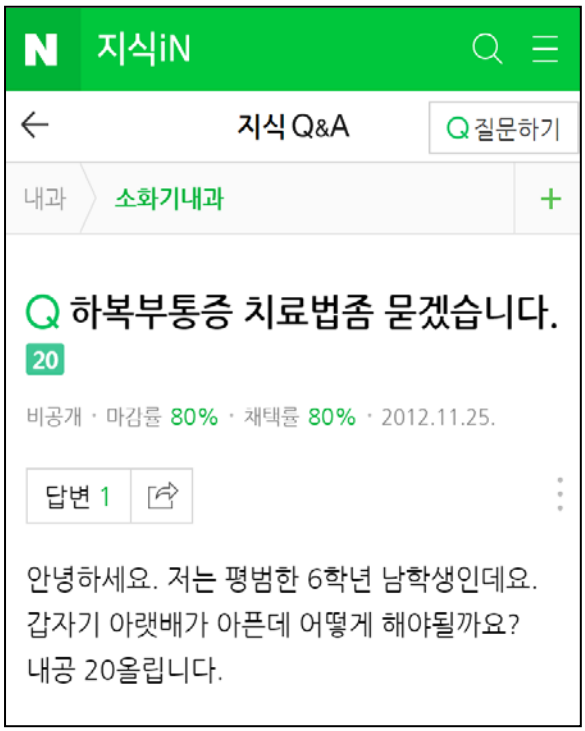


그림 1. 네이버 지식iN 질문 구성 예

2. 관련 연구

cQA 시스템은 질문-답변 DB 에서 사용자의 신규 질문과 유사한 질문을 찾는 것이 중요하다. 유사한 질문을 찾는다면 해당 질문에 달린 답변을 신규 질문에 대한 답으로 제공할 수 있다.

질문간의 유사도를 구하는 방법으로는 TF-IDF, BM25, vector space model 등의 방법들이 연구 되었다. 또한 질문간의 번역 확률로서 유사도를 계산하는 번역 기반 모델과 질문들 간의 topic 정보를 사용하는 topic model 방법들이 연구 되었다. [3, 4]는 knowledge base 로 Wordnet [5] 사전 과 TF-IDF 를 활용한 vector space model 을 이용해 질문과 질문간의 유사도를 계산하였다. [6]에서는 단어와 단어 단위, [7]에서는 구문과 구문 단위 번역 모델을 이용하여 질문간의 유사도를 계산하였다. [8]은 단어 단위의 매칭에 topic model 을 추가하여 질문 검색을 수행하였다. [9]에서는 topic 과 질문에서의 핵심 부분(focus) 을 추출한 뒤 이를 language model 과 조합하여 질문을 검색하였다.

최근에는 이미지, 음성, 자연어 등 여러 도메인에서 각광을 받고 있는 딥러닝을 이용한 방법이 많이 연구 되고 있다. [10]은 단어의 워드 임베딩 벡터와 카테고리 정보를 이용하여 유사도 계산의 feature 로 사용하였다.

[11]은 CNN 과 Bag-of-words 의 도합으로 유사 질문을 검색하였다. 질의와 질문을 CNN 을 통해 feature 를 추출하고, 이들을 결합한 후 추가적인 MLP(Multi Layer Perceptron) 를 통해 유사도를 계산하였다. [12] 에서는 RNN(Recurrent Neural Network) 의 기법중 하나인 LSTM (Long Short-Term Memory) 을 통해 질문과 답변의 feature 를 추출하고, 전통적인 자연어 기반 feature 를 결합하여 추가적인 MLP 를 통해 유사도를 계산하였다.

하지만 이런 방법은 질문과 질문간의 feature 를 결합한 후 또다시 MLP 연산을 해야 하므로, 대량의 질문 셋에서 실시간으로 유사도를 추정하기가 쉽지는 않다.

본 논문에서는 질문 자체에 대한 벡터 표현(representation)을 추출함으로써 각 질문간의 벡터 거리 계산만으로 최종 유사도를 결정할 수 있도록 하였다. 또한, pre-trained 된 word vector 의 일부만 학습하고 일부는 고정한 채 사용하는 semi-training word embedding 과 CNN 을 결합하여 의미적인 연결을 더 강화할 수 있도록 하였다.

3. 의미 매칭을 이용한 질문 검색 모델

본 논문에서는 네이버의 대표적인 커뮤니티 기반의 질의응답 시스템인 지식iN 데이터를 이용하여 의미적 질문 검색 모델을 구성하였다. 질문을 의미 공간으로 사상하기 위하여 Semi-training Word Embedding 과 CNN(SWECNN)을 이용하였고, 질문 제목과 질문 내용의 벡터 유사도를 높게 학습함으로써 의미적인 연결고리를 부여하였다.

3.1. 학습 데이터

문장간의 의미적 유사도를 판단하기 위해서는 유사 의미의 문장 데이터 셋이 존재해야 한다. 하지만, 이러한 데이터를 구축하기 위해서는 상당한 시간과 비용이 발생한다. 본 논문에서는 질문이 제목과 내용으로 구성된 지식iN 데이터를 이용하여 이러한 한계점을 극복하였다. 그림 1과 같이 질문 제목은 질문 내용에 대해 어느 정도의 대표성을 지닌다고 볼 수 있기 때문에, 이들을 의사 유사 문장 쌍으로 간주하였다. 그리고, 이들의 의미 공간 내 거리를, 질문 제목과 다른 임의의 질문 내용간의 거리보다 상대적으로 더 가깝게 되도록 학습함으로써 의미적인 연결고리를 부여하였다. 즉, 질문 제목에 대해 같은 쌍인 질문 내용을 positive sample 로 설정하였고, 같은 쌍이 아닌 임의의 질문 내용을 negative sample 로 설정하였다. negative sample 은 질문 제목 당 4개를 사용하였다. 질문 제목과 질문 내용이 연관이 없는 내용을 가지고 있는 경우를 필터링 하기 위하여 질문 제목과 질문 내용의 교집합 단어의 개수가 질문 제목 단어 개수의 절반이 안 되는 샘플들은 제거하였다.

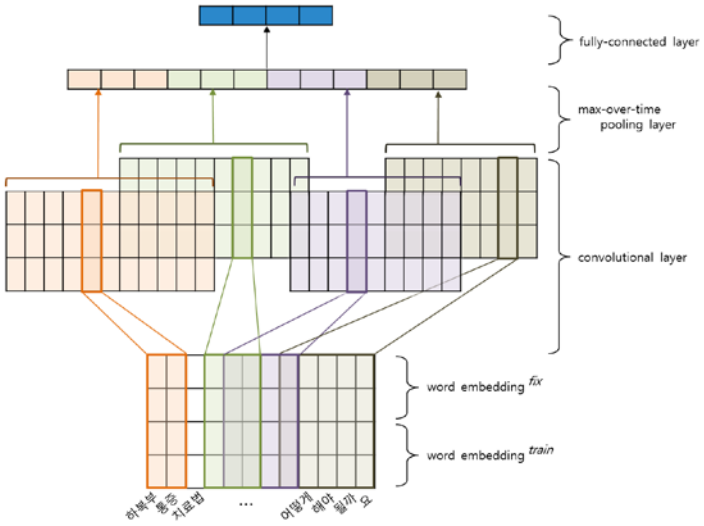


그림 2. 의미 매칭 모델(SWECNN). Pre-trained word embedding 두 개 중 하나는 고정하고(fix), 다른 하나는 fine-tuning(train) 하였다. Word embedding 된 값을 convolutional 연산을 이용하여 feature map 을 추출하고, 이를 max-over-time pooling 을 이용하여, 가장 높은 특징 값들을 추출하였다. 마지막으로 fully-connected layer를 통해 의미 벡터를 추출하였다.

3.2. word representation layer

Word embedding 을 이용하기 위해 별도의 코퍼스를 구축하였고, GloVe [13] 방법을 이용하여 300차원의 word embedding vector 를 학습하였다. 이렇게 학습된 동일한 word embedding vector 두 개를 식 (1) 과 같이 연결하여 semantic matching model 의 입력으로 사용하였다. 식 (1) 에서 $\oplus$ 는 연결 연산이다.

$$x_i = x_i^{fix} \oplus x_i^{train} \quad (1)$$

i 번째 단어의 word vector $x_i^{fix} \in \mathbb{R}^{300}$ 는 학습이 끝날 때까지 vector 값을 고정하였고, i 번째 단어의 또 다른 word vector $x_i^{train} \in \mathbb{R}^{300}$ 은 의미 매칭 모델 학습 시에 fine-tuning 하였다. x_i^{fix} 와 x_i^{train} 을 연결해 구성한 최종 word vector $x_i \in \mathbb{R}^{600}$ 를 통해 기존의 워드 벡터가 가지고 있는 구조적, 의미적 특징은 물론, 본 논문의 데이터와 모델에 맞게 학습되는 문장 단위의 의미적인 정보도 내재될 수 있도록 하였다.

3.3. 의미 매칭 모델(SWECNN)

그림 2와 같이 Semi-training Word Embedding 과 CNN 을 이용한 SWECNN 의미 매칭 모델을 구성하였다. CNN 의 입력으로 질문의 각 단어 별 embedding vector x_i 를 사용하였고, convolution 연산을 위해 filter $w \in \mathbb{R}^{h \times 600}$ 를 사용하였다. h는 convolution 연산의 window 사이즈이고, window 만큼의 word embedding vector 가 filter 와의 연산을 통해 한 개의 feature 로 추출된다. 본 논문에서는 2, 3, 4, 5 의 window 크기를 갖는 4 종류의 filter 를 각 300개씩 사용하였다. Convolution 연산 후에는 vanishing gradient 에 좋은 효과를 보이는 Rectified linear unit (ReLU) f_{ReLU} 를 non-linear 함수로 사용하였다[15].

$$f_{ReLU}(x) = \max(0, x) \quad (2)$$

ReLU 함수 적용 후 max-over-time pooling 을 통해 가장 중요한 활성 값(activation value) 만 남기도록 하였고, 4 종류의 filter 별로 300차원의 feature 를 추출하였다. 추출된 각 feature 들을 모두 인접시키고 1개의 fully-connected layer 를 연결하여 최종적인 300 차원의 semantic 질문 벡터를 추출하였다 (그림 2).

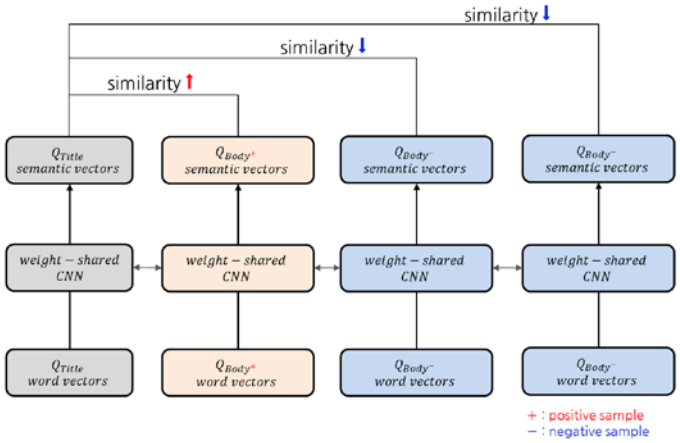


그림 3. Weight-shared network 을 이용한 시맨틱 매칭 모델의 학습. 질문 제목과 질문 내용의 word vector 를 weight 를 공유하는 CNN 의 입력으로 하여, 각 질문의 semantic vector 를 추출한다. 그리고, Q_{Title} 과 positive Q_{Body^+} 의 벡터 유사도는 높게, Q_{Title} 과 negative Q_{Body^-} 의 벡터 유사도는 낮게 학습함으로써, 의미 정보를 내재화하였다.

3.4. weight-shared network 를 이용한 학습

본 논문에서는 질문 제목 Q_{Title} 과 같은 쌍인 질문 내용 Q_{Body^+} 를 positive sample 로, 질문 제목과 같은 쌍이 아닌 임의의 질문 내용 Q_{Body^-} 를 negative sample 로 사용하였다. 그리고 각 질문들을 weight 를 공유[15]하는 CNN 의 입력으로 사용하여 각 질문의 의미 벡터 y_{Title} 과 $y_{Body^{+/-}}$ 를 추출하였다. 그리고 질문 제목과 positive 질문 내용의 유사도 $S(Q_{Title}, Q_{Body^+})$ 를 높게, 질문 제목과 negative 질문 내용의 유사도 $S(Q_{Title}, Q_{Body^-})$ 를 낮게 학습하였다 (그림 3). 유사도 S 는 식 (3) 과 같이 코사인 유사도를 이용하였다.

$$S(Q_{Title}, Q_{Body}) = \cos(y_{Title}, y_{Body}) = \frac{{y_{Title}}^T y_{Body}}{\|y_{Title}\|\|y_{Body}\|} \quad (3)$$

[16]과 같이, 질문 제목과 positive 질문 내용의 상대적 유사도를 식 (4) 의 softmax 함수를 이용하여 posterior 확률로 간주하였다.

$$P(Q_{Body^+}|Q_{Title}) = \frac{\exp(\gamma\, S(Q_{Title}, Q_{Body^+}))}{\sum_{Body' \in \boldsymbol{Body}} \exp(\gamma\, S(Q_{Title}, Q_{Body'}))} \quad (4)$$

식 (4) 에서 $\boldsymbol{Body}$ 는 모든 질문 내용의 집합이고, γ 는 softmax 함수의 smoothing factor 이다. Loss 함수는 식 (5) 와 같이 cross entropy 함수를 이용하였다.

$$Loss = -\log P(Q_{Body^+}|Q_{Title}) \quad (5)$$

Loss 함수에 대한 gradient를 구하기 위해 back-propagation [17]을 이용했고, Stochastic Gradient Descent 방법 중 하나인 RMSProp [18]으로 모델 파라미터들을 학습하였다.

4. 실험 및 평가

본 논문에서 사용한 모델의 효과를 검증하기 위하여 약 230만건의 지식iN 질문-답변 문서를 추출하고, 실제 사용자들이 수행하였던 질문들 중 임의의 200건에 대하여 얼마나 의미적으로 동일한 질문을 가진 문서들이 검색 되는지 실험하였다. 의미 벡터 유사도를 이용한 검색 방법으로는 Approximate Nearest Neighbor [19] 검색을 이용하였다. 검색 결과 평가를 위하여 서로 상의하지 않은 세 명의 평가자가 평가 질문과 검색된 질문이 의미적으로 유사한지 아닌지 각각 판단을 하였다. 각 샘플은 최소 두 명의 동의를 받은 결과를 최종 결과로 간주하였다. 예를 들어, 특정 샘플에 대해 두 명의 평가자가 '유사하다' 고 판단하고, 다른 한 명의 평가자는 '유사하지 않다' 라고 판단했을 경우, 해당 샘플은 '유사하다'로 최종 판단 하였다. 모델 별 평가 기준으로는 P@K (Precision at K) 를 사용하였다. 본 논문에서 사용한 신경망 모델과의 비교를 위하여 BM25 와 pre-trained word vector 들의 평균값(avg-emb)을 baseline 으로 이용하였다. 추가적으로 제안 모델에서 사용한 semi-training word embedding 방법에 대한 효과 검증을 위하여 emb_300^{fix}, emb_300^{train}, emb_600 의 방법과 비교를 하였다. emb_300^{fix}, emb_300^{train} 그리고 emb_600 의 경우 제안 모델에 대한 word embedding layer 의 변형된 모델이고 word embedding 이후의 CNN 및 fully-connected layer 의 아키텍처는 동일하다.

- BM25 : 학습셋으로 BM25 모델을 구축하고, 평가 질문과 지식iN 질문간의 BM25 score 를 계산한 후 가장 높은 질문들을 추출하였다.
- avg-emb : pre-trained 된 word vector에 대해, 평가 질문의 단어 벡터 평균과 지식iN 질문의 단어 벡터 평균의 유사도를 비교하여 가장 score 가 높은 질문들을 추출하였다.
- emb_300^{fix}+CNN: pre-trained 된 word vector 를 사용하고 의미 매칭 모델을 학습할 때 word vector를 고정하였다.
- emb_300^{train}+CNN : pre-trained 된 word vector 를 사용하고 의미 매칭 모델 학습시에 fine-tuning 하였다.
- emb_600+CNN : 제안 모델과 동일한 파라미터 개수로 비교하기 위하여 600 차원의 word vector 를 설정하여 임의의 값으로 초기화 한 후 의미 매칭 모델 학습시에 같이 학습하였다.

의미 매칭 모델의 비교 평가 결과는 표 1 과 같다. 기존 키워드 매칭 검색 방법인 BM25 의 경우 가장 좋지 않은 성능을 보였고, 본 논문의 제안 모델과는 큰 성능 차가 있었다. word embedding 과 CNN을 통해 의미 벡터 공간으로 질문을 사상한 후 검색하는 방법이 키워드 매칭 방법보다 탁월함을 알 수 있었다. 또한 word embedding 의 단순 평균만을 이용했던 avg-emb 도 BM25 보다는 좋은 결과를 얻을 수 있었다. 이는 단순 word embedding 만으로도 각 단어가 내포하고 있는 의미를 잘 표현해줄 수 있다고 볼 수 있다. 그리고 word embedding 을 처음부터 학습하는 emb_600 보다는 pre-trained word embedding 과 CNN을 결합한 방법들이 더 좋은 성능을 보였다. 이는 pre-trained word embedding 이 의미 매칭 학습 데이터에서 얻지 못한 단어 별 의미 정보를 보유하고 있기 때문으로 분석이 된다. emb_300^{fix} 와 emb_300^{train} 의 비교 결과는 큰 차이는 없으나 emb_300^{train} 이 근소하게 높은 성능을 보였다. 본 논문에서 제안한 SWECNN 모델이, P@1 기준 51.5% 로 다른 비교 모델들보다 가장 우수한 결과를 보임을 확인할 수 있었다.

표 1. 유사 질문 검색 실험에 대한 결과

모델	P@1 (%)	P@3 (%)	P@5 (%)
BM25	32.5	25.3	20.9
avg-emb	42	34	28.9
emb_300^{fix}+CNN	47	35.8	30
emb_300^{train}+CNN	48	37.3	31.8
emb_600+CNN	43.5	33.8	27.9
SWECNN (proposed model)	**51.5**	**40**	**33.2**

표 2. 제안 모델과 다른 비교 모델간의 샘플 비교 결과

평가 질문 1. 발바닥 아픈데 어느 병원으로 가야 하나요?	
BM25	아랫배가 너무 아픈데 어느 병원으로 가야하나요? 양쪽어깨가 아프고 종아리와 발바닥 통증이 있다면 어느과 병원을 가야하나요?
SWECNN	발바닥 아플때는 어느병원에 가야 되나요 발바닥 통증...어디병원을 가야할까요.
평가 질문 2. 겨드랑이 털은 보통 언제 나나요?	
word2vec	겨드랑이 레이저 제모 후 언제쯤 털이 안나요? 팔 털 제모하면 털이 더 나나요?
SWECNN	겨드랑이 털나는시기 겨드랑이 털이 나기시작하는데요...
평가 질문 3. 일본 동경에서 가볼만한곳은?	
emb_300^{fix}	도쿄 가볼만한곳 오사카 저녁시간에 가볼만한 곳
SWECNN	도쿄 가볼만한곳 일본 도쿄 가볼만한곳 추천좀해주세요
평가 질문 4. 햇빛많이 받으면 머리아파요?	
emb_300^{train}	햇빛을 보면 머리가 아파요 요즘 머리가아파요
SWECNN	햇빛을 보면 머리가 아파요 여름에만 머리가 아파요
평가 질문 6. 비타민디가 들어있는 음식?	
emb_600	비타민 c가 들어있는 음식 무엇이 있는가요 비타민씨 풍부한 음식??
SWECNN	비타민 d 많은 음식 비타민D 가득한음식!

표 2 는 제안된 모델과 다른 모델간의 비교 샘플이다. 평가 질문 1 에서는 BM25 를 이용한 키워드 매칭 방법과의 비교이다. BM25는 '아픈데', '어느 병원으로 가야 하나요' 등의 단순 키워드를 이용하여 검색을 하였지만, 검색된 질문은 평가 질문과 의미적으로 큰 차이를 보였다. 제안 모델은 '발바닥' 에 대하여 '아픈데' 와 '통증' 을 의미적으로 연결하였다. 평가 질문 2 에서는 avg-emb 모델과 비교 하였다. avg-emb 는 단순 word embedding 만을 사용했기 때문에, 신체 일부인 '겨드랑이' 와 '팔' 이 가깝게 매칭 되었고 검색된 질문은 평가 질문과 의미가 일치하지 않는다. 제안 모델은 '언제 나나요' 와 '시기' 에 대하여 의미적으로 바르게 연결이 되었다. 평가 질문 3 에서는 emb_300^{fix} 의 경우 '동경' 과 '도쿄'가 잘 연결이 되기도 하였지만, pre-trained 된 word embedding 의 고정된 값만 사용해서 '도쿄' 와 '오사카' 가 유사하게 검색 되기도 하였다. 평가 질문 4 에서는 제안 모델의 경우 '여름에만 머리가 아파요' 라는 질문이 검색 되었다. 단순히 보면 의미적으로 일치하지 않지만 햇빛이 많은 계절이 여름이라는 것을 보면 '햇빛많이' 와 '여름' 의 경우 어느 정도는 상호간에 의미가 연결 되었음을 확인할 수 있었다. 평가 질문 5 에서는 제안 모델의 경우 '비타민디' 가 '비타민 d' 로 연결이 되었다. '디' 와 'd' 의 경우 pre-trained word embedding 의 효과로 상호간 연결이 되었지만, 학습된 word embedding 을 사용하지 않은 emb_600 의 경우는 '디' 와 'd' 의 연결점을 찾지 못하였다.

모델 비교 실험 결과, 질문 제목과 내용을 이용하는 의사 의미 학습에는 pre-trained word embedding 이 중요하며, 그 중 제안 모델에서 이용하는 semi-training word embedding 의 성능이 탁월함을 확인 할 수 있었다.

5. 결론

본 논문에서는 의미 매칭 모델링을 위한 대량의 학습 데이터를 구축하기 위하여 지식iN cQA 셋에서 질문 제목과 내용을 이용하는 방법을 제안하였다. 질문 제목과 내용을 각각 의미 벡터 공간으로 사상한 후, 이들의 거리를 가깝게 되도록 학습함으로써 의사 의미 연결고리를 부여하였다. 의미 벡터 추출로는 Semi-training Word Embedding과 CNN(SWECNN)을 이용하는 딥러닝 모델을 제안하였다. 모델 별 비교 실험 결과 SWECNN 이 가장 좋은 성능을 보임을 확인 할 수 있었다.

향후 연구로는, 단어 단위 데이터와 음절 또는 자소 단위의 데이터도 함께 이용함으로써 Out-of-vocabulary

의 한계를 극복하고자 한다. 또한, 카테고리와 같은 부가 정보를 추가적으로 이용한다면 더 정밀한 질문 검색을 할 수 있을 것이다.

참고문헌

[1] LeCun, Y., Kavukcuoglu, K., & Farabet, C. Convolutional networks and applications in vision. In Circuits and Systems (ISCAS), Proceedings of 2010 IEEE International Symposium on, pp. 253-256. IEEE, 2010.

[2] Kim, Y. Convolutional neural networks for sentence classification. In EMNLP, 2014.

[3] Burke, R. D., Hammond, K. J., Kulyukin, V., Lytinen, S. L., Tomuro, N., & Schoenberg, S. Question answering from frequently asked question files: Experiences with the faq finder system. AI magazine, 18(2), 57. 1997.

[4] Song, W., Feng, M., Gu, N., & Wenyin, L. Question similarity calculation for FAQ answering. In Semantics, Knowledge and Grid, Third International Conference on, pp. 298-301. IEEE, 2007.

[5] Fellbaum, C. WordNet. John Wiley & Sons, Inc., 1998.

[6] Jeon, J., Croft, W. B., & Lee, J. H. Finding similar questions in large question and answer archives. In Proceedings of the 14th ACM international conference on Information and knowledge management, pp. 84-90. ACM, 2005.

[7] Zhou, G., Cai, L., Zhao, J., & Liu, K. Phrase-based translation model for question retrieval in community question answer archives. In Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1, pp. 653-662. 2011.

[8] Zhang, K., Wu, W., Wu, H., Li, Z., & Zhou, M. Question retrieval with high quality answers in community question answering. In Proceedings of the 23rd ACM international conference on conference on information and knowledge management, pp. 371-380. ACM, 2014.

[9] Duan, H., Cao, Y., Lin, C. Y., & Yu, Y. Searching Questions by Identifying Question Topic and Question Focus. In ACL, Vol. 8, pp. 156-164. 2008.

[10] Zhou, G., He, T., Zhao, J., & Hu, P. Learning Continuous Word Embedding with Metadata for Question Retrieval in Community Question Answering. In ACL (1), pp. 250-259. 2015.

[11] Dos Santos, C., Barbosa, L., Bogdanova, D., & Zadrozny, B. Learning hybrid representations to retrieve semantically equivalent questions. In Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing, Vol. 2, pp. 694-699. 2015.

[12] Nassif, H., Mohtarami, M., & Glass, J. Learning semantic relatedness in community question answering using neural models. ACL, 137. 2016.

[13] Pennington, J., Socher, R., & Manning, C. Glove: Global vectors for word representation. In Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP), pp. 1532-1543. 2014.

[14] Nair, V., & Hinton, G. E. Rectified linear units improve restricted boltzmann machines. In Proceedings of the 27th international conference on machine learning (ICML-10), pp. 807-814. 2010.

[15] Chopra, S., Hadsell, R., & LeCun, Y. Learning a similarity metric discriminatively, with application to face verification. In Computer Vision and Pattern Recognition, IEEE Computer Society Conference on, vol. 1, pp. 539-546. IEEE, 2005.

[16] Huang, P. S., He, X., Gao, J., Deng, L., Acero, A., & Heck, L. Learning deep structured semantic models for web search using clickthrough data. In Proceedings of the 22nd ACM international conference on Conference on information & knowledge management, pp. 2333-2338. ACM, 2013.

[17] Rumelhart, D. E., Hinton, G. E., & Williams, R. J. Learning representations by back-propagating errors. Cognitive modeling, 5(3), 1. 1988.

[18] Hinton, G., Srivastava, N., & Swersky, K. RMSProp: Divide the gradient by a running average of its recent magnitude. Neural networks for machine learning, Coursera lecture 6e. 2012.

[19] Kalantidis, Y. and Avrithis, Y. Locally optimized product quantization for approximate nearest neighbor search. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 2321-2328. 2014.

CNN-LSTM 신경망을 이용한 발화 분석 모델

김민경○, 김학수
강원대학교 컴퓨터정보통신공학과
kmink0817@kangwon.ac.kr, nlpdrkim@kangwon.ac.kr

Utterance Intention Analysis Using CNN-LSTM Neural Network

Min-Kyoung Kim○, Harksoo Kim
Kangwon National University Computer and Communication Engineering

요 약

대화시스템이 적절한 응답을 제시해 주기 위해서는 사용자의 의도를 분석하는 것은 중요한 일이다. 사용자의 의도는 도메인에 독립적인 화행과 도메인에 종속적인 서술자의 쌍으로 나타낼 수 있다. 사용자 의도를 정확하게 분석하기 위해서는 화행과 서술자를 동시에 분석하고 대화의 문맥을 고려해야 한다. 본 논문에서 제안하는 모델은 합성곱 신경망에서 공유 계층을 이용하여 화행과 서술자간 상호작용이 반영된 발화 임베딩 모델을 학습한다. 그리고 순환 신경망을 통해 대화의 문맥을 반영하여 발화를 분석한다. 실험 결과 제안 모델이 이전 모델들 보다 높은 성능 (F1-measure로 화행에 대해 0.973, 서술자 0.919)을 보였다.

주제어: 화행, 서술자, 공유계층, CNN-LSTM

1. 서론

목적 지향 대화시스템은 한정된 도메인 안에서 사용자 발화(utterance)에 대해 적절한 응답을 제시해 주는 시스템을 말한다. 목적 지향 대화시스템은 사용자와 자연스럽게 의사소통하기 위해 발화에 내포된 화자의 의도를 분석하는 것이 중요하다. 본 논문에서는 발화에 내포된 의도를 화행(speech-act)과 서술자(predicator)의 쌍으로 표현한다. 화행은 도메인에 독립적으로 사용자가 전달하고자 하는 일반적인 의도를 나타낸다. 서술자는 도메인에 종속적이며 주된 서술어의 의미 범주를 나타낸다. 표 1은 일정 관리 도메인에서 목적 지향 발화의 예와 해당 의도를 보여준다.

표 1 목적 지향 대화의 예

	발화	의도	
		화행	서술자
U	(1) 안녕~	Greeting	Null
S	(2) 무엇을 도와드릴까요?	Opening	Null
U	(3) 약속 잡아줘	Request	Update -appointment
S	(4) 날짜는 언제로 할까요?	Ask-ref	Update-date
U	(5) 10월 8일	Response	Update-date

화행과 서술자는 문맥에 의존적이기 때문에 하나의 발화만으로 추론하는 것은 매우 어렵다. 예를 들어 표 1의 발화 (5)는 두 가지 의도로 분석이 가능한데 현재 설정되어 있는 일정을 알려주는 "Inform & Select-date"와 일정이 뭐로 변경되었는지를 묻는 질문에 대해 답해주는 "Response & Update-date"가 될 수 있다. 이러한 모호성을 해결하기 위해 발화 (5)에 문맥이 반영되어야 한다. 위 예시에서 바로 이전 발화인 (4)를 고려하면 발화 (5)의 올바른 의도인 "Response & Update-date"를 선택할 수 있다.

2. 관련 연구

사용자 의도를 분석하기 위해 다양한 자질에 기반을 둔 기계 학습 모델들이 제안되었지만, 기존의 연구들은 주로 발화의 화행 분류만을 다루었거나[1][2] 화행과 서술자를 개별적으로 다루어왔다[3]. 그러나 사용자의 의도를 더 정확히 파악하기 위해 화행과 서술자를 동시에 식별할 필요가 있다. 관련 연구로 [4]는 서술자 예측 결과가 화행 분류를 위한 입력으로 사용되는 통합 신경망 모델을 제안하였다. [5]는 통합 모델의 성능 향상을 위해 상호 재학습 방법을 제안하였다. [5]가 제안한 모델은 화행 분류 모델과 서술자 분류 모델로 나뉘며 학습 동안 한 모델의 출력 값을 다른 모델의 입력 자질로 사용한다. 본 논문에서는 합성곱 신경망(Convolutional Neural Network)[6]와 LSTM 순환 신경망(Long Short-Term Memory Recurrent Neural Network)[7]을 이용하여 화행과 서술자를 동시에 분석하는 발화 분석 모델을 제안한다. 제안된 모델은 합성곱 신경망을 기반으로 한 새로운 발화 임베딩 방법을 이용하여 화행과 서술자간의 상호작용이 가능하게 한다. 그리고 LSTM 순환 신경망을 기반으로 대화의 문맥을 반영하여 의도 분석의 성능을 향상시킨다.

3. 발화 분석 모델

본 논문에서 제안하는 발화 분석 모델의 구조도는 [그림 1]과 같다. 발화 분석 모델은 화행 분류 모델(SA Classifier)과 서술자 분류 모델(PR Classifier), 발화 임베딩 모델(Utterance embedding model)로 구성된다.

대화의 문맥을 고려하여 화행과 서술자를 분류하기 위해 각 분류 모델은 LSTM 순환 신경망을 적용한다. 발화 임베딩 모델을 이용하여 m개의 발화($Utterance_{1,m}$)에 대해 화행 분류를 위한 임베딩 벡터($Emb_S^{1,m}$)와 서술자 분류를 위한 임베딩 벡터($Emb_P^{1,m}$)를 얻는다. 최종적으로 화행 분

류 모델과 서술자 분류 모델은 각각의 임베딩 벡터를 입력받아 화행과 서술자를 출력한다.

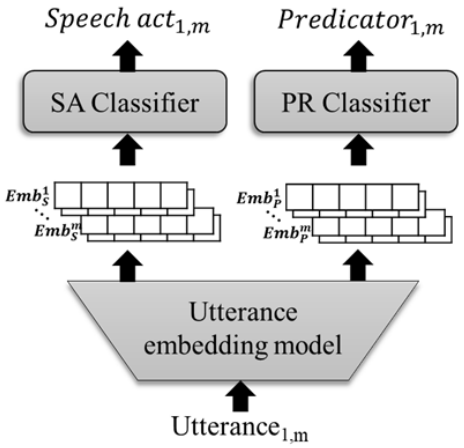


그림 1. 제안 모델의 구조도

아래 [그림 2]는 발화 임베딩을 위한 합성곱 신경망의 구조이다.

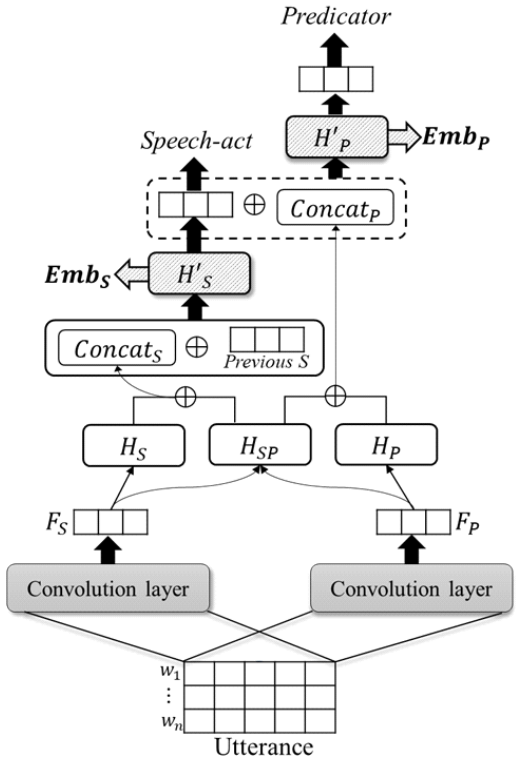


그림 2. 합성곱 신경망을 이용한 발화 임베딩 모델

[그림 2]에서 입력 발화의 각 단어 w_n은 50차원의 Word2Vec 임베딩 벡터이다[8]. 입력된 발화는 두 개의 독립된 convolution 계층을 통해 화행과 서술자에 적합한 자질 벡터 F_S, F_P를 생성한다. 은닉 노드 H_X는 자질 벡터 F_X만을 입력으로 하며($X \in S, P$), H_{SP}는 F_S, F_P를 모두 입력으로 한다. 이는 공유 계층으로 화행과 서술자의 조합된 정보를 추상화 할 수 있다. F_X를 입력으로 하는 은닉 노드들은 H'_X의 입력이 된다. 예를 들어, [그림 1]에서 은닉 노드 H_S와 H_{SP}가 H'_S의 입력이 된다. 화행을 분류할 때, 입력된 이전 화행(Previous S)을 자질로 사용하며 서술자를 분류할 때는 모델이 예측한 현재 화행을 자질로 사용한다. 모델을 학습할 때 예측 화행과 정답 화행간의 오류가 화행과 관련된 노드들(i.e., H'_S, H_S, H_{SP})로 부분적 역 전파되며, 같은 방식으로 서술자에 대한 오류가 역 전파된다. 학습이 완료된 임베딩 모델에서 H'_S와 H'_P를 각각 화행과 서술자를 분류하기 위한 임베딩 값 Emb_S, Emb_P로 이용한다.

4. 실험 및 평가

4.1 실험 준비

본 논문에서는 실험을 위해 일정 관리 도메인 대화 말뭉치를 실험 데이터로 사용하였다[9]. 실험 데이터는 899개의 대화로 구성되어 있으며 전체 발화 수는 10,043개(대화 당 평균 11개의 발화)이다. 실험에서 화행과 서술자 범주는 각각 11개, 47개를 사용하였다. 실험은 10배 교차 검증을 시행하였다. 제안 모델의 성능을 평가하기 위한 척도로 정확도(accuracy), 각 클래스별 정확률의 평균(macro precision:MP), 각 클래스별 재현율의 평균(macro recall:MR), macro F1-measure를 사용하였다.

제안 모델에 세부적인 파라미터는 아래 표 2와 같다.

표 2 모델 파라미터

	발화 임베딩 모델	발화 분석 모델
num epoch	300	300
batch size	64	16
learning rate	0.0001	0.0001

4.2 실험 결과

제안 모델의 성능을 확인하기 위해 동일한 실험 데이터를 사용한 다른 모델들과 비교하였다. 표 3은 제안 모델과 비교 모델들 사이의 성능 차이를 보여준다.

표 3 제안 모델과 이전 모델들의 성능 비교

		accuracy	MP	MR	F1
화행	제안모델	**0.985**	**0.989**	**0.958**	**0.973**
	[5]	0.941	0.878	0.915	0.896
	[4]	0.861	-	-	-
서술자	제안모델	**0.975**	**0.936**	**0.903**	**0.919**
	[5]	0.909	0.827	0.768	0.796
	[4]	0.738	-	-	-

[5]는 SVM을 기반으로 화행 분류모델과 서술자 분류모델의 출력이 서로의 입력이 되어 학습하는 상호 재학습 방법을 이용한 모델의 성능이고 [4]는 서술자 예측 결과가 화행 분류를 위한 입력이 되는 통합 신경망 모델의

성능 결과이다. 표 3에서 제안 모델은 자질 추출 및 선택에 많은 비용을 들이지 않고도 뛰어난 성능을 보이는 것을 확인할 수 있다. 특히 상호 재학습 방법을 이용하지 않고도 [5]보다 높은 성능을 보였다. 이는 공유계층을 이용한 임베딩한 방법이 화행과 서술자가 서로 상호작용하는데 도움을 준다는 것을 보여준다.

5. 결론

본 논문에서는 발화를 분석하기 위해 합성곱 신경망과 LSTM 순환 신경망을 결합한 모델을 제안하였다. 합성곱 신경망을 통해 화행과 서술자간의 상호작용이 반영되게 발화를 임베딩하고 LSTM 순환 신경망을 이용하여 대화의 문맥을 반영하였다. 실험결과 제안 모델이 자질 튜닝 없이도 비교 모델들 보다 뛰어난 성능을 보였다.

감사의 글

이 논문은 2016년도 정부(미래창조과학부)의 재원으로 한국연구재단의 지원을 받아 수행된 연구임 (No.2016R1A2B4007732)

참고문헌

[1] 은종민, 이성욱, 서정연, “지지벡터기계 (Support Vector Machines)를 이용한 한국어 화행 분석”, 정보과학회논문지(B), 제12권, 제3호, pp.365-368, 2005.

[2] S. Kang, H. Kim, J. Seo, “A reliable Multidomain model for speech act classification”, Pattern Recognition Letters, vol.31(1), pp.71-74, 2010.

[3] H. Lee, H. Kim, J. Seo, “Domain Action Classification Using a Maximum Entropy Model in a Schedule Management Domain”, AI Communications, vol.21(4), pp.221-229, 2008.

[4] H. Lee, H. Kim, J. Seo, “An Integrated Neural Network Model for Domain Action Determination in Goal-oriented Dialogues”, JIPS, vol.9(2), pp.259-270, 2013.

[5] C. Seon, H. Kim, J. Seo, “Improving Domain Action Classification in Goal-oriented Dialogues Using a Mutual Retraining Method”, Pattern Recognition Letters, vol.45, pp.154-160, 2014

[6] A. Krizhevsky, I. Sutskever, GE. Hinton, “Imagenet classification with deep convolutional neural networks”, Advances in neural information processing systems, pp.1097-1105, 2012.

[7] S. Hochreiter, J. Sschmidhuber, “Long short-term memory”, Neural computation, 1997.

[8] Y. Goldberg, O. Levy, “word2vec Explained: deriving Mikolov et al.’s negative-sampling word-embedding method”, arXiv preprint arXiv:1402.3722, 2014.

[9] H. Kim, C. Seon, J. Seo, “Review of Korean Speech Act Classification: Machine Learning Methods”, Computing Science and Engineering, vol.5(4), pp.288-293, 2011

색인어 인코딩과 음절 디코딩에 기반한 생성 채팅 모델

김진태○, 김시형, 김학수, 이연수*, 최맹식*
강원대학교 컴퓨터정보통신공학과, ㈜엔씨소프트*
wlsxo1119@kangwon.ac.kr, sureear@kangwon.ac.kr, nlpdrkim@kangwon.ac.kr, yeonsoo@ncsoft.com,
mschoi@ncsoft.com

Generative Chatting Model based on Index-Term Encoding and Syllable Decoding

JinTae Kim○, Sihyung Kim, HarkSoo Kim, Yeonsoo Lee*, Maengsic Choi*
Kangwon National University Computer and Communication Engineering
NCSOFT Corp.*

요 약

채팅 시스템은 사람이 사용하는 자연어를 이용해 컴퓨터와 대화를 하는 시스템이다. 한국어 특성상 대화체에서 동일한 의미를 가졌지만 다른 형태를 가진 경우가 많다. 본 논문에서는 Attention mechanism Encoder-Decoder Model을 사용해 한국어 특성에 맞는 효과적인 생성 모델을 만들 수 있는 입력, 출력 단위를 제안한다. 실험에서 정성 평가와 ROUSE, BLEU 평가를 진행한 결과 형태소 단위의 입력 보다 본 논문에서 제안한 색인어 입력 단위의 성능이 높고, 의사 형태소 단위 출력 보다 음절 단위 출력을 사용한 시스템이 더 문법적 오류가 적고 적합한 응답을 생성하는 것을 보였다.

주제어: 채팅 시스템, 색인어

1. 서론

채팅 시스템은 사람이 사용하는 자연어를 통해 컴퓨터와 사용자간 대화가 이루어지는 시스템이다[1]. 최근 채팅 시스템에 관한 연구가 빠르게 진행 되고 있다. 채팅 시스템은 검색 모델[2]과 생성 모델[1]로 구분된다. 검색 모델은 이미 정의된 발화-응답 쌍의 데이터 중 사용자의 입력과 가장 유사한 발화를 찾아 해당 발화의 정해진 응답을 하는 모델이다. 사용자의 입력과 가장 유사한 발화를 빠르게 찾기 위해서는 발화-응답 쌍의 데이터의 발화 데이터를 색인하여 사용하며, 사용자의 입력도 발화 데이터 색인 방법과 동일한 형태로 만들어 가장 유사한 색인된 발화 데이터를 찾는다. 색인 데이터는 비슷한 형태의 발화 데이터가 동일한 모습을 가지게 된다. 예를 들어 “안녕하세요”를 “안녕하세용”, “안녕하세여”와 같이 다른 형태를 보이지만, 의미가 같은 발화 데이터를 동일한 모습으로 변환하여 색인 데이터를 만든다. 본 논문에서는 검색 모델에서 사용하는 색인 데이터를 [3]Attention mechanism Encoder-Decoder 모델의 입력으로 사용하여 한국어 특성에 적합한 색인어 입력 단위를 사용하는 채팅 시스템을 제안한다.

2. 관련 연구

기본적인 Recurrent Neural Encoder-Decoder 모델을 사용한 생성 채팅 모델로 [4]가 있다. 한국어 채팅 시스템 연구로는 Attention mechanism을 적용한 [5]가 있다. Recurrent Neural Encoder-Decoder 모델을 사용한 채팅 생성 모델의 문제점 중 하나는 “I don’t know.”와 같이 일반적인 응답을 많이 하는 문제가 있다. 이를 목적함수를 변형해 다양한 응답을 출력하는 방법으로 [6]이 있다. 채팅 시스템을 검색 모델과 생성 모델을 결합하여 검색 모델의 단점과 생성모델의 단점을 보완하는 방법은 [7]이 있다. 추가적으로 다양한 자연어 처리 분야에서 Sequence-to-Sequence 구조를 사용한 연구가 진행되고 있다[8][9]. [10]은 말뭉치를 분석하여 한국어 술어와 보조 용언의 양상을 연구 하였다. 본 논문에서는 채팅 입력 단위를 색인어로 사용한 채팅 시스템을 제안한다.

3. 색인어 입력과 음절 출력의 채팅 시스템

3.1 전체 구성도

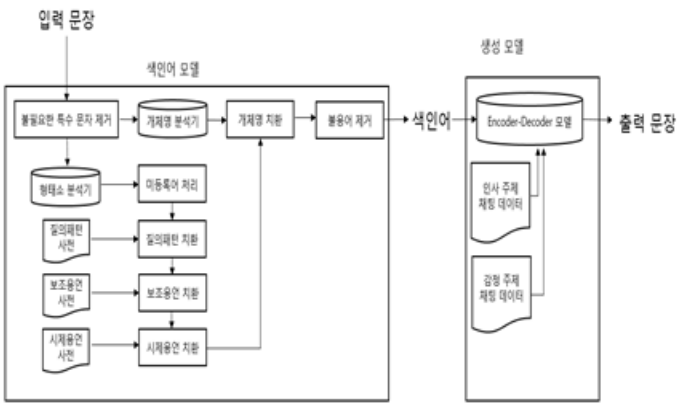


그림 1 제안 모델의 구조도

그림 1은 본 논문에서 제안 모델의 전체 구조도를 보여준다. 색인어 모델과 생성 모델로 구성된다. 그림 1과 같이 입력 문장이 색인어 모델을 통해 색인어가 추출되고, 추출된 색인어는 생성 모델의 입력으로 사용하여 출력 문장을 생성한다.

3.2 색인어 추출 방법

본 논문에서는 입력 단위를 색인어로 하는 모델을 제안한다. 색인어를 추출하기 위해서 입력 문장이 끝날 때 존재하는 '.', '?', '!'를 제외한 불필요한 특수 문자를 제거하고 형태소를 분석 한다. 형태소 분석의 결과 중 미등록어로 분석된 형태소는 앞의 3음절만 사용하여 유사한 형태의 미등록어를 동일한 모양으로 구성한다. 예를 들어 "미안해욧/NA", "미안해욤/NA"과 같은 미등록어를 "미안해/NA"로 동일한 모양으로 만든다. 그리고 개체명을 찾아 해당 어휘를 개체명 태그로 치환하여 유사한 형태를 동일한 모양으로 구성한다. 예를 들어 "임꺽정 멋있어" 라는 문장과 "홍길동 멋있어" 라는 문장을 "@PER 멋있어"로 동일한 모양으로 구성한다. 입력 데이터가 미리 구축한 질의 패턴 사전의 내용을 가질 때 해당 내용을 '@Q'로 치환하여, 다양한 질의 패턴의 내용을 하나의 형태의 색인어로 통합한다.

표 1 보조 용언의 분포 예시

양상	보조 용언	치환
가능	ㄹ 수 있	M#1
가식	ㄴ 체하 ㄴ 척하	M#2

[5]의 보조 용언의 분포를 참고하여 입력 데이터의 양상을 보고 해당 보조 용언을 표 1과 같이 치환하여 "못 본체한다", "못 본척한다"처럼 같은 양상을 가진 입력 문장들을 "못 보 M#2 다"와 같이 다른 형태의 보조 용언들을 하나의 형태의 색인어로 통합한다.

표 2 시제 용언 표

시제	시제 용언	치환
과거	았 었 있었 었었	T#1
미래	겠	T#2

입력 데이터의 시제 용언을 표 2와 같이 치환하여 "밥 먹었어", "밥 먹었었어"처럼 같은 시제를 가진 입력 문장들을 "밥 먹 T#1 어"와 같이 하나의 형태의 색인어로 통합한다. 마지막으로 체언, 용언, 부사, 독립언, 어근, 외국어, 숫자, 미등록어를 제외한 품사는 잡음으로 작용 할 수 있기 때문에 모두 제거한다. 생성된 색인어는 동일한 의미를 가진 유사한 모양의 입력 데이터를 하나의 모양으로 만들어 입력 데이터의 약간의 차이로 잘못된 출력이 생성 되는 것을 막을 수 있고, 불필요한 특수 문자와 잡음으로 작용 할 수 있는 형태소를 모두 제거하여 입력 문장의 핵심 내용만 남게 되어 좋은 입력으로 사용할 수 있다.

3.3 학습 모델

본 논문에서는 [3]에서 사용한 Attention mechanism Encoder-Decoder 모델을 사용하였다. 입력된 문장을 하나의 벡터로 표현해 문장의 의미가 추상화되어 정보가 손실되는 기존에 사용하던 Sequence-to-Sequence 모델의 문제점을 해결하기 위해 사용한 방법이다. Encoder의 모든 은닉 상태에 가중치를 반영하기 위해 주의집중 벡터를 연결한다. Decoder는 입력을 전역적으로 참고해 특정 은닉 상태에 높은 가중치를 주는 방법이다. Beam search는 [5]와 동일한 Beam search를 사용하였다. 본 논문에서 사용한 모델은 그림 2과 같다.

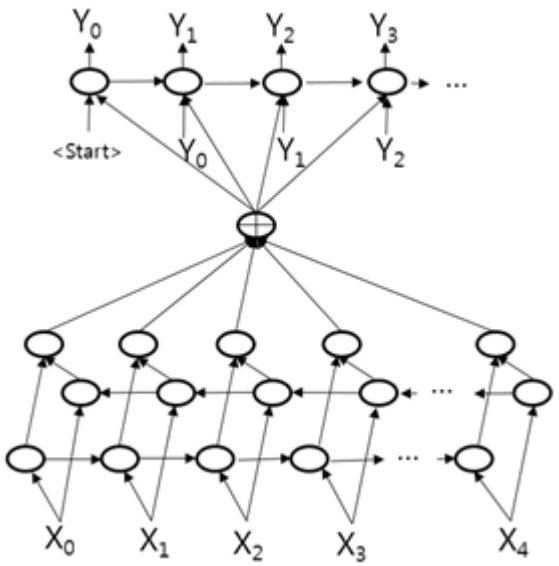


그림 2 Attention mechanism Encoder-Decoder 모델

4. 실험

4.1 실험 준비

데이터는 인사, 감정의 주제를 가지고 있으며, 데이터의 수는 n:n으로 확장하여 전체 데이터는 111,447개이고, 이 중 약 90%인 100,300개를 학습 데이터로 사용 하였고, 나머지 11,147개를 실험 데이터로 사용 하였다. 이 중 감정 데이터는 41,899개, 인사 데이터는 69,548개가 있다. 입력 단위는 형태소 단위와 색인어 단위를 사용하고, 출력 단위는 음절 단위와 의사 형태소 단위를 사용하였다. 출력 단위를 형태소로 사용하면 결합이 복잡하여 소리를 유지하고 최소한의 의미를 가져 분리 및 결합이 간편한 의사 형태소를 사용하였다. 예를 들어

“기뻐했다”를 형태소 분석 하면, “기뻐하 았 다”의 결과를 보이지만, 의사 형태소 분석을 하면 “기뻐 했다”의 결과를 보여 결합이 간편하다. 본 논문에서 사용한 의사 형태소는 [5]의 의사 형태소 분석 방법을 사용하였다. 자동 평가 방법으로 BLEU[11], ROUSE[12]를 사용하였고, 추가로 다른 평가 방법으로 정성 평가를 진행하였다. 정성 평가는 문법 점수와 적합성 점수로 나누어 진행하였다. 문법 점수는 응답이 문법에 맞으면 1점, 아니면 0점을 부여하였다. 적합성 점수는 출력 결과가 전혀 관련이 없으면 0점, 입력 문장의 특정 문맥에 적합하면 1점, 입력 문장의 문맥에 완전하게 적합하면 2점을 부여하는 방법을 사용했고, 사람이 구축한 50개의 질의로 정성 평가를 진행하였다.

4.2 음절 단위 출력 채팅 시스템 평가

형태소 입력 단위와 색인어 입력 단위를 이용해 음절 단위 출력 채팅 시스템을 평가하였다. 본 논문에서 사용하는 평가 방법뿐만 아니라 실제 출력 결과 표본을 보여주고, 이를 분석하였다.

표 3 음절 단위 출력 ROUGE , BLEU 점수 비교

입력 단위	ROUGE-L	ROUGE-1	ROUGE-2	BLEU
형태소	0.1160	0.1263	0.0400	0.1941
색인어	0.1703	0.2137	0.0635	0.2958

표 3은 음절 단위 출력에 따른 자동 평가 점수를 비교한 것이다. 색인어 단위 입력을 사용한 채팅 시스템이 ROUGE, BLEU 평가에서 모두 높은 성능을 보여주었다.

표 4 형태소 단위 입력 음절 단위 출력 결과 표본

입 력 : 빠이빠이~~ 나 이제 가요~~ 형태소: 빠이 빠이 ~ ~ 나 이제 가요 ~ ~ 출 력 : 잘 가 ~
입 력 : 너 울보야? 왜 울어 형태소: 너 울보 이 야 ? 왜 울 어 출 력 : 내 <sp> 짓 정 이 <sp> 그 리 <sp> 무 섭 소 ?
입 력 : 흑흑.. 너무 아파 형태소: 흑흑.. 너무 아프 아 출 력 : 그 렇 담 . . . <sp> 한 <sp> 밉 <sp> 나 솔 <sp> 주 시 구 옷 ~
입 력 : 아.. 울고싶어.. 시험망쳤어 형태소: 아 . . 울 고 싶 어 . . 시험 망 치 었 어 출 력 : 그 러 은 <sp> 돼 <sp> 먹 으 거 <sp> 다 효
입 력 : 나는 너가 너무 싫다. 형태소: 나 는 너 가 너무 싫 다 . 출 력 : 나 랑 <sp> 같 이 <sp> 좋 아 주 다 ~
입 력 : 야 욕하지마라.. 형태소: 야 욕 하 지 말 아라 . . 출 력 : 뭐 래 니 ?

표 4는 형태소 단위 입력 음절 단위 출력 결과의 표본이다. 이를 보면 일부 답변은 올바른 답변을 했지만, 음절 출력이라 말이 문법에 맞지 않아 어색한 경우가 있고, 전혀 알 수 없는 말을 하는 경우도 있다.

표 5 색인어 단위 입력 음절 단위 출력 결과 표본

입 력 : 빠이빠이~~ 나 이제 가요~~ 색인어 : 빠이 빠이 나 이제 가요 출 력 : 그 래 ~ <sp> 조 심 히 <sp> 가 ~
입 력 : 너 울보야? 왜 울어 색인어 : 너 울보 이 왜 울 출 력 : 내 가 <sp> 우 는 <sp> 걸 로 <sp> 보 여 ?
입 력 : 흑흑.. 너무 아파 색인어: 흑흑. 너무 아프 출 력 : 그 러 냐 ?
입 력 : 아.. 울고싶어.. 시험망쳤어 색인어: 아 울 싶 시험 망 치 T#1 출 력 : 나 도 <sp> 잘 <sp> 모 르 겠 는 데 <sp> 눈 물 이 <sp> 나
입 력 : 나는 너가 너무 싫다. 색인어: 나 너 너무 싫 출 력 : 싫 으 면 <sp> 시 집 가 시 오 .
입 력 : 야 욕하지마라.. 색인어: 야 욕 말 출 력 : 미 안 하 니 <sp> 방 법 하 지 는 <sp> 마 시 오

표 5는 색인어 단위 입력 음절 단위 출력 결과 표본이다. 이를 보면 형태소 단위를 입력으로 사용한 것 보다 문법 오류가 많이 보완되어 전혀 알 수 없는 말을 하는 경우가 적다. 그리고 많은 부분에서 적합한 응답을 하는 것을 볼 수 있다.

표 6 형태소 단위 입력 음절 단위 출력 정성 평가 결과

	문법 점수	적합성 점수
Human1	0.74	0.58
Human2	0.74	0.74
Human3	0.72	0.7
Human4	0.7	0.82
평균	0.725	0.71

표 7 색인어 단위 입력 음절 단위 출력 정성 평가 결과

	문법 점수	적합성 점수
Human1	0.96	1.06
Human2	0.96	1.16
Human3	0.96	1.22
Human4	0.96	1.06
평균	0.96	1.125

표 6은 형태소 단위 입력 음절 단위 출력 정성 평가 결과이고, 표 7은 색인어 단위 입력 음절 단위 출력 정성

평가 결과이다. 문법 점수는 색인어 입력 단위를 사용했을 때 크게 좋아지고, 대부분 문장이 문법에 맞게 생성된 것을 확인할 수 있다. 그리고 적합성 점수도 크게 향상되었다. 음절 단위 출력을 사용할 때 색인어 단위 입력을 사용하는 것이 적합하다고 판단된다.

4.3 의사 형태소 단위 출력 채팅 시스템 평가

형태소 입력 단위와 색인어 입력 단위를 이용해 의사 형태소 단위 출력 채팅 시스템을 평가하였다. 본 논문에서 사용하는 평가 방법뿐만 아니라 실제 출력 결과 표본을 보여주고, 이를 분석하였다.

표 8 의사 형태소 단위 출력 ROUGE, BLEU 점수 비교

입력 단위	ROUGE-L	ROUGE-1	ROUGE-2	BLEU
형태소	0.1096	0.1308	0.0070	0.2140
색인어	0.1457	0.1446	0.0198	0.2587

표 8은 의사 형태소 단위 출력에 따른 자동 평가 점수를 비교한 것이다. 색인어 단위 입력을 사용한 채팅 시스템이 ROUGE, BLEU 평가에서 모두 높은 성능을 보여주었다.

표 9 형태소 단위 입력 의사 형태소 단위 출력 결과 표본

입 력 : 빠이빠이~~ 나 이제 가요~~ 형태소: 빠이 빠이 ~ ~ 나 이제 가요 ~ ~ 출 력 : 그래 <sp> 그럼 <sp> 또 <sp> 봐 ~ <sp> 잘가 ~
입 력 : 너 울보야? 왜 울어 형태소: 너 울보 이 야 ? 왜 울 어 출 력 : 내 가 ? <sp> 내 가 <sp> 어디 가 <sp> 무서워 ? <sp>
입 력 : 흑흑.. 너무 아파 형태소: 흑흑.. 너무 아프 아 출 력 : 불쌍 한것
입 력 : 아.. 울고싶어.. 시험망쳤어 형태소: 아 . . 울 고 싶 어 . . 시험 망 치 었 어 출 력 : 내 가 <sp> 울꺼 <sp> 같 냐 <sp> 우헤헤
입 력 : 나는 너가 너무 싫다. 형태소: 나 는 너 가 너무 싫 다 . 출 력 : 나 도 <sp> 미우 니 <sp> 무효 !
입 력 : 야 욕하지마라.. 형태소: 야 욕 하 지 말 아라 . . 출 력 : 용서 해 <sp> 드릴께요^^

표 9는 형태소 단위 입력 의사형태소 단위 출력 결과의 표본이다. 의사 형태소를 이용해 출력을 생성하여 문법적 오류가 많이 해결된 것을 볼 수 있다. 출력 문장이 입력 문장에 적합한 응답이 생성되는 것을 볼 수 있으나, 일부 출력 문장은 입력 문장의 출력으로 사용하기에는 아쉬운 문장이 있다.

표 10 색인어 단위 입력 의사 형태소 단위 출력 결과 표본

입 력 : 빠이빠이~~ 나 이제 가요~~ 색인어: 빠이 빠이 나 이제 가요 출 력 : 그래 <sp> 다음 에 <sp> 또 <sp> 놀 자 ~
입 력 : 너 울보야? 왜 울어 색인어: 너 울보 이 왜 울 출 력 : 눈 에 <sp> 뭐 <sp> 들어갔 어 .
입 력 : 흑흑.. 너무 아파 색인어: 흑흑. 너무 아프 출 력 : 불쌍 한것
입 력 : 아.. 울고싶어.. 시험망쳤어 색인어: 아 울 싶 시험 망 치 T#1 출 력 : 나 도 <sp> 마찬가지 야 !
입 력 : 나는 너가 너무 싫다. 색인어: 나 너 너무 싫 출 력 : 나 도 <sp> 마찬가지 야 !
입 력 : 야 욕하지마라.. 색인어: 야 욕 말 출 력 : 앞 으로 <sp> 그러 지 <sp> 마세요^^

표 10은 색인어 단위 입력 의사 형태소 단위 출력 결과의 표본이다. 형태소 단위 입력 의사형태소 단위 출력 결과의 표본과 큰 차이를 보이지 않는다.

표 11 형태소 단위 입력 의사 형태소 단위 출력 정성 평가 결과

	문법 점수	적합성 점수
Human1	0.9	0.7
Human2	0.92	0.88
Human3	0.96	0.82
Human4	0.94	0.8
평균	0.93	0.8

표 12 색인어 단위 입력 의사 형태소 단위 출력 정성 평가 결과

	문법 점수	적합성 점수
Human1	0.92	0.78
Human2	0.92	0.88
Human3	0.96	1.04
Human4	0.98	1.06
평균	0.945	0.94

표 11은 형태소 단위 입력 의사 형태소 단위 출력 정성 평가 결과이고, 표 12는 색인어 단위 입력 의사 형태소 단위 출력 정성 평가 결과이다. 문법 점수와 적합성 점수 모두 색인어 단위 입력을 사용했을 때 더 좋은 결과를 보였다.

5. 결론

본 논문에서는 효과적인 입력 단위인 색인어를 제안하였다. 형태소 입력 단위와 비교 실험하여 음절 단위 출력인 경우와 의사 형태소 단위 출력 모두 색인어를 사용했을 때 자동 평가 점수와 정성 평가 문법 점수, 적합성 점수에서 높은 평가를 받았다. 특히 음절 단위 출력을 사용한 경우 문법적 오류가 많이 보완되고 응답의 내용에서도 많은 차이를 보였다. 색인어를 입력 단위로 사용했을 때, 의사 형태소 단위 출력 방법 보다는 음절 단위 출력 방법이 자동 평가 점수와 정성 평가 문법 점수, 적합성 점수에서 더 높은 점수가 나왔다. 색인어를 입력으로 사용했을 때, 음절 출력 방법에서 나오는 문법적 오류를 많이 보완시켜 의사 형태소 출력 방법 보다 좋은 결과가 나타나는 것으로 보인다.

감사의 글

감사의 글: 이 논문은 엔씨소프트 산학협력과제의 지원을 받아 수행된 연구임. 또한 부분적으로 2017년도 정부(과학기술정보통신부)의 재원으로 정보통신기술진흥센터의 지원을 받아 수행된 연구임. (R0126-15-1117, 언어학습을 위한 자유발화형 음성대화처리 원천기술 개발)

참고문헌

[1] 김종환, 장두성, 김학수, “복합 자지 정보를 이용한 통계적 한국어 채팅 문장 생성.”, 인지과학, 제20권, 제4호, pp. 421-437, 2009

[2] 전원표, 송영길, 김학수, “채팅 시스템 구현을 위한 3단계 문장 검색 방법”, 한국마린엔지니어링학회지, 제37권, 제2호, pp. 205-212, 2013

[3] Bahdanau, Dzmitry, Kyunghyun Cho, and Yoshua Bengio. "Neural machine translation by jointly learning to align and translate." arXiv preprint arXiv:1409.0473, 2014

[4] Vinyals, Oriol, and Quoc Le. “A neural conversational model.” arXiv preprint arXiv:1506.05869, 2015

[5] 김시형, 김학수, “의사 형태소 단위 채팅 시스템”, 제 28회 한글 및 한국어 정보처리 학술대회 논문집, pp. 263-267, 2016

[6] Li, Jiwei, et al. "A diversity-promoting objective function for neural conversation models." arXiv preprint arXiv:1510.03055 (2015).

[7] Qit, Minghui, et al. "AliMe Chat: A Sequence to Sequence and Rerank based Chatbot Engine." Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers). Vol. 2. 2017

[8] Li, Jiwei, et al. "A persona-based neural conversation model." arXiv preprint arXiv:1603.06155, 2016

[9] 이현구, 김학수, “주의집중 및 복사 작용을 가진 Sequence-to-Sequence 순환신경망을 이용한 제목 생성 모델”, 정보과학회논문지, pp 674-679, 2017

[10] 안동언, “corpus를 기반으로 하는 한국어 술어의 양상 생성”, 한국과학기술원: 전산학과 박사 학위 논문, 1995

[11] Papineni, Kishore, et al. "BLEU: a method for automatic evaluation of machine translation." Proceedings of the 40th annual meeting on association for computational linguistics. Association for Computational Linguistics, 2002.

[12] Lin, Chin-Yew. "Rouge: A package for automatic evaluation of summaries." Text summarization branches out: Proceedings of the ACL-04 workshop. Vol. 8. 2004.

● 구두발표 7: 정보추출

Bidirectional LSTM-CRF 앙상블을 이용한 공간 개체 추출

민태홍○, 이재성
충북대학교, 충북대학교
mintaehong@cbnu.ac.kr, jasonlee@cbnu.ac.kr

Spatial Entities Extraction using Bidirectional LSTM-CRF Ensemble

Tae Hong Min○, Jae Sung Lee
Chungbuk National University

요 약

공간 정보 추출은 대량의 텍스트 문서에서 자연어로 표현된 공간 관련 개체 및 관계를 추출하는 것으로 질의응답 시스템, 챗봇 시스템, 네비게이션 시스템 등에서 활용될 수 있다. 본 연구는 한국어에 나타나 있는 공간 개체들을 효과적으로 추출하기 위한 앙상블 기법이 적용된 Bidirectional LSTM-CRF 모델을 소개한다. 한국어 공간 정보 말뭉치를 이용하여 실험한 결과, 기존 모델보다 매크로 평균이 향상되어 전반적인 공간 관계 추출에 유용할 것으로 기대한다.

주제어: 공간 정보, 정보 추출, 딥러닝, 앙상블

1. 서론

대량의 텍스트가 컴퓨터 시스템에 기록되고 있고, 특히 웹이나 모바일 시스템을 통해 수집되는 텍스트의 양이 급속히 증가함에 따라, 이를 분석하거나 유의미한 정보를 추출하는 기술 또한 발전해 왔다. 공간 정보 추출은 정보 추출의 한 종류로 공간 개체와 그들 사이의 관계를 연결시켜주는 공간 관계를 추출하는 기술이다. 이는 질의응답 시스템, 챗봇 시스템, 네비게이션 시스템 등 공간 정보 추출과 공간 추론을 해야 하는 시스템에 활용될 수 있다.

공간 정보 표기법은 국제 표준인 ISO-Space[1]로 제정되었다. 이 표준에 따르면 공간 정보는 공간 정보 개체와 관계로 이루어져 있으며, 개체는 Place(장소), Path(경로), Spatial Entity(공간 안에 존재하는 개체), Spatial Signal(정적인 관계 어휘), Motion(동적인 관계 어휘), Motion Signal(이동을 설명하는 어휘), Measure(개체 측정치)로 총 7개 이며, 관계를 표현하는 링크는 Qualitative Spatial Link(개체간의 상대적인 위치), Orientation Link(개체간의 위상 정보), Movement Link(개체의 움직임 혹은 상태), Measurement Link(개체 측정치의 관계) 총 4개의 링크로 이루어져 있다. 국제적으로 이 표준에 기반을 두어 공간 정보를 추출하는 연구들이 진행되어 왔다[2,3]. 국내의 공간 정보 연구는 영어권에서 연구된 내용을 한국어의 특성에 맞게 변형을 하고 보완한 연구로 진행이 되었다[4].

본 논문에서는 공간 정보 개체 추출의 성능 향상을 위하여 앙상블 기법이 적용된 bidirectional LSTM-CRF을 제안한다. 또 이 모델을 다양한 단어 임베딩에 적용하여 평가하고 그 결과를 제시한다.

2. 모델 소개

bidirectional LSTM-CRF 모델은 개체명 인식(Named Entity Recognition) 등의 연구에서 매우 좋은 성능을 보인다[5]. 공간 정보 추출도 sequence labeling 문제로 볼 수 있으므로 본 논문에서는 해당 모델을 사용하였다.

2.1 Bidirectional LSTM-CRF

일반적으로 사용되는 bidirectional LSTM-CRF의 모델을 공간 정보 추출에 적용한 구조는 그림 1과 같다[5]. 그림에 나타난 bidirectional LSTM-CRF는 각각 단어 표상을 입력으로 받는다. 이를 오른쪽 방향으로 분석하는 R층, 왼쪽 방향으로 분석하는 L층 총 2개의 LSTM Layer을 통해 나온 데이터를 C로서 합친다. 이 C를 CRF(Conditional Random Field)를 이용하여 각각 태그의 점수를 벡터로 계산하고, 이를 최대로 하는 태그를 선정한다.

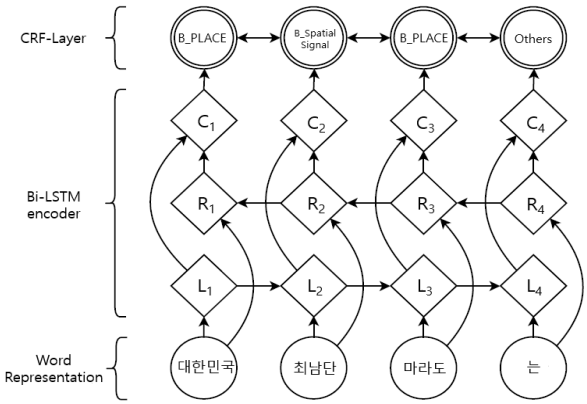


그림 1. bidirectional LSTM-CRF 모델

2.2 공간 정보 추출을 위한 단어 표상 확장

단어 표상(word representation)이란 각각 단어에 관하여 정보를 표기하는 방법이다. 본 연구는 워드 임베딩 벡터(100), 형태소 정보(46), 개체명 인식(35), 기분석 사전(15) 총 196차원의 벡터로 단어를 표현하였다.

(1) 워드 임베딩 벡터(word embedding vector)

워드 임베딩 벡터는 단어 의미 자체를 특정 차원인

벡터로 표현하는 것을 의미한다. 본 실험은 Word2Vec의 CBOW(Continuous Bag of Words) 및 Skip-gram[6]과 Stanford의 GloVe[7], Facebook의 fastText[8] 총 4가지 워드 임베딩 벡터 모델을 이용하여 100차원의 워드 벡터를 만들어 비교 실험하였다.

(2) 형태소 정보

형태소 정보는 [9]에서 규정한 45개와 문장 시작, 띄어쓰기, 문장 끝(<SOS>, <BLN>, <EOS>)를 1개의 태그로 총 46개를 one-hot 벡터로 표현하였다.

(3) 개체명 인식(Named Entity Recognition)

개체명 인식의 정보는 [10]에서 규정한 대분류들을 사용하였으며, LC, AF, QT는 소분류 까지 사용하여 총 35개의 태그를 one-hot 벡터로 표현하였다.

(4) 기분석 사전

기분석 사전이란 학습데이터에서 동일한 태그로 주석된 어휘에 대하여, 해당 정보를 주는 것이다. 본 연구에서는 동일한 태그로 3번 이상 주석 된 어휘를 대상으로 공간 정보 개체의 개수인 15차원 one-hot 벡터로 표현하였다.

2.3 앙상블 모델

신경망 모델에서 성능을 향상시키기 위한 연구 중의 하나로 앙상블 알고리즘을 적용한 연구들이 있다[11]. 기본적인 앙상블 모델의 구조는 그림 2와 같다.

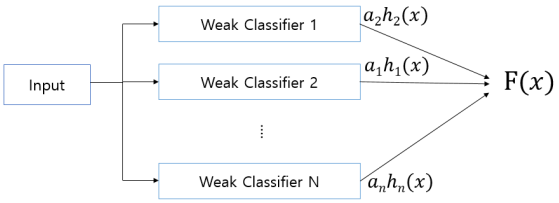


그림 2. 앙상블 모델의 구조

앙상블 기법은 단일 모델을 여러 개 학습시킨 후 그 모델들의 결과 $h(x)$에 가중치 a를 곱한 값들을 더하여 결과 $F(x)$를 도출한다. 이를 표현하는 수식은 다음과 같다.

$$F(x) = \sum_{i=0}^{n} (a_i h_i(x))$$

동일한 모델이라도 초기의 신경망 가중치를 랜덤으로 지정하는 점, dropout 기법 등 랜덤 수치가 적용되는 점이 있기에 학습되는 결과가 조금씩 다르다. 이를 통해 동일한 모델을 여러 개 (현 실험에서는 5개) 학습시킨 후 이들 값을 입력받아, 최종적인 분류를 하는 모델을 만들어 성능을 향상시켰다. 앙상블 기법을 적용시킨 bidirectional LSTM-CRF모델은 그림 3과 같다.

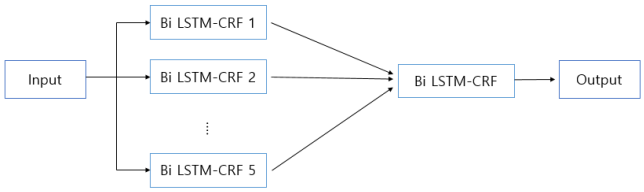


그림 3. 앙상블 기법을 적용시킨 최종 모델

3. 실험 및 평가

본 실험은 한국어 공간 정보 주석 말뭉치 v1.0을 이용하였으며, 각각의 개체 개수는 아래 표 1과 같다[12].

표 1. 한국어 공간 정보 주석 말뭉치 개체 개수

개체	개수	개체	개수
문장	1654	Spatial Entity	390
Place	5049	Spatial Signal	1171
Path	275	Motion	261
Measure	235	Motion Signal	245

실험은 5-fold test로 진행하였으며, 한국어 앙상블 기법을 적용한 모델들과 적용하지 않은 단일 모델들로 나누고, 각 모델에 대해 4가지의 워드 임베딩 모델을 적용하여 실험하였다. 그 결과는 표 2와 같다.

표 2. 공간 정보 개체 추출 성능 결과(%)

모델	워드 임베딩	정확률	재현율	F1
단일	CBOW	79.9	85.1	82.4
	Skip-gram	79.4	85.2	82.2
	GloVe	79.2	85.0	82.0
	fastText	81.7	86.7	84.1
앙상블	CBOW	81.3	87.3	84.1
	Skip-gram	81.3	86.7	83.9
	GloVe	81.2	88.2	84.5
	fastText	**82.8**	**88.4**	**85.5**

표 2에서 보듯이 fastText를 워드 벡터로 사용하였을 경우 성능이 단일 모델이나 앙상블 모델에서 가장 좋게 나왔으며, 앙상블 기법을 적용하였을 때 성능이 더 우수하다.

한국어 공간 정보 개체 추출 시스템으로서 기존 연구 중 가장 성능이 좋은 것은 CRF 모델을 이용한 것이다[4]. 이 모델에서는 형태소 원형, 형태소 품사, 어절 띄어쓰기 정보, 형태소의 의미 부류, 개체명 인식 정보, 의존 구문 레이블, 의존 구문 head 레이블, 의존 구문 head 레이블의 형태소, 워드 클러스터 정보, 의존 구문 head의 워드 클러스터 정보, 총 10가지 자질을 각각 개체마다 조금씩 다르게 적용하여 학습을 진행하였다. 그리고 성능을 높이기 위해 앙상블 모델을 사용하여 각각

결과를 합친 후, 유효한 태그를 분별하기 위해 태그 벡터를 사용하였다. 기존의 CRF 기반 공간정보 추출 모델과 본 연구에서 최고 성능을 보인 모델의 성능 비교는 표 3와 같다.

표 3. 기존 모델[4]과 제안 모델 비교(%)

개체	정확률		재현율		F1	
	기존 모델	제안 모델	기존 모델	제안 모델	기존 모델	제안 모델
Place	96.1	90.0	95.8	91.0	96.0	90.5
Path	55.2	61.3	53.9	75.2	54.5	67.2
S.Entity	32.6	42.5	44.4	68.8	37.6	52.3
Motion	54.4	59.4	70.9	74.1	61.6	65.8
M.Signal	55.6	49.2	69.4	74.2	61.7	58.2
S.Signal	89.2	80.1	83.6	85.0	86.3	82.4
Measure	95.1	90.4	90.6	95.0	92.8	92.5
마이크로 평균	**86.1**	**82.8**	**88.0**	**88.4**	**87.0**	**85.5**
매크로 평균	**68.3**	**67.6**	**72.7**	**80.5**	**70.1**	**72.7**

표 3을 보면 Path, Spatial Entity, Motion의 추출 성능이 기존 모델보다 오른 것을 볼 수 있다. 특히 Path와 Spatial Entity는 각각 12.7%포인트, 14.7%포인트만큼 큰 폭으로 추출 성능이 향상되었다. Spatial Entity는 이동하는 개체 혹은 이동의 가능성이 있는 개체에 대한 태그이기에 태그들 간의 관계를 형성하는 경우에만 추출한다. 그렇기에 Spatial Entity의 성능이 오른 점은 차후 공간 정보 관계 추출에 유용할 것으로 기대한다.

전체적인 성능으로는 제안 모델의 F1값은 마이크로 평균은 85.5%이며, 매크로 평균은 72.7%이다. 기존 모델과 비교해 볼 때, 마이크로 평균은 1.5%포인트 하락하였으며, 매크로 평균은 2.6%포인트 상승하였다. 비록 마이크로 평균은 하락하였지만, 매크로 평균이 상승하여, 학습 데이터의 데이터 편중의 문제를 완화하였음을 보여준다.(한국어 공간 정보 말뭉치 특성상 Place 태그의 빈도가 다른 태그들의 빈도보다 높아 데이터가 편중되어 있다. 그 결과 Place 태그 추출의 성능은 좋게 나오는 반면 다른 태그 추출 성능은 낮은 것을 볼 수 있다. 특히 Place, Path, Spatial Entity는 주로 명사이기에 이를 구분하는 것이 어렵다는 점이 [4]의 성능 저하의 원인 중 하나다.) 본 연구에서 Path, Spatial Entity, Motion의 성능이 전반적으로 상승하여 데이터 편중 문제를 어느 정도 보완한 것으로 판단할 수 있다.

4. 결론 및 향후 연구계획

본 연구는 앙상블 기법이 적용된 bidirectional LSTM-CRF을 사용하여 공간정보를 추출하는 방법을 제안하였다. 다양한 단어 임베딩을 사용하여 실험한 결과, fastText 임베딩을 이용하고, 앙상블 기법을 사용한 모델이 가장 우수하였다. 기존 CRF 모델의 성능과 비교하여 보면, 마이크로 평균은 하락하였지만, 매크로 평균이 상승하여, 학습데이터의 데이터 편중의 문제를 완화하였다고 볼 수 있다.

향후 연구로는 보다 다양한 파라미터를 사용하여 앙상블 모델을 확장하고 이를 딥러닝 기반의 공간 관계 추출 시스템과 통합하여 전체 공간 정보 추출의 성능을 향상시킬 계획이다.

감사의 글

본 연구는 미래창조과학부 및 정보통신기술진흥센터의 정보통신·방송 연구개발 사업의 일환으로 하였음. [2013-0-00131,(엑소브레인-1세부) 휴먼 지식증강 서비스를 위한 지능진화형 WiseQA 플랫폼 기술 개발]

참고문헌

[1] ISO 24617-7:2014, language resource management - part 7: Spatial information (ISOspace).

[2] Pustejovsky, James, Parisa Kordjamshidi, Marie-Francine Moens, Aaron Levine, Seth Dworman, and Zachary Yocum, “SemEval-2015 task 8: SpaceEval,” SemEval 2015, 2015.

[3] Kolomiyets, Oleksandr, Parisa Kordjamshidi, Steven Bethard, and Marie-Francine Moens, “SemEval-2013 task 3: spatial role labeling,” SemEval 2013, 2013.

[4] Kim, Bogyum and Jae Sung Lee. “Extracting spatial entities and relations in Korean text,” COLING. 2016.

[5] Lample, Guillaume, Miguel Ballesteros, Sandeep Subramanian, Kazuya Kawakami, and Chris Dyer. “Neural architectures for named entity recognition,” arXiv preprint arXiv:1603.01360, 2016.

[6] Mikolov, Tomas, Kai Chen, Greg Corrado, and Jeffrey Dean, “Efficient estimation of word representations in vector space,” arXiv preprint arXiv:1301.3781, 2013.

[7] “GloVe: Global Vectors for Word Representation,” https://nlp.stanford.edu/projects/glove

[8] Bojanowski, Piotr, Edouard Grave, and Armand Joulin, Tomas Mikolov, “fastText,” https://research.fb.com/fasttext/

[9] 국립국어원, 21세기 세종계획 최종성과물(2011년 12월 수정판), 2011.

[10] TTAK0.KO-10.0852:2015, 개체명 태그세트 및 태깅 말뭉치.

[11] 권순재, 허윤석, 이건철, 임지수, 최호정, 서정연, “가중 투표 기반의 앙상블 기법을 이용한 한국어 개체명 인식기”, 제28회 한글 및 한국어 정보처리 학술대회 논문집, pp.333-336, 2016.

[12] 충북대학교 언어지식공학 연구실, “한국어 공간 정보 주석 가이드라인”, 2016.

다중-어의 단어 임베딩을 적용한 CNN 기반 원격 지도 학습 관계 추출 모델

남상하○, 한기종, 김은경, 권성구, 정유성, 최기선
한국과학기술원
{nam.sangha, han0ah, kekeeo, fanafa, wjd1004109, kschoi}@kaist.ac.kr

CNN-based Distant Supervision Relation Extraction Model with Multi-sense Word Embedding

Sangha Nam○, Kijong Han, Eun-Kyung Kim, Key-Sun Choi
KAIST

요 약

원격 지도 학습은 자동으로 매우 큰 코퍼스와 지식베이스 간의 주석 데이터를 생성하여 기계 학습에 필요한 학습 데이터를 사람의 손을 빌리지 않고 저렴한 비용으로 만들 수 있어, 많은 연구들이 관계 추출 문제를 해결하기 위해 원격 지도 학습 방법을 적용하고 있다. 그러나 기존 연구들에서는 모델 학습의 입력으로 사용되는 단어 임베딩에서 단어의 동형이의어 성질을 반영하지 못한다는 단점이 있다. 때문에 서로 다른 의미를 가진 동형이의어가 하나의 임베딩 값을 가지다 보니, 단어의 의미를 정확히 파악하지 못한 채 관계 추출 모델을 학습한다고 볼 수 있다. 본 논문에서는 원격 지도 학습 기반 관계 추출 모델에 다중-어의 단어 임베딩을 적용한 모델을 제안한다. 다중-어의 단어 임베딩 학습을 위해 어의 중의성 해소 모듈을 활용하였으며, 관계 추출 모델은 문장 내 주요 특징을 효율적으로 파악하는 모델인 CNN과 PCNN을 활용하였다. 본 논문에서 제안하는 다중-어의 단어 임베딩 적용 관계추출 모델의 성능을 평가하기 위해 추가적으로 2가지 방식의 단어 임베딩을 학습하여 비교 평가를 수행하였고, 그 결과 어의 중의성 해소 모듈을 활용한 단어 임베딩을 활용하였을 때 관계추출 모델의 성능이 향상된 결과를 보였다.

주제어: Relation Extraction, Distant Supervision, Word Embedding, Convolution Neural Network

1. 서론

관계 추출(Relation Extraction) 이란 문장 내 등장한 두 개체(Entity) 사이의 관계(Relation)를 알아내는 작업을 일컫는다. 예를 들어, “마크 저커버그는 페이스북 설립자이다.” 라는 문장으로부터 Founder(페이스북, 마크_저커버그) 와 같은 관계를 추출하는 것이다. 최근 들어 지식베이스의 중요성이 대두되고 DBpedia, YAGO, Wikidata 등의 대규모 지식베이스 구축을 위한 연구들이 활발히 진행 중이며, 그에 따라 웹 규모 말뭉치(Web Scale Corpus)에서 지식을 추출하고자 하는 연구들이 진행중이다. 그러나 많은 연구들에서 관계 추출 시스템을 설계하기 위해 기계학습(Machine Learning) 방식을 활용하고 있기 때문에, 많은 양의 지도 학습 데이터(Supervised Learning Data)를 생성하기에는 고비용(High-cost) 문제가 발생하였고, 이를 해결하기 위해 [1]의 논문에서 원격 지도 학습(Distant Supervision) 방식을 소개하였다. 원격 지도 학습 방식은 “두 개체가 지식베이스에서 특정 관계로 연결되어 있고 이 두 개체가 함께 포함된 문장을 말뭉치에서 모두 수집하면, 수집된 문장들은 두 개체간의 특정 관계를 설명하고 있을 것이다.”라는 가정을 기반으로 한다.

그림 1은 자동으로 주석 데이터(Labeled Data)를 수집하는 원격 지도 학습 방식 예시이다. 원격 지도 학습 방식은 대용량 말뭉치, 대규모 지식베이스 간 학습 데이터를 자동으로 생성해준다는 점에서 상당히 효율적이지만, 자동 수집된 학습 데이터의 품질이 항상 좋은 것은 아니라는 문제점을 가지고 있다. 그림 1에서 보는바와 같이 ‘페이스북’과 ‘마크 저커버그’가 동시에 들어있는 문장을 수집해보면 1번 문장과 같이 실제 두 개체 간의 관계(예시: founder)를 의미하는 문장도 수집되지만, 2번 문장과 같이 두 개체가 포함되어 있을 뿐 두 개체간의 명확한 관계를 의미하지 않는 문장도 학습데이터로 수집될 수 있다.

이러한 단점을 해결하기 위해 자동 수집된 학습 데이터의 품질을 향상시키기 위한 다양한 연구들이[2, 3, 4] 소개되었으나, 관계 추출 시스템을 설계할 때 전통적인 자연언어처리 분야에서 사용하던 특징들(Feature)은 자연언어처리 도구에서 발생하는 오류로 인해 에러 전파 및 축적의 문제가 발생하였다. 그에 따라, 전통적인 특징들을 사용하지 않고 단어 임베딩(Word Embedding)과 DNN(Deep Neural Network) 기반의 관계 추출 연구들이 [5, 6] 소개되었고, 기존 연구들보다 향상된 관계 추출 성능을 보였다. 특히 [6]에서 소개된 PCNN(Piecewise max pooling Convolution Neural Network) 모델은 CNN 학습 모델을 관계 추출에 더 적합한 형태로 확장한 것으로, 입력 벡터에 ‘문장 내 주어 및 동사의 위치’를 추가시킨 것과 ‘문장 내 두 개체의 위치를 기준으로 3개의 Max pooling 연산을 수행’하도록 확장한 것이 큰 특징이

다.

그러나 기존 연구들에서는 모델 학습의 입력으로 사용되는 단어 임베딩에서 단어의 정확한 의미를 반영하지 못한 단점이 있다. 예를 들어, '부르다' 라는 단어는 '입으로 소리를 내는 것' 과 '속이 꽉 찬 느낌이 드는 것' 과 '무엇인가 퍼뜨리고 펼치는 것' 등의 여러 의미가 존재하고, 또 세부적으로는 '노래를 부르는 것', '어떤 행동을 동참하도록 유도하는 것' 등 더욱 다양한 의미로 사용된다. 따라서 한 단어에 대해 하나의 벡터값으로 학습을 진행하게 되면 동형이의어의 특성을 제대로 반영하지 못한 결과가 발생할 수 있다. 그에 따라 영어권에서는 다중-어의 단어 임베딩(Multi-sense Word Embedding)에 대한 연구들이[7, 8] 진행 중이지만, 지금까지 다중-어의 단어 임베딩을 관계 추출 모델에 적용한 결과는 발견하지 못했다.

본 논문에서는 원격 지도 학습 기반 관계 추출 모델에 다중-어의 단어 임베딩을 적용한 모델을 제시하고 그 결과에 대해 소개한다. 관계 추출 모델은 [5]에서 제시한 CNN 모델과 [6]에서 제시한 PCNN 모델 두 가지를 활용하였으며, 어의 중의성 해소 모듈을 활용한 다중-어의 단어 임베딩을 관계추출 모델의 입력으로 사용하였다. 단어 임베딩은 Word2Vec[9]의 Skip-gram 모델을 기반으로 단어와 형태소, 그리고 단어 의미 번호(word sense)를 함께 토큰화하여 학습을 진행하였으며, 본 논문에서 제안한 방식의 우수성을 입증하기 위해 비교대상으로 (1) 단어 단위 임베딩, (2) 형태소 단위 임베딩을 추가 학습하여 관계 추출 성능에 대한 비교 평가를 수행하였고 그 결과를 소개한다.

2. 관련 연구

2.1 단어 임베딩 스킵-그램 모델

스킵 그램(Skip-gram) 모델은 그림 2와 같은 구조로, 타겟 단어를 기준으로 주변에 등장할 단어의 여부를 유추하는 것으로 학습을 진행한다. Skip-gram 모델에서는 주어진 단어(w_t)와 w_t 주변에서 등장한 단어들(c)의 벡터값을 아래 목적 함수(Objective Function)를 최대화하는 방향으로 학습을 진행한다. 아래 식에서 w_t는 학습 코퍼스 내 타겟 단어를 의미하고, c_t는 w_t단어 주변에서 실제 나타난 단어를 의미한다. 그리고 c'_t는 w_t주변에 등장하지 않은 단어들 중 랜덤하게 선택한 Negative sampling을 의미한다. 즉, 타겟 단어 주변에 실제 등장한 단어들을 예측할 확률과 실제 등장하지 않은 단어들을 예측하지 않을 확률을 최대화하는 방식으로 학습이 진행된다.

$$J(\theta) = \sum_{(w_t,c_t)\in D+} \sum_{c\in c_t} logP(D=1|v(w_t),v(c)) + \sum_{(w_t,c'_t)\in D-} \sum_{c'\in c'_t} logP(D=0|v(w_t),v(c'))$$

2.2 PCNN 관계 추출 모델

Convolutional Neural Network(CNN)은 이미지 분류 및 문장 감성 분류(Sentiment Classification) 등에서 우수한 성능을 보이는 모델이다. CNN의 특징이자 장점 중 하나인 입력 데이터 내 주요 특징을 효율적으로 찾아낸다는 점에 착안하여 [5, 10]의 논문에서는 CNN을 이용한 관계 추출 모델을 제안하였다. 그 중 [5]의 논문에서는 위치 임베딩(Position Embedding) 개념을 도입하여, 문장 내 두 개체의 위치를 입력 벡터에 추가해줌으로써 관계 추출 모델의 성능 향상을 보였다. 위치 임베딩이란 문장 내 두 개체와 개체가 아닌 단어들 간의 상대적 거리를 n 차원의 벡터로 임베딩 한 것이다. 예를 들어, 그림 3에서 보는바와 같이 '설립' 이라는 단어가 '마크_저커버그' 개체로부터 3단어만큼 떨어져있고, '페이스북' 개체로부터 1단어만큼 떨어져있다. 이 상대적 거리를 n 차원의 벡터로 임베딩하고, 그 값을 모델 학습의 입력 벡터 중 일부로 사용한다.

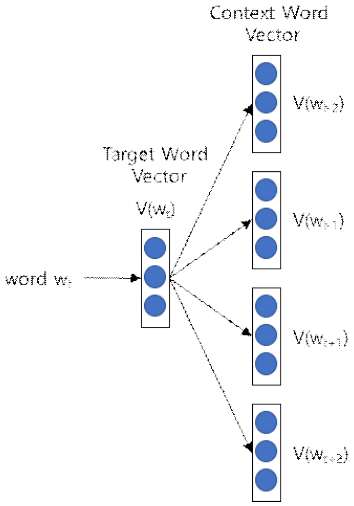


그림 2 Skip-gram 모델 구조 (Window size = 2)

그림 3 위치 임베딩의 상대적 거리 예시

PCNN(Piecewise max pooling Convolutional Neural Network)은 [6]의 논문에서 제안한 관계 추출 모델의 하나로 [10]에서 제안한 CNN 모델을 확장한 것인데, CNN에서 흔히 사용하는 Single Max Pooling Layer를 Piecewise Max Pooling Layer로 확장하였다는 것이 큰 차이점이다. CNN에서 맥스 풀링(Max Pooling)은 Activation Map 혹은 Feature Map이라 불리는 Convolution Layer의 출력 매트릭스에서 가장 큰 값 즉, 가장 중요한 특징을 추출하는 방법이다. 그러나 Single Max Pooling Layer는 은닉 층(Hidden Layer) 결과값 중 최대값 하나만을 선택함으로써 관계 추출에 필요한 특징을 세밀하게 파악하기 힘들다. PCNN은 이러한 단점을 해결하기 위해 Single Max Pooling Layer를 3덩이로 나눈 Piecewise Max Pooling Layer를 제안하였다. 이 층(Layer)의 큰 특징은 문장을 총 3덩이로 나누어서 각각 맥스 풀링을 수행한다는 점이다. 관계 추출에 사용되는 문장은 항상 두 개체를 포함하고 있기 때문에 두 개체를

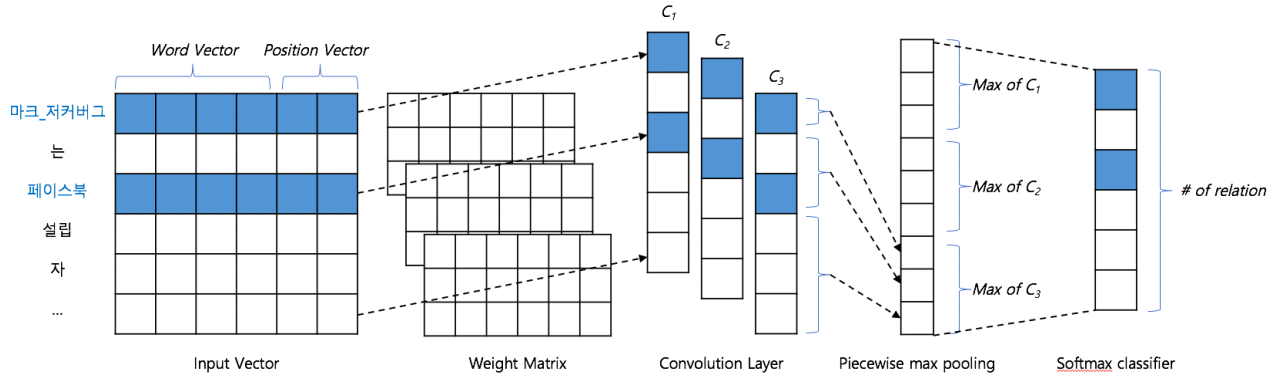


그림 4 PCNN 구조 및 예시

기준으로 문장을 총 3덩이로 나눌 수 있고 각 덩이마다 최대값을 추출하여 학습에 반영한다. PCNN의 구조는 그림 4와 같다. 입력 벡터(Input Vector)는 단어 벡터(Word Vector)와 위치 벡터(Position Vector)로 구성되며 3개의 Convolution Layer, Piecewise Max Pooling Layer, Softmax Output 단계로 전체 구조가 이루어진다.

3. 방법론

본 논문에서는 CNN 및 PCNN 기반의 관계 추출 모델을 활용하여 어의 중의성 해소(Word Sense Disambiguation) 기반 다중-어의 단어 임베딩을 적용한 관계 추출 모델을 제안한다.

3.1 단어 임베딩

단어 임베딩(Word Embedding)은 최근 자연언어 처리 분야에서 상당히 유용하게 사용되고 있다. 일반적으로는 입력 말뭉치를 토큰(Token) 단위로 분할한 다음 의미적 연관성이 높은 토큰들을 유사한 실수 벡터 값으로 생성한다. 이때 영어에서는 일반적으로 단어(Word) 단위 즉, 띄어쓰기 단위로 토큰을 생성한다. 그러나 한국어는 조사, 어미, 그리고 접미사 등 한 단어를 이루는 복수개의 구성 요소들로 인해 띄어쓰기 단위로 단어 임베딩 학습을 진행하면 그 결과가 상대적으로 영어만큼 좋지 않다. 그에 따라 한국어에서는 형태소 단위로 토큰을 구성하여 단어 임베딩을 학습하는 방식이 사용되고 있고, 이때 토큰의 구성 요소로 품사태그가 함께 사용되기도 한다. 예를 들어, '사과/Noun'과 같이 품사태그를 함께 학습할 경우 이 단어와 유사한 단어들로 '사죄/Noun', '애도/Noun', '죄송하다/Adjective' 등이 위치하게 된다. 품사태그를 함께 단어 임베딩 학습에 활용했을 때 좋은 점은 동일한 형태의 단어가 동사로 쓰였는지 명사로 쓰였는지 등에 대한 구분이 가능하다. 예를 들어, '가지'라는 단어는 식물을 뜻하는 명사로 쓰이기도 하고 소유하다는 의미의 동사 어근으로 쓰이기도 한다. 따라서 한국어에서는 단어 임베딩 생성 시 형태소 단위 그리고 품사태그를 함께 학습에 이용하는 것이 효율적이다.

그러나 위 방식의 단어 임베딩은 단어의 실제 의미를 반영하지 못한다는 단점이 있고, 이는 한국어뿐만 아니라 영어에서도 마찬가지이다. 예를 들어, 'apple'이라는 단어는 과일로 쓰이기도 하고 회사로 쓰이기도 한다. 단어 임베딩은 앞서 2장에서 언급하였듯이 주변 단어가 어떤 것들로 구성되느냐에 따라 학습되는데, '과일 apple' 주변에 등장하는 단어와 '회사 apple' 주변에 등장하는 단어를 모두 'apple' 단어 주변에 등장하는 단어로 뭉쳐서 학습을 진행하기 때문에 결국 'apple'은 하나의 n 차원 실수 벡터값을 가지게 되고 여러 의미를 구분 지을 수 없는 단어 임베딩이 학습된다. 한국어에서도 마찬가지로 '사과'를 '과일 사과'와 '사죄의 사과'를 섞어 학습하게 된다. 그에 따라 아래 식과 같은 삼각 부등식 문제(Triangle Inequality Problem)가 발생할 수 있다[7].

$$distance(a,c) \leq distance(a,b) + distance(b,c)$$

예를 들어 pollen(꽃가루)와 refinery(정제 공장)간의 유사도(distance)가 pollen(a)과 plant(b)와의 거리 그리고 refinery(c)와 plant(b)와의 유사도 합 보다 작아지게 되는 문제가 발생한다. 즉, plant라는 동형이의어를 중심으로 'pollen'과 'refinery'라는 두 단어의 유사도가 실제 의미적 연관성보다 가까워지는 문제이다. [7]

이러한 문제를 해결하기 위해 단어의 의미를 여러 개의 군집으로 클러스터링 하여 분할하는 방법 [7], 워드넷(WordNet)을 기반으로 단어 의미를 구분하는 방법 [8], 사전 뜻풀이를 학습에 활용하여 단어 의미를 구분하는 방법 [11] 등이 발표되었다. 본 논문에서는 단어의 의미를 구분 짓는 어의 중의성 해소(WSD) 모듈의 결과를 활용하여 단어 임베딩 학습을 진행하였고, 어의 중의성 해소 모듈은 본 연구팀에서 연구 개발한 모듈을 활용하였다.

아즐리 고등학교를 다닐 당시 그는 서양고전학 (classics) 과목에서 우수한 성적을 거두었다. 이후 3학년 때 필립스 엑세터 아카데미로 학교를 옮긴 그는 과학(수학, 천문학 및 물리학)과 서양고전 연구(Classical

그림 5 한국어 위키피디아 개체 태깅 예시. '아즐리 고등학교', '서양고전학', '필립스 엑세터 아카데미'

추가적으로, 관계 추출에 적합한 단어 임베딩을 학습하기 위해 여러 단어로 된 개체(Multi-word Entity)는

하나의 토큰으로 묶는 개체-반영 단어 임베딩 학습을 진행하였다. 그림 4에서 보는 바와 같이 '마크 저커버그'는 두 단어로 구성된 하나의 개체이다. 한 개체는 여러 단어로 구성되었다 하더라도 하나의 단어 임베딩 값을 가지도록 학습하는 것이 단어 임베딩과 관계추출 모델 설계에 적합하다. 만약, 개체에 대한 구분 없이 모든 토큰을 형태소 단위로 학습을 진행하게 되면 여러개의 단어로 구성된 개체에 대한 임베딩은 얻어낼 수 없다. 그 결과, 예를 들어, '코피 아난'과 유사한 단어로는 '국제 연합 사무총장', '반기문', '유엔 사무총장', '연방 준비 제도' 등이 위치한다. 말뭉치 내 개체 판별은 한국어 위키피디아에 태깅되어 있는 개체들을 사용하였으며, 예시는 그림 5와 같다. 파란색으로 표기되는 이 개체들은 위키피디아 내용 작성자들이 손으로 태깅한 것으로 정확도가 매우 높다고 할 수 있다.

3.2 관계 추출 모델

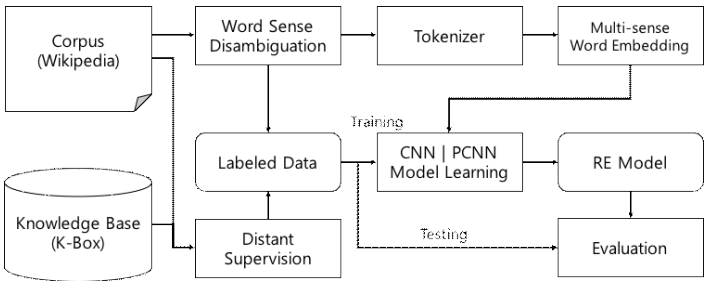


그림 6 관계 추출 시스템 구성도

본 논문의 관계 추출 시스템 구성도는 그림 6과 같고, 크게 단어 임베딩 학습과 원격 지도 학습 관계 추출 모델 학습 및 평가 부분으로 구성된다. 먼저 코퍼스를 입력으로 받아 어의 중의성 해소(Word Sense Disambiguation)를 수행한다. 어의 중의성 해소 단계는 본 연구팀의 비-지도 학습 방식 기반 MRF WSD 모듈을 활용하였으며, 이 모듈은 코어넷[13]의 개념체계를 기준으로 중의성을 해소한다. 그 다음 토큰화(Tokenizer) 단계를 수행하는데, 이때 형태소 분석기는 [12]의 Twitter 형태소 분석기를 사용하였다. 그리고 3.1절에서 설명한 것과 같이 개체-반영 토큰화를 수행하여 여러 단어로 구성된 하나의 개체는 하나의 토큰으로 인정하였다. 그 다음 Skip-gram 모델로 다중-어의 단어 임베딩을 학습하여 단어별로 의미 번호가 부착된 형태의 토큰들이 각각 임베딩 값을 가질 수 있게 하였다.

그 다음, 지식베이스(Knowledge Base)와 말뭉치(Corpus) 간 원격 지도 학습 데이터 수집(Distant Supervision)을 수행하고, 이때 수집된 문장은 단어 임베딩 학습과 동일한 방법으로 토큰화를 수행한다. 그리고 이 데이터(Labeled Data)를 두 그룹으로 나누어 한 그룹은 학습, 나머지 한 그룹은 평가에 사용한다. 관계 추출 모델은 [5]의 CNN 모델과 [6]의 PCNN 모듈 2개를 각각 학습 및 평가에 사용하였고, 두 모델 모두 가중치 매트릭스를 3개씩 사용하여 3개의 Convolution Layer를 생성한 다음 매트릭스 연결(concat)방식으로 결합(merge) 하였다. CNN 모델은 Single Pooling Layer 그리고 PCNN 모델은 Piecewise Max Pooling Layer 방식으로 구현하였으며, 나머지 층은 모두 동일하게 구성하였다.

4. 실험

4.1 실험 데이터

실험을 위해 한국어 위키피디아 2017년 7월 1일 버전의 말뭉치의 6,941,760 문장과 디비피디아(DBpedia) 2016-10 버전 기반의 K-Box를 지식베이스로 사용하였다. K-Box는 한국어 관계(Local Property)로 정의된 트리플을 디비피디아 공용 관계(Ontological Property)로 변환 및 확장한 지식베이스이다. 예를 들어, 'prop-ko:탄생지'와 같은 한국어 관계를 'dbo:birthPlace'와 같은 디비피디아 공용 관계로의 변환을 말하며, 관계 변환 테이블(Mapping Table)은 온톨로지 전문가 3명이 수작업으로 생성하였다. 원격 지도 학습 데이터 수집 결과 총 451개의 관계를 기준으로 358,464개의 Labeled Data를 수집하였고, 이때 대부분의 관계들이 수집된 데이터의 양이 적은 Long tail 문제에 해당한다. 다중 분류기(Multi-class Classifier) 모델에서 클래스 별 학습할 데이터가 적으면 제대로 학습이 진행되지 않기 때문에, 원활한 관계 추출 모델 학습을 위해 각 관계별로 수집 데이터의 개수가 1000개 이상인 68개 관계 기준 200,323개의 데이터를 학습 및 평가에 사용하였고, 수집 데이터 수 기준 Top 10개의 관계 통계는 표 1과 같다.

표 1 수집 데이터 기준 수 기준 Top 10 관계 통계

Property	# of collected data	Property	# of collected data
country	73,890	team	6,110
isPartOf	49,957	birthPlace	5,962
capital	12,953	successor	5,228
location	11,570	deathPlace	5,191
part	7,947	owner	4,971

4.2 실험 결과

본 논문에서 제안한 방식의 우수성을 입증하기 위해 총 3가지 방식의 단어 임베딩을 학습하였으며, 학습 시 공통으로 설정한 하이퍼파라미터(Hyperparameter)는 표 2와 같다. (1) 어절단위 토큰화, (2) 형태소단위 토큰화, (3) 어의중의성 해소 토큰화

표 2 단어 임베딩 하이퍼 파라미터

Dimension	Window	Min. Word
100	5	1

샘플 단어에 대한 유의어 결과는 표 3과 같다. 표에서 볼 수 있듯이 어의 중의성 해소 결과를 반영한 단어 임베딩은 그렇지 않은 단어 임베딩보다 동일한 단어에 대해 여러 의미로 구분이 가능하며, 각 의미마다 연관 있는 단어들끼리 군집이 이루어짐을 확인할 수 있다. 또한, 개체를 반영한 단어 임베딩을 학습하였기 때문에 여러 단어로 구성된 개체를 하나의 임베딩으로 학습하는

것을 확인할 수 있고 근접한 단어들도 상당히 의미있음을 확인할 수 있다.

학습된 다중-어의 단어 임베딩 결과를 활용하여 관계 추출 모델의 Held-out 성능 평가를 진행하였다. Held-out 평가는 수집된 데이터를 절반으로 나누어 한 그룹으로 학습, 나머지 한 그룹으로 평가를 진행하여 정확도(Precision), 재현율(Recall), 그리고 F1-score를 측정하는 방식이고, 그 결과는 표 5와 같다.

본 논문에서 제안한 관계 추출 모델 학습 시 어의 중의성 해소 임베딩의 효과를 확인하기 위하여 3개의 각기 다른 임베딩 값을 입력으로 사용하여 성능을 측정해보았고, 각 모델의 하이퍼파라미터는 표 4과 같이 설정하였다.

표 3 '시장'과 '사과'의 유사단어

Token	Word	Similar Words
Word/POS	시장	투자, 유통, 수익, 수출, 자산, 대기업, 매출, 수입, 업계, 가격
	사과	묻다, 사죄, 죄송하다, 건네다, 애도, 봉투, 제보, 하소연, 해명, 발언
Word/POS /WSD	시장-0 (물건을 사고파는 장소)	시장, 산업-0, 업계-0, 경쟁력, 중소기업-0, 사업-4, 투자-0, 산화방지제, 금융-0
	시장-1 (지방 자치 단체 장)	교육감-0, 기초자치단체장, 새누리당, 박순자, 고진화, 박영선_(1960년), 송광호, 대한민국_제5회_지방_선거, 전병헌
	사과-3 (자기의 잘못을 인정하고 용서를 빎)	사죄-0, 건네다-0, 고소하겠, ㅠㅠ, 봉투-0, 죄송하다, 모닝스타, 낸시랭, 앉은뱅이-
	사과-4 (사과나무의 열매)	과일-1, 완두-0, 잠자리-1, 밤-1, 포도-0, 뱀장어, 살구-0, 호두, 견과류, 옻나무
Entity	유엔	유럽_연합, UN, 국제_연합, 유럽_공동체, 북대서양_조약기구, 국제_연맹, 국제연합, 안전보장

표 4 모델별 하이퍼파라미터

Model	Activation	Optimizer	Dropout
CNN	RELU	ADADELTA	1
PCNN	RELU	ADAM	1

표 5 단어 임베딩 별 관계추출 모델 평가 결과

Model	Embedding	Precision	Recall	F1-score
CNN	Word	0.5537	0.3506	0.4275
	+POS	0.5315	0.4279	0.4739
	++WSD	**0.5921**	**0.5039**	**0.5443**
PCNN	Word	0.457	0.3251	0.3799
	+POS	0.4555	0.3472	0.394
	++WSD	0.4529	0.3713	0.4081

평가 결과, 두 모델 모두 단어만 사용한 임베딩보다 형태소를 함께 사용하는 것의 성능이 향상되었고, 형태소를 함께 사용한 것 보다 어의 중의성 해소 모듈 결과를 반영한 단어 임베딩을 사용하는 것의 성능이 향상됨을 확인할 수 있다. 단, 이때 모든 단어 임베딩 방법에서 개체-반영은 공통적으로 수행하였다. 그리고 PCNN 모델보다 CNN을 사용한 모델이 성능이 약간 더 높았는데, 한국어에서는 문장 내 두 개체의 위치가 문장의 맨 첫 번째에 나오는 경우, 그리고 두 개체가 연접해있는 경우가 많아 PCNN 보다 CNN 방법이 더 높은 성능을 보인 것으로 판단된다. 또한 하이퍼파라미터를 바꿔가며 수행한 여러 번의 반복 실험에서 Dropout은 하지 않는 것이 더 높은 성능을 보여주었다.

5. 결론 및 향후연구

본 논문에서는 원격 지도 학습 기반 관계 추출 모델에 다중-어의 단어 임베딩을 적용한 관계추출 모델 성능향상 방법을 제안하였고, CNN과 PCNN기반의 두 관계 추출 모델에 적용한 실험을 수행하였다. 또한 관계 추출을 위한 단어 임베딩 생성 시 여러 단어로 구성된 개체에 대해 하나의 토큰으로 반영하는 개체-반영 단어 임베딩을 기본적으로 활용하였으며, 그 결과 다중-어의를 해소한 단어 임베딩이 관계 추출 모델의 성능을 향상시킴을 확인할 수 있었다.

향후에는 자연언어의 특성인 시계열성을 반영하여 CNN과 RNN의 결합 모델인 Convolutional RNN 모델을 관계추출 문제에 적용하는 방향의 연구를 수행할 예정이다. 그리고 원격지도학습 데이터 수집 시 발생하는 문제점 중 하나인 에러 데이터 제거 방법에 대한 연구를 진행할 계획이다.

사사

이 논문은 2017년도 정부(미래창조과학부)의 재원으로 정보통신기술진흥센터의 지원을 받아 수행된 연구임 (2013-0-00109, WiseKB: 빅데이터 이해 기반 자가학습형 지식베이스 및 추론 기술 개발)

참고문헌

[1] Mintz, Mike, et al. "Distant supervision for relation extraction without labeled data." Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th

International Joint Conference on Natural Language Processing of the AFNLP: Volume 2-Volume 2. Association for Computational Linguistics, 2009.
[2] Sebastian Riedel, Sebastian Riedel, Limin Yao, and Andrew McCallum. "Modeling relations and their mentions with- out labeled text." In Proceedings of ECML PKDD, pages 148-163. 2010.
[3] Raphael Hoffmann, et al. "Knowledge-based weak supervision for information extraction of overlapping relations." In Proceedings of ACL, pages 541-550. 2011.
[4] Mihai Surdeanu, et al. "Multi-instance multi-label learning for relation extraction." In Proceedings of EMNLP-CoNLL, pages 455-465. 2012.
[5] Yoon Kim. “Convolutional neural networks for sentence classification.” In Proceedings of EMNLP, pages 1746-1751. 2014.
[6] Zeng, D., et al. "Distant Supervision for Relation Extraction via Piecewise Convolutional Neural Networks." In Proceedings of EMNLP, pages 1753-1762. 2015.
[7] Neelakantan, A., et al. "Efficient Non-parametric Estimation of Multiple Embeddings per Word in Vector Space." CoRR, cs.CL. 2015.
[8] Rothe, S., & Schütze, H. "AutoExtend: Extending Word Embeddings to Embeddings for Synsets and Lexemes." arXiv.org. 2015.
[9] Mikolov, Tomas, et al. "Distributed representations of words and phrases and their compositionality." Advances in neural information processing systems. 2013.
[10] Zeng, D., et al. "Relation classification via convolutional deep neural network." In Proceedings of COLING, pages 2335-2344. 2014.
[11] 이주상, 신준철, and 옥철영. "단어 의미와 자질 거울 모델을 이용한 단어 임베딩." *정보과학회 컴퓨팅의 실제 논문지* Vol.23 No.4, pages 226-231. 2017.
[12] 박은정, 조성준, “KoNLPy: 쉽고 간결한 한국어 정보처리 파이썬 패키지”, 제 26회 한글 및 한국어 정보처리 학술대회 논문집, 2014.
[13] 한국과학기술원 전문용어언어공학연구센터, “다국어 어휘의미망 제1권: 어휘의미망 구축론”, KAIST Press, 2005.

제한된 언어 자원 환경에서의 다국어 개체명 인식

천민아[o†], 김창현[‡], 박호민[†], 노경목[†], 김재훈[†]
한국해양대학교[†], 한국전자통신연구원[‡]
minah0218@kmou.ac.kr[†], chkim@etri.re.kr[‡], homin2006@daum.net[†],
kmq7542@gmail.com[†], jhoon@kmou.ac.kr[†]

Multilingual Named Entity Recognition with Limited Language Resources

Min-Ah Cheon[†], Chang-Hyun Kim[‡], Ho-min Park[†], Kyung-Mok Noh[†], Jae-Hoon Kim[†]
Korea Maritime and Ocean University[†], Electronics and Telecommunications Research Institute[‡]

요 약

심층학습 모델 중 LSTM-CRF는 개체명 인식, 품사 태깅과 같은 sequence labeling에서 우수한 성능을 보이고 있다. 한국어 개체명 인식에 대해서도 LSTM-CRF 모델을 기본 골격으로 단어, 형태소, 자모음, 품사, 기구축 사전 정보 등 다양한 정보와 외부 자원을 활용하여 성능을 높이는 연구가 진행되고 있다. 그러나 이런 방법은 언어 자원과 성능이 좋은 자연어 처리 모듈(형태소 세그먼트, 품사 태거 등)이 없으면 사용할 수 없다. 본 논문에서는 LSTM-CRF와 최소한의 언어 자원을 사용하여 다국어에 대한 개체명 인식에 대한 성능을 평가한다. LSTM-CRF의 입력은 문자 기반의 n-gram 표상으로, 성능 평가에는 unigram 표상과 bigram 표상을 사용했다. 한국어, 일본어, 중국어에 대해 개체명 인식 성능 평가를 한 결과 한국어의 경우 bigram을 사용했을 때 78.54%의 성능을, 일본어와 중국어는 unigram을 사용했을 때 각 63.2%, 26.65%의 성능을 보였다.

주제어: 개체명 인식, 다국어, limited language resources, sequence to sequence labeling

1. 서론

심층학습(deep learning)은 입력 데이터들에 대해 높은 수준의 추상화된 정보를 추출할 수 있다. 이로 인해 자연어처리, 영상처리 등과 같은 분야에서는 최적의 자질 조합을 찾기 위해 많은 시간과 노력이 필요했던 기계학습 알고리즘 대신 심층학습을 이용한 연구가 활발히 진행되고 있다[1]. 심층학습 모델 중 순차 데이터(sequential data)를 모델링 하는 방법인 LSTM과 출력 데이터(output data) 간의 전이 확률을 추가시킨 LSTM-CRF 방식이 개체명 인식 및 품사 태깅 문제에서 높은 성능을 보이고 있다[1-3].

개체명(named entity)은 문서에서 나타나는 고유한 의미를 가지는 명사이다. 개체명은 크게 인명(Person), 지명(Location), 기관명(Organization)으로 나눌 수 있다. 개체명 인식(named entity recognition)은 문서에서 개체명을 추출하고, 추출된 개체명의 종류를 결정하는 작업이다[4]. 한국어 개체명 인식의 성능을 향상을 위해 LSTM-CRF 모델과 사전 등의 외부 자원을 이용한 연구가 진행되고 있다[5-8]. 그러나 공개된 자원이 풍부한 영어에 비해 한국어와 일본어와 같은 언어들은 자연어처리에서 사용할 수 있는 자원과 공개된 자연어 처리 모듈이 한정적이다. 본 논문은 LSTM-CRF와 최소한의 언어 자원만을 사용하여 다국어에 적용 가능한 개체명 인식 방법을 찾는 데 초점을 맞춘다.

본 논문의 구성은 다음과 같다. 2장에서는 관련 연구를 소개하고, 3장에서는 다국어 개체명 인식에 관한 LSTM-CRF 모델을 소개한다. 4장에서는 한국어, 일본어, 중국어에 대해 3장의 모델을 적용한 결과를 분석하고, 마지막으로 5장에서 결론 및 향후 연구에 관해 기술한다.

2. 관련 연구

개체명 인식은 문서에서 인명, 지명, 기관명 등의 고유한 의미를 나타내는 단위인 개체명을 추출하고, 추출된 개체명의 종류를 결정하는 작업이다[4]. 개체명 인식의 성능 향상은 정보검색 분야에서 활발하게 연구되고 있는 질의응답이나 기계번역 시스템의 성능 개선을 위해 필수적이다[4].

개체명 인식이 어려운 이유의 핵심 키워드는 미등록어(unknown word)와 중의성(ambiguity)이다. 언어의 특성상 시간이 흐름에 따라 새로운 단어가 계속 생겨나고, 해당 단어들을 모두 개체명 사전에 등록할 수 없기 때문에 사전으로 개체명을 처리하는 데는 한계가 있다. 또 문맥에 따라 같은 단어가 개체명이 될 수도, 되지 않을 수도 있으며 시간의 흐름에 따라 과거 개체명이 아니었던 단어가 개체명이 되거나 그 반대의 경우도 있다. 과거에는 규칙 기반이나 통계 기반의 기계학습을 이용하여 개체명 인식을 처리했으나[9-10], 최근 심층학습 모델 중 하나인 LSTM-CRF 모델이 좋은 성능을 보이고 있다[1-3,5-8]. [5-8]은 LSTM-CRF를 한국어 개체명 인식에 적용한 논문이다. [5-6]은 입력된 문자열의 각 문자를 양방향 LSTM을 적용하여 문자 임베딩을 얻은 후, CNN으로 합성하여 단어의 임베딩 벡터를 얻는다. 단어 임베딩 벡터에 품사 정보, 띄어쓰기 정보, 사전 자질 등의 외부 자원을 추가하여 양방향 LSTM-CRF의 입력이 되는 확장단어 임베딩을 구성하여 86.53%의 성능을 얻었다. [7]은

단어/품사 임베딩을 기본으로 음절 정보, 각 음절의 개체명 품사 분포 정보, 사전 자질 벡터를 결합하여 80.68%의 성능을 보였다. [8]의 연구 역시 형태소/품사를 입력으로 하여 형태소 임베딩, 자음/모음 자질, 품사, 기구축 사전 정보를 결합하여 85.71%의 성능을 보였다. 이처럼 기존의 개체명 인식 연구에서는 다양한 자질의 조합과 외부 자원을 통한 성능 향상에 초점을 맞추고 있다. 이러한 방법은 학습에 사용할 수 있는 자원이 상대적으로 부족한 언어에 대해서는 성능 향상이 쉽지 않다는 문제점이 존재한다.

3. 다국어 개체명 인식을 위한 LSTM-CRF

공개된 자원이 풍부한 영어에 비해 한국어와 일본어와 같은 언어들은 자연어처리에서 사용할 수 있는 자원이 한정적이다. 특히 모국어가 아닌 외국어에 대한 자연어처리에서는 해당 언어에 대한 깊은 이해가 필요하므로 필요한 데이터를 수집 및 가공하는 일에 어려움이 따른다. 본 장에서는 최소한의 학습 자원을 사용한 다국어 개체명 인식에 관해 설명한다.

3.1 입력 형식 : 문자 단위의 n-gram 기반의 임베딩

기존 연구는 단어, 형태소, 품사 임베딩 벡터를 입력 노드의 기본 입력 단위로 삼는다. 해당 임베딩 벡터에 문자(음절)에 대한 정보를 결합하는 형태로 확장하는데, 본 논문에서는 문자의 n-gram을 입력 단위로 사용하여 gensim[11]의 word2vec 기능을 이용하여 임베딩 시킨다. n-gram을 임베딩 시키기 전, 아래의 규칙을 적용한다.

1. **공백 문자**는 **@sp@**로 치환
2. 학습할 말뭉치에서 나타난 **n-gram의 빈도수가 5 이하**인 경우, 해당 n-gram은 **@unk@**로 치환
3. n-gram을 위한 **패딩 문자**는 **@pad@**로 치환
4. 이 외의 n-gram은 그대로 사용

n-gram 임베딩 학습 parameter에 관한 정보는 다음과 같다. 학습 단위는 unigram, bigram이며, 임베딩 차원의 크기는 32, 64, 128, 256이다. window size는 10으로 정했다. window size가 10인 이유는 단어 임베딩의 경우 window size가 기본 4로 고정되어 있기 때문이다. 한국어, 중국어, 일본어 단어의 평균 문자수가 2~3글자이므로 2.5글자×4(단어의 window)=10이라는 숫자를 도출했다. 그 외의 parameter는 gensim의 기본값으로 설정했다.

3.2 출력 형식 : 개체명 태그

기존의 한국어 개체명 인식기에서는 위치 표시 접두어 방법인 BIO(Begin, Inside, Outside)나 BIEOS(Begin, Inside, End, Outside, Single) 태그와 개체명 태그를 결합한 태그를 사용했다. 본 논문에서는 [3]과 같이 각 개체명 태그가 최장 일치법일 경우를 가정하여 표시 접두어를 제외한 개체명 태그 그 자체를 사용한다. 각 위치의 출력 개체명 태그는 그림 1과 같이 각 n-gram의 가장 처음에 오는 문자에 대응되는 개체명 태그를 기준으로 한다. 진한 표시의 음절이 개체명 태그와 대응된다.

<**서호프**:PES>와 <**파사노**:PES>에게 연속 안타를									
서호	**호**프	**프**와	**와**_	_파	**파**사	**사**노	**노**에	…	**를**#
PES	**PES**	**PES**	O	O	**PES**	**PES**	**PES**	…	O

그림 1. n-gram 기반 개체명 인식에서 출력 개체명 태그 형식 예시

3.3 양방향 LSTM-CRF

그림 2는 양방향 LSTM-CRF의 구조이다. 양방향 LSTM은 각 LSTM cell에 입력 문자열을 양방향으로 받아서 각 입력 단위에 대해 은닉 벡터를 얻는다. 이 결과에 전이 확률(의존성)을 추가한 것이 LSTM-CRF이다. 입력은 3.1절에서 설명한 n-gram 기반의 임베딩 벡터이고, 최종 출력은 3.2에서 설명한 개체명 태그이다. 설명한 데이터 외의 형태소, 품사 정보나 기구축 사전 등의 외부 자원은 사용하지 않는다.

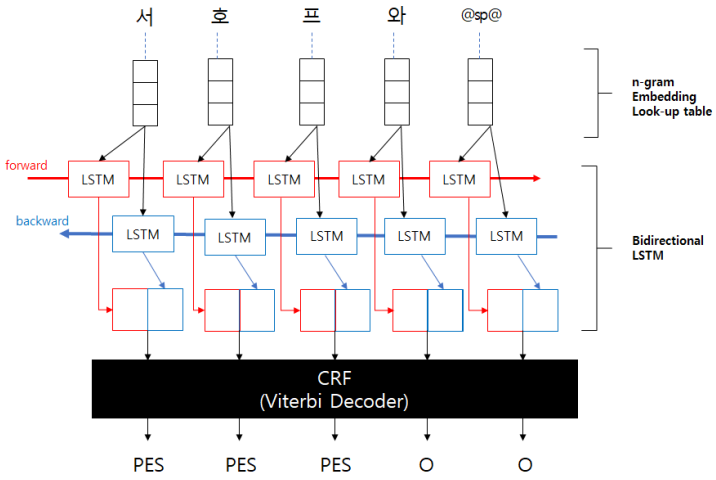


그림 2. 양방향 LSTM-CRF 모델 (unigram을 입력으로 했을 때)

4. 실험

제안한 개체명 인식에 대한 성능 평가를 위한 3.3의 전체 모델 구현은 gensim[11]과 Tensorflow[12]를 사용했다.

4.1 실험 환경

다국어 개체명 인식에 대한 대상 언어는 한국어, 일본어, 중국어이다. 개체명 인식의 범위는 인명(PES), 지명(LOC), 기관명(ORG), 날짜(DAT), 시간(TIM)의 다섯 가지다. 각 언어에 대한 개발, 학습, 평가 말뭉치에 대한 정보는 표 1과 같다. 한국어는 ETRI에서 배포한 개체명 말뭉치[13] 5,000문장을 사용했다. 일본어는 개인이 구축하여 공개한[14] 개체명 말뭉치 500문장을 표1의 일본어前의 항목처럼 나눈 후, 각 항목의 문장을 5배씩 증가시킨 상황에서 실험했다. 중국어는 CONLL2007형식으로 제

작되어 배포된 웨이보 개체명 말뭉치[15] 1,887문장을 사용했다.

표 1. 성능 평가에 사용한 각 언어 개체명 말뭉치 정보

언어	개발(dev)	학습(train)	평가(test)
한국어	**500**	**3,500**	**1,000**
일본어前	50	400	100
일본어	**250**	**2,000**	**500**
중국어	**269**	**1,349**	**269**

n-gram 임베딩 실험에서 본 논문에서 성능 평가를 할 때 사용한 단위는 unigram과 bigram이다. 한국어 n-gram 임베딩은 뉴스 말뭉치 약 2GB를 사전학습(pre-train)한 결과를 사용했다. 일본어와 중국어의 경우 대량의 뉴스 말뭉치를 모으는 도중이라 개발, 학습, 평가 말뭉치로부터 n-gram 임베딩을 사전 학습했다.

성능의 평가 단위는 각 개체명의 청크(chunk) 단위이다. 전체적인 실험 성능은 개발 데이터에서 가장 좋은 성능을 보인 15 epoch에서 평가했다. 실험에 사용한 평가방법은 precision, recall, F_1-measure이다.

$$precision = \frac{|\text{시스템(개체명)} \bigcap \text{정답(개체명)}|}{|\text{시스템(개체명)}|}$$

$$recall = \frac{|\text{시스템(개체명)} \bigcap \text{정답(개체명)}|}{|\text{정답(개체명)}|}$$

$$F_1 - measure = \frac{2 \times precision \times recall}{precision + recall}$$

양방향 LSTM-CRF에서 입력의 크기는 n-gram 임베딩 벡터의 크기와 같다. mini-batch는 50, 100, 200일 때 hidden node의 수는 128, 256개 일 경우에 대해 각각 실험했다. 최적화에는 Adam 함수를 사용했다.

4.3 실험 결과

표 2는 각 언어에 대한 개체명 인식 성능 평가 결과를 unigram, bigram에 대해 각각 정리한 것이다. 각 입력 단위에서 가장 좋은 성능을 냈을 때의 상황을 보여준다. 입력은 임베딩의 크기를 나타내고, n-gram은 입력에 사용한 n-gram의 종류를 나타낸다. mini-batch는 각 batch에서 학습한 문장의 수를 나타내고, 은닉 노드는 양방향 LSTM에서 forward와 backward를 구성하는 LSTM cell의 개수이다. 은닉 노드가 128이라면 forward와 backward에 각 128개의 노드가 부여되어 최종적으로는 256개의 은닉 노드를 사용한 것이 된다. 한국어의 경우 bigram을 사용했을 때가 unigram을 사용했을 때보다 2.5%p 정도 향상된 성능을 보였다. 일본어와 중국어의 경우에는 unigram을 사용했을 때가 bigram을 사용했을 때보다 더 높은 성능을 나타냈다.

표 2. 양방향 LSTM-CRF를 사용한 다국어 개체명 인식 성능 평가 결과

언어	입력	n-gram	mini-batch	은닉 노드	precision	recall	F_1
한국어	64	unigram	200	256	84.30	69.26	76.04
	256	bigram	200	256	**84.46**	**73.40**	**78.54**
일본어	64	unigram	100	128	**65.86**	**60.75**	**63.20**
	128	bigram	50	128	24.62	27.30	25.89
중국어	128	unibram	100	128	**31.48**	**23.10**	**26.65**
	256	bigram	100	256	7.79	16.76	10.64

(※ 모든 성능 평가를 위한 실험에서 사용한 epoch 수는 15로 고정했다.)

중국어의 경우 unigram과 bigram 모두 다른 언어에 비해 현격히 낮은 성능을 보이는 것을 볼 수 있다. 실제 같은 말뭉치를 사용하여 중국에서 진행된 중국어 개체명 인식 결과[16]는 F_1-measure가 44.09%이다. 한국어의 경우에는 충분한 양의 뉴스 말뭉치를 사용하여 유효한 n-gram 임베딩을 구축할 수 있었으나, 일본어와 중국어의 경우에는 개체명 말뭉치로부터 n-gram 임베딩을 구축했기 때문에 n-gram의 임베딩이 충분하게 학습되었다고 보기 어렵다. 일본어와 중국어 모두 한자를 포함하는 언어이다. 유니코드에 등록된 한중일 통합한자의 경우 그 수가 8만여 개에 이른다. 특히 중국어의 경우에 같은 문자를 쓰는 방법이 번자체와 간자체의 두 종류가 있다. 예를 들어 바퀴가 달린 수레를 의미하는 차(車)를 번자체로 쓸 경우 車가 되지만 간자체로 쓸 경우 车로 표기된다. 즉, n-gram을 충분히 반영하기 위해서는 더 많은 말뭉치가 필요하다는 의미이다. 일본어와 중국어에 대해 많은 양의 뉴스 말뭉치를 수집하여 n-gram 임베딩을 추가 학습한 뒤, 다시 성능평가를 해 볼 필요가 있다.

한국어의 경우에는 unigram과 bigram에 대해 모두 76% 이상의 성능을 보였다. 이는 사전학습된 단어/품사 임베딩이나 형태소/품사 임베딩만을 사용했을 때와 비슷한 수준의 결과이다.

5. 결론 및 향후 연구

본 논문에서는 양방향 LSTM-CRF를 사용하여 한국어, 일본어, 중국어의 개체명 인식을 실험하고 그 결과를 살펴봤다. 형태소, 품사나 사전 정보 등의 자원이 충분하지 않을 경우를 고려하여 문자를 n-gram 단위로 입력했을 때의 결과를 살펴봤다. 한국어의 경우 unigram을 사용했을 때 76.04%, bigram을 사용했을 때는 2.5%p 증가한 결과인 78.54%의 성능을 보였다. 이는 문자를 n-gram 단위로 사전 임베딩 하는 것으로 단어/품사 임베딩이나 형태소/품사 임베딩만을 사용했을 때와 비슷한 수준의 결과에 도달할 수 있다는 의미이다. 일본어와 중국어의 경우에는 문자 n-gram을 사전학습하기 위한 말뭉치가 충분하지 않아 bigram을 사용했을 때가 unigram을 사용했을 때보다 현격히 낮은 성능을 보였다.

향후에는 일본어와 중국어의 뉴스 말뭉치를 한국어와 비슷한 수준까지 수집하여 문자 단위의 n-gram 임베딩을 다시 학습한 후, 성능 평가를 진행할 예정이다. 또한,

학습 말뭉치에서 얻을 수 있는 정보인 문자 단위 n-gram의 개체명 품사 분포 정보와 n-gram 임베딩을 결합한 결과를 입력으로 사용하여 다국어 개체명 인식에 대한 성능 평가할 계획이다.

감사의 글

이 논문은 2017년도 정부(과학기술정보통신부)의 재원으로 정보통신기술진흥센터의 지원을 받아 수행된 연구임(R7119-16-1001, 지식증강형 실시간 동시통역 원천기술 개발)

참고문헌

[1] Ronan Collobert, et al., Natural language processing (almost) from scratch. The Journal of Machine Learning Research, 12, 2011.
[2] Xuezhe Ma, Eduard Hovy, “End-to-end Sequence Labeling via Bidirectional LSTM-CNNs-CRF”, arXiv:1603.01354, 2016.
[3] Onur Kuru, Ozan Arkan Can, and Deniz Yuret, “CharNER: Character-Level Named Entity Recognition”, arXiv:1603.01354, 2016.
[4] David nadeau and Stasoshi Sekine, “A survey of named entity recognition and classification”, Journal of Linguisticae Investingations, 30(1): 3-26, 2007.
[5] 민진우, 오효정, 나승훈, “식품 도메인 개체명 인식을 위한 문자 기반 LSTM CRF”, 한국정보과학회 2016년 동계학술대회 논문집, pp.500-502, 2016.
[6] 나승훈, 민진우, “문자 기반 LSTM CRF를 이용한 개체명 인식”, 한국정보과학회 2016년 한국컴퓨터종합학술대회 논문집, pp.729-731, 2016.
[7] 유홍연, 고영중, “Bidirectional LSTM CRF 기반의 개체명 인식을 위한 단어 임베딩의 확장”, 정보과학회논문지 44(3):306-313, 2017.
[8] 신유현, 이상구, “양방향 LSTM-RNNs-CRF를 이용한 한국어 개체명 인식기”, 제28회 한글 및 한국어 정보처리 학술발표 논문집, pp.340-341, 2016.
[9] Sergey Brin, “Extracting Patterns and Relations from the World Wide Web”, WebDB '98 Selected papers from the International Workshop on The World Wide Web and Databases, pp.172-183, 1998.
[10] Daniel M. Bikel et al., “Numble: a High-Performance Learning Named-finder”, In: Proc, The Fifth Conference on Applied Natural language Processing, 1997.
[11] gensim [Online] https://radimrehurek.com/gensim/index.html
[12] Onur Kuru, Ozan Arkan Can, and Deniz Yuret, “CharNER: Character-Level Named Entity Recogition”, Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers, pp.911-921, 2016.
[13] Tensorflow [Online] https://www.tensorflow.org/
[14] ETRI 지식마이닝 연구실, 엑소브레인 언어분석 말뭉치(ver. 1.0), 2015.
[15] 일본어 개체명 말뭉치 [Online] https://github.com/Hironsan/IOB2Corpus
[16] 중국어 개체명 말뭉치 [Online] https://github.com/hltcoe/golden-horse/tree/master/data

한국어 특질을 고려한 단어 벡터의 Bi-LSTM 기반 개체명 모델 적용

남석현O, 함영균, 최기선

한국과학기술원
obiwan96@kaist.ac.kr, hahmyg@kaist.ac.kr, kschoi@kaist.ac.kr

Application of Word Vector with Korean Specific Feature to Bi-LSTM model for Named Entity Recognition

Sukhyun NamO, Younggyun Hahm, Key-Sun Choi
KAIST

Deep learning의 개발에 따라 개체명 인식에도 neural network가 적용된 연구가 활발히 일어나고 있다. 영어권 개체명 인식에서는 F1 score 90%을 웃도는 성능을 내는 연구들이 나오고 있다. 하지만 한국어는 영어와 언어적 특질이 많이 달라 이를 그대로 적용시키는 데는 어려움이 있어 영어권 개체명 인식기에 비해 비교적 낮은 성능을 보인다. 본 논문에서는 "하다" 접사의 동사형이 보존된 워드 임베딩을 사용하고 한국어 개체명의 특징을 담은 one-hot 벡터를 추가하여 한국어의 특질에 보다 적합한 데이터를 deep learning 기술에 적용하였다.

주제어: 개체명 인식

1. 서론

개체명 인식(Named Entity Recognition)은 문서로부터 개체명(Named Entity)을 추출하고, 추출된 개체명의 종류를 분류하는 자연언어처리의 한 분야이다. 개체명은 문서에서 나타나는 고유한 의미를 가지는 명사나 숫자 표현 등으로, 대체로 하나 이상의 단어가 결합되어 구성된다. 본래는 시간이나 수식 표현까지 포함하는 포괄적인 개념이나, 본 연구에서는 인명(PS), 지명(LC), 기관명(OG)의 고유명사와 숫자(DT), 시간(TI)을 대상으로 한정한다. 표 1은 개체명이 표시된 문장의 예시이다.

'<부천 영화제 사무국:OG>'은 <내달 3일:DT> <오후 3시30분~5시30분:TI> <부천:LC> <중동:LC>신도시 <중앙공원:LC>에서 'PiFan 사랑 걷기 대회'를 개최한다.

표 1. 개체명이 표시된 문장 예

개체명 인식에 대한 연구는 1995년, MUC-6(the Sixth Message Understanding Conference)[1]에서 처음 촉발되었다. MUC-6에 참가한 많은 시스템들은 특정 언어에 제한된 규칙과 제한된 입출력방법을 사용하여 다른 언어나 시스템에 적용하지 못하는 문제가 있었으나, 이후 기계학습 방식과 BIO 태깅의 통일된 입출력방식을 도입하여 체계적으로 연구되어오고 있는 분야이다. 이 때, BIO태깅이란 개체명의 시작을 "B"로, 개체명이 이어지고 있는 경우에는 "I"로, 개체명이 아닌 경우에는 "O"로 태깅하는 방식을 일컫는다. 정보 검색, 질의응답 시스템 등 매우 다양한 분야의 시스템에서도 개체명 인식이 사용됨에 따라 성능향상을 목표로 현재까지도 연구되어오고 있다.

최근에는 자연언어처리뿐만 아니라 많은 분야에서 괄목할 만한 성과를 보여주고 있는 딥러닝 기술의 개발로 개체명 인식에서도 이를 이용한 연구가 진행되고 있다. 문장 내 단어의 의미를 분산된 에너지 벡터로 표현할 수 있는 연구인 단어 임베딩(word embedding) 방법론의 개발로[2] 단어를 벡터화 시킨 후 deep neural network를 적용시킬 수 있게 되었고, 상당한 성능을 보여주고 있다. 영어의 경우에는 딥러닝을 이용하여 90%를 웃도는 높은 정확도를 보이는 연구들이 나오고 있다[3][4]. 하지만 한국어는 언어적 특질이 영어와 달라 영어 개체명 인식에서 쓰인 기술을 그대로 적용하기에는 어려움이 있고 따라서 영어권 개체명 인식에 비해 다소 낮은 성능을 보이고 있다. 또한 최근 한글 개체명 인식에도 딥러닝 기술을 적용한 연구가 나오고 있다[5]. 이에 본 논문에서는 한국어의 특징을 분석하여 딥러닝 기술을 한국어에 알맞게 적용시켜 더 높은 성능을 보이고자 하였다.

본 논문에서는 기존의 BiLSTM을 이용한 영어 개체명 인식 모델[6]을 구현하고 이에 대한 다음의 추가 작업을 통해 성능이 더 향상될 수 있음을 보였다.

1) "하다" 접사의 동사형이 보존된 단어 임베딩 사용
2) 한국어 개체명의 특징을 담은 one-hot 벡터 추가
3) 평가데이터 오류 수정

2. 개체명 학습 모델

본 논문에서는 딥러닝의 다양한 모델 중에서도 최근의 영어권과 한국어 개체명 인식에서 모두 가장 높은 성능을 보이고 있는 BiLSTM을 기반으로 하여 CRF방식을 결합한 BiLSTM-CRF 방식을 이용하였다. 그림 1은 BiLSTM을 사용한 개체명 인식기의 흐름도를 보여주고 있다.

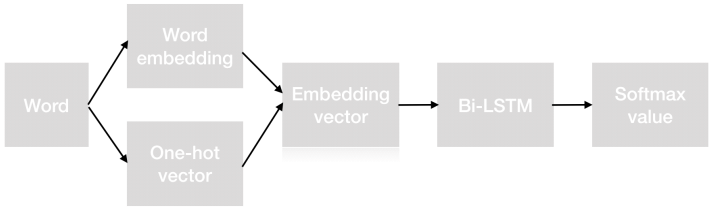


그림 1. 개체명 인식 BiLSTM 흐름도

개체명 인식은 POS 태깅이 완료된 데이터를 대상으로 이루어진다. POS태깅이 되어있는 데이터 셋에 대해 각 단어를 워드 임베딩과 one-hot벡터를 합친 임베딩 벡터로 변환시킨다. 학습데이터의 임베딩 벡터를 BiLSTM을 통해 학습시켜 BiLSTM 모델을 생성한다. 학습된 모델에 테스트 데이터의 임베딩 벡터를 입력하여 나온 벡터를 softmax 함수에 적용시켜 개체명 인식 결과를 얻을 수 있다.

2.1 “하다” 접사의 동사형이 보존된 워드 임베딩 사용

본 논문에서는 한국어 위키피디아를 대상으로 skip-gram모델의 word2vec[7]을 사용하였다. 벡터 값의 차원은 100, 300, 500으로, 창크기는 3, 5로 변화시키며 어떤 환경에서 생성된 워드 임베딩 데이터를 이용하였을 때 성능이 가장 높은지 확인하여보았다. 생성된 데이터는 약 26만여 개의 단어를 포함한다.

기존의 한국어 자연언어 처리에서 워드 임베딩을 활용하는 연구들은 워드 임베딩을 생성하면서 형태소 분석을 한 후 생성하였다. 그 때문에 동사표현인 ‘출생하’를 예로 들면 ‘출생/NNG’와 ‘하/XSV’의 각각의 임베딩을 생성하는 연구가 이루어졌다[5][8]. 본 논문에서는 개체명 인식에 있어서는 개체명 앞 혹은 뒤에 동사가 있는 것이 중요한 것으로 판단하여, 이러한 경우 ‘출생하/VV’에 대한 임베딩을 생성하는 차이를 두었다.

이러한 워드 임베딩 생성 방식에 차이를 두었을 때 명사형을 가져왔을 때 발생하는 오차를 없앨 수 있다. 예를 들어 문장에서 ‘조정하다’와 같은 경우 ‘조정’이 ‘조정하’의 동사적 의미로 쓰인 것인데, ‘조정/NNG’의 워드 임베딩을 가져오게 되면 운동 종목인 조정의 워드 임베딩을 가져와 오류가 발생한다. 표 2에 명사로 분석을 하게 되면 뜻이 달라지는 단어들을 평가 데이터에서 찾아 일부를 예시로 나타내었다. “하다” 접사의 동사형이 보존된 워드 임베딩을 사용하면 이러한 오류를 없앨 수 있으며, 이에 대한 성능 비교를 3장에 수행하였다. 2016 국어 정보 처리 경진대회에서 제공한 워드임베딩도 위키피디아를 대상으로 워드 임베딩을 제작하였기 때문에 성능 비교 대상으로 사용하였다.

지적, 자부, 주도, 구가, 전역, 투구, 선사, 대패, 구상, 주도, 대신

표 2. 명사형으로 분석하였을 때 동사형과 의미가 달라지는 단어의 예시

임베딩 데이터를 사용함에 있어서 워드 임베딩 대상 corpus에 존재하지 않아 embedding vector가 존재하지 않는 단어에 대해서 같은 차원의 벡터를 임의의 값으로 채워서 사용하는 경우가 많다[6]. 본 연구에서는 임의의 값으로 채운 백터를 사용하여보기도 하였고, 일관적으로 같은 차원의 zero vector를 사용하여보기도 하였다.

2.2 한글 one-hot 벡터

영어의 경우, 워드임베딩 이외에도 추가적인 자질로서 품사태그 정보와 대문자로 시작하는지 등을 워드 벡터에 포함시켰을 때 성능이 향상되는 연구가 이루어졌다[4][6]. 이 때, 영어의 경우 개체명은 대체로 대문자로 시작하기 때문에 이러한 정보가 개체명 인식에 굉장히 도움이 된다. 이에 따라 한국어의 경우에도 한국어의 자질을 워드 벡터에 포함할 필요성이 있다. 본 논문에서는 한국어 개체명의 특징을 담고 있는 one-hot 벡터를 생성하였고, 이를 개체명 모델에 적용하여 비교평가를 수행하였다.

데이터 셋에서 개체명으로 태깅된 단어들의 특징을 분석하여서 공통적으로 어떤 특징이 있는지 분석하여보았다. 특히 한국어는 어원이 한자인 이루어진 단어가 많아, 개체명에 공통적으로 포함되어있는 글자가 많다. 다음은 각 개체명 종류 별로 이를 분석한 것이며, 괄호 안은 평가 데이터 셋에서 해당 조건을 만족하는 어휘가 개체명일 확률을 의미한다.

LC - ‘국’(71%), ‘동’(41%), ‘구’(35%), ‘시’(34%), ‘도’(19%)로 끝남
OG - ‘팀’(26%), ‘회’(35%)로 끝남
DT - ‘지나’(90%), ‘올해’(97%), ‘월’(98%), ‘일’(97%), ‘년’(94%)을 포함
TI - ‘오전’(100%), ‘오후’(100%), ‘분’(72%), ‘시’(49%)를 포함

이때, 50%가 넘는 특징들만 one-hot 벡터의 특징으로서 사용하였다. 다음은 그 결과 선택된 특징들이다.

- 글자 수가 셋 이상이다.
- 한글이 아닌 숫자로 이루어져있다.
- ‘지나’, ‘올해’, ‘월’, ‘일’, ‘년’을 포함한다.
- ‘국’으로 끝난다.
- ‘오전’, ‘오후’, ‘분’을 포함한다.

또한 품사태그 정보를 확인하여 일반명사인 NNG와

의존명사인 NNB를 묶어서 일반 명사, 외래어인 SL, 고유명사인 NNP, 동사인 VV, 그 외의 5가지로 나누어 vector 정보를 추가하여주었다. 또한 괄호 속에 들어가있는 단어들도 이에 대한 정보를 표기하여 총 11차원의 one-hot vector를 생성하였다. One-hot 벡터는 워드 임베딩 벡터의 끝에 추가하여서 최종 임베딩 벡터를 생성하였다.

2.3 BiLSTM

BiLSTM은 순차적 데이터 활용에 있어서 가장 많이 쓰이는 딥러닝 모형인 LSTM을 두 개를 함께 학습시켜, 각 데이터에 대해 왼쪽(forward)뿐만 아니라 오른쪽(backword) 데이터를 고려하도록 보완한 모델이다. 특히나 앞뒤 문맥을 모두 고려해야하는 자연언어처리에서 높은 성능을 보이는 알고리즘이다. BiLSTM 구현에는 구글에서 오픈소스로 공개한 기계학습 라이브러리인 텐서플로우(TensorFlow)[9]를 활용하였다. Hidden dimension이 256이고 drop out을 0.5로 설정한 LSTM을 2 layer로 생성하여 backword, frontword cell로 설정하였다. Optimizer로는 Adam optimizer을 사용하였으며, batch size는 128로 설정하여 생성된 BiLSTM을 train data로 학습시키며 매번 학습된 BiLSTM으로 test data에 적용시켜 인식결과를 확인하였다. 매번 F1 score를 측정하여 만약 최대치를 기록하면 학습된 BiLSTM을 저장시키는 방식으로 50 epoch 학습시켰다.

2.4 CRF

개체명 인식을 실제로 사용하기 위해서는 각각의 형태소를 태깅하는 것만으로는 부족하다. BIO태깅을 하기 위해서는 하나의 개체명을 추출해낼 수 있어야하는데, '연세대학교 원주캠퍼스'를 예로 들었을 때, 연세대학교와 원주캠퍼스를 같은 개체명으로 인식하여 '연세대학교 B-OG 원주캠퍼스 I'로 태깅해야 한다. BiLSTM은 각각의 형태소를 태그 벡터로서 출력하며 LC, PS, OG, DT, TI, O 중 어느 것인지 결정할 뿐 하나의 개체명이 무엇인지 알아내지 못하는 문제가 있다. 이 때문에 CRF를 추가할 필요성이 있었다. CRF에는 CRF++ Tool[10]을 활용하였다.

2.5 개체명 사전 활용

태깅 결과를 분석한 결과 개체명의 boundary를 제대로 찾지 못해 틀린 경우가 전체 오류의 6.1%를 차지했다. 이에 개체명 사전을 활용하여 이미 찾은 개체명의 boundary를 확정짓는 함수를 추가하였다. 개체명 사전의 데이터 베이스는 [11]에서 사용한 데이터 베이스를 활용하였다. 개체명인 단어의 다음 단어가 명사(NN)이며 개체명이 아닌 경우, 개체명에 다음 단어를 포함시켜 개체명 사전에 검색하였을 때 존재한다면 그 단어도 개체명에 포함시키도록 하는 함수를 추가하여주었는데, F1 score 기준 추가하기 전보다 1%낮아져 효과가 없었다.

3. 성능 평가 및 분석

3.1 성능 평가

데이터 셋은 2016 국어 정보 처리 경진대회[12]에서 제공한 데이터 셋을 활용하였다. 학습 데이터는 3,555문장으로 이루어져있으며, 실험 데이터는 501문장으로 이루어져있다.

3.1.1 성능 평가 방식

성능 평가 방식은 두 가지를 이용하였다. 첫 번째는 각각의 형태소에 대해 개체명 태깅이 맞았는지를 확인하여 F1 score를 측정하는 방법이다 (3.1.2절과 3.1.3절). 두 번째는 찾아낸 개체명과 정답 데이터 셋을 비교하여 개체명을 정확히 찾아내었는지 확인하여 F1 score를 측정하는 방식이다 (3.1.4절). 두 번째 방식은 2016 국어 정보 처리 경진대회에서 채점에 사용된 프로그램을 그대로 사용하였다.

워드 임베딩 데이터		F1(%)
2016 국어 정보 처리 경진대회 워드 임베딩		76.88
"하다" 접사의 동사 표현을 보존한 워드 임베딩	300차원, 창 크기 3	**83.82**
	300차원, 창 크기 5	83.11
	500차원, 창 크기 3	83.64
	500차원, 창 크기 5	83.29
	100차원, 창 크기 3	83.26
	300차원, 창 크기 3 + one-hot 벡터	**84.19**

표 3. 워드 임베딩 데이터 별 개체명 인식기 성능

3.1.2 워드 임베딩에 따른 개체명 인식 성능

위키피디아로부터 워드 임베딩 데이터를 생성할 때 차원과 창 크기를 변화시켜가며 성능비교를 하여보았다. 또한, 성능 평가를 위하여 2016 국어 정보 처리 경진대회에서 함께 제공한 워드 임베딩 데이터를 이용하였을 때와도 비교를 하여보았다. 결과는 표3과 같다.

대체적으로 2016 국어 정보 처리 경진대회 워드 임베딩 데이터를 이용했을 때보다 6% 가량 높아 훨씬 좋은 성능을 보임을 알 수 있다. 또한 300차원과 창 크기는 3일 때 성능이 가장 높아 해당 워드 임베딩을 이후로도 사용하였다.

3.1.3 One-hot벡터 및 워드 임베딩 외의 벡터

3.1.1의 결과에서는 one-hot 벡터를 추가하지 않고 임베딩 벡터를 생성할 때 워드 임베딩에 없는 단어들에 대해서는 같은 차원의 랜덤 벡터를 생성하여 사용하였다. 한국어 개체명의 특징을 분석하여 생성한 one-hot 벡터를 추가하였을 때 F1 score가 84.19%로 소량 증가하였다. 워드 임베딩에 없는 단어들에 대하여 랜덤 벡터가 아닌 zero 벡터로 대체를 한 경우는 84.73%로 가장 높은 성능을 기록하였다.

3.1.4 CRF

BiLSTM의 결과는 각각의 형태소가 어느 개체명에 포함되는지 벡터로 출력된다. CRF를 추가하였을 때와의 성능 비교를 위해 같은 개체명이 연속해서 태깅되면 같은 개체명이라 정하여 json 데이터를 형성하여 평가하였다.

CRF를 사용하지 않았을 때 66.2%, CRF를 사용하였을 때 72.8%로 성능이 향상되었음을 확인하였다..

3.2 오류 분석

3.1.3에서 방식1의 F1 score가 84.73%일 때 기준으로 테스트 데이터 셋에 있는 17,394개의 단어 중 642개의 단어에 대해 태깅이 잘못 된 것을 확인하였다. 이를 분석해본 결과, 오류를 표 4와 같이 4가지로 분류할 수 있었다. 전체 오류 중 나타난 빈도와 예시도 함께 표기하였다.

구분	예시
개체명의 경계 인식이 잘못된 경우(5.6%)	'20세기 폭스사'는 전체가 하나의 기관명인 개체명인데, 폭스사만 OG로 태그.
지역명과 기관명을 혼용한 경우(6.1%)	'홍대' 같은 경우는 대학교로서의 기관명일 수도 있고, 지역으로서의 홍대를 나타낼 수도 있다. '홍대'가 학습데이터 셋에서는 LC로 태그되어 있는데 OG로 태그.
데이터 셋의 오류로 인해 발생한 경우(9.8%)	'경찰'의 경우, 데이터 셋에서 'OG'라고 태그되어 있는 경우도 있고 'O'로 태그 되어 있는 경우도 있다.
그 외(78%)	'투수진'을 사람으로 태그 하는 등 개체명 태그 자체의 오류

표 4. 오류 분류 및 예시

3.2.1 데이터 셋 수정

데이터 셋의 오류로 인해 성능이 낮게 측정되는 경우가 예상보다 많아 테스트 데이터에서 명확하게 데이터 셋의 오류로 판단되는 사례를 수정하였으며 그 경우는 표 5와 같다. 수정한 데이터 셋을 이용하여 성능을 측정하여본 결과 86.3%을 기록하여 기존보다 1.6% 향상되었다. 수정된 전체 데이터는 총 41사례로 [13]에서 공개하였다.

구분	변경 사례
개체명 태그 누락된 경우 수정	전: '<LG:OG>는 <7일:DT> **잠실구장**에서 계속된' 후: '<LG:OG>는 <7일:DT> **<잠실구장:LC>**에서 계속된'
품사태그 수정	전: '태 NNP 어 NNP 난 NNP' 후: '태어나 VV ㄴ ETM'
잘못된 개체 인식	전: '비디오점 <체인 씨네타운:OG>이' 후: '비디오점 체인 <씨네타운:OG>이'

표 5. 테스트 데이터 노이즈 제거 작업 사례

4. 결론 및 향후 과제

본 논문에서는 한국어 개체명 인식의 성능 향상을 위해 기존의 BiLSTM을 이용한 영어 개체명 인식 시스템을 구현한 후 "하다" 접사의 동사형이 보존된 자체 워드 임베딩을 제작, 한국어 특징에 맞는 one-hot 벡터 추가를 통하여 성능이 더 향상되었음을 보였다. 하지만 개체명의 경계를 정확하게 찾지 못해 개체명을 잘못 태깅하는 경우가 있어 사전을 이용하여 개체명의 경계를 정확하게 찾는 시도를 하여보았으나, 오히려 성능을 낮추는 결과를 보였다. 개체명 사전을 이용하여 BiLSTM의 결과에 나온 개체명의 경계 인식 연구가 향후 진행된다면 한글 개체명 인식의 성능을 더욱 효과적으로 증대시킬 수 있을 것으로 보인다.

사사

본 연구는 과학기술정보통신부 및 정보통신기술진흥센터의 SW중심대학 지원사업의 연구결과로 수행되었음(2016-0-00018)

참고 문헌

[1] http://cs.nyu.edu/faculty/grishman/muc6.html

[2] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado, Jeffrey Dean, "Distributed Representations of Words and Phrases and their Compositionality", In Advances on Neural information Processing Systems, 2013.

[3] Guillaume Lample, Miguel Ballesteros, Sandeep Subramanian, Kazuya Kawakami, Chris Dyer, "Neural Architectures for Named Entity Recognition", 2016.

[4] Jason P.C. Chiu , Eric Nichols "Named Entity Recognition with Bidirectional LSTM-CNNs", 2015.

[5] 나승훈, 민진우, "문자 기반 LSTM CRF를 이용한 개 체명인식",한국컴퓨터종합학술대회논문집,2016.

[6] Vinayak Athavale, Shreenivas Bharadwaj, Monik Pamecha, Ameya Prabhu, Manish Shrivastava, "Towards Deep Learning in Hindi NER: An approach to tackle the Labelled Data Scarcity", 2016

[7] https://code.google.com/archive/p/word2vec

[8] 최윤수, 차정원, "Word Embedding 자질을 이용한 한국어 개체명 인식 및 분류", 2016

[9] https://www.tensorflow.org

[10] https://taku910.github.io/crfpp/

[11] Jeong-Uk Kim, "KoEL: Korean entity linking system using sentence features and extended entity relation", KAIST, Master Thesis, 2017

[12] https://sites.google.com/site/2016hclt

[13]https://github.com/machinereading/KoreanNERCorpus

● 구두발표 8: 의미분석

코어넷을 활용한 비지도 한국어 어의 중의성 해소

한기종○, 남상하, 김지성, 함영균, 최기선

한국과학기술원
{cumgi31, nam.sangha, jiseong, hahmyg, kschoi}@kaist.ac.kr

Unsupervised Korean Word Sense Disambiguation using CoreNet

Kijong Han○, Sangha Nam, Jiseong Kim, YoungGyun Hahm, Key-Sun Choi
KAIST

요 약

본 논문은 한국어 어휘 의미망인 코어넷(CoreNet)을 활용한 비지도학습 방식의 한국어 어의 중의성 해소(Word Sense Dsiambiguation)에 대한 연구이다. 어의 중의성 해소의 실질적인 응용을 위해서는 합리적인 수준으로 의미 후보를 나눌 필요성이 있다. 이를 위해 동형이의어와 코어넷의 개념체계를 활용하여 의미 후보를 나누어서 진행하였으며 이렇게 나눈 것이 실제 활용에서 의미가 있음을 실험을 통해 보였다. 접근 방식으로는 문맥 속에서 서로 영향을 미치는 어휘의 의미들을 동시에 고려하여 중의성 해소를 할 수 있도록 마코프랜덤필드와 의존구조 분석을 바탕으로 한 지식 기반 모델을 사용하였다. 이 과정에서도 코어넷의 개념체계를 활용하였다. 이 방식을 통해 임의의 모든 어휘에 대해 중의성 해소를 하도록 직접 구축한 데이터 셋에 대하여 80.9%의 정확도를 보였다.

주제어: 어의 중의성 해소, 코어넷, 마코프랜덤필드

1. 서론

어휘는 같은 형태에서도 다른 의미를 가질 수 있다. 예를 들면 '배' 라는 어휘는 '배를 타다' 와 '배를 먹다' 라는 두 문장에서 각각 교통수단 배와 과일 배의 의미로 사용된다. 어의 중의성 해소(Word Sense Disambiguation)는 이처럼 문맥 속에서 어휘가 사용된 의미를 파악하는 것이다. 어의 중의성 해소는 기계번역, 정보 추출 등 자연어처리의 여러 문제에서 활용할 수 있는 중요한 문제이다[1].

어의 중의성 해소에서 각 어휘의 의미를 선택하는 후보 대상으로 영어권에서는 어휘의미망인 Princeton WordNet(PWN)에 등재된 의미(Sense)를 기반으로 연구들이 진행되고 있다[1,2]. 본 연구에서는 한국어 어휘의미망인 코어넷(CoreNet)[3]에 등재된 의미를 기반으로 어의 중의성 해소를 진행한다. 코어넷에 대한 자세한 내용은 2장에서 다루었다.

PWN이나 코어넷은 의미가 세분화 된, 잘게 나눔(fine-grained)된 어휘의미망이다. 미묘한 차이를 가지는 의미 후보들이 있어서 사람도 특정 어휘가 어떤 의미로 사용된 것인지 선택하기 어려울 정도이다. 어의 중의성 해소의 실질적인 응용을 위해서는 합리적으로 의미 구분을 할 필요성이 있다[2]. 이에 영어에서는 반자동으로 PWN의 의미를 군집화하여 크게 나눔(coarse-grained)된 어의 중의성 해소를 진행한다[2]. 한국어에서는 동형이의어 수준에서 중의성 해소 연구가 진행된 바 있다[4].

어의 중의성 해소 접근 방식은 크게 지도학습(Supervised Learning)과 비지도학습(Unsupervised Learning) 방식이 있다. 지도학습은 의미가 태깅된 말뭉치를 학습하며 비교적 높은 성능을 보인다. 그러나 말뭉치를 구축하는 데에 많은 시간과 비용이 들며 학습 데이터가 없는 어휘에 대해서는 중의성 해소가 어렵다는 단점이 있다. 한국어에서는 의미가 태깅된 약 1000만 어절의 세종말뭉치 학습을 기반으로 하고 어휘의미망을 활용하여 96.5% 수준의 정확도를 나타낸 연구가 있다[4].

비지도 방식은 비교적 성능은 낮지만 학습 데이터가 없어도 가능하다. 이 방식에서는 어휘 의미망을 활용하는 지식기반의 알고리즘 연구가 성과를 내고 있다[1,5]. 이 연구들은 어휘의미망의 구조화된 정보를 활용해 그래프 형태의 의미 네트워크를 구성하여 가장 적합한 의미를 찾는 방식이다. 넓은 범위의 어휘를 커버하고 좋은 성능을 내고 있다[6]. 대표적인 최신 연구로 마코프랜덤필드(Markov Random Field) 모델을 기반으로 한 연구가 있다[1]. 문장을 형태소 분석과 의존구조 분석(Dependency parsing)을 통해서 마코프랜덤필드 모델로 구축한 뒤 해당 모델에 최대 사후 확률 추론(Maximum a posteriori inference)을 통해 문장 속의 모든 중의성 후보 대상 어휘의 의미를 찾는 방식이다. 이 모델을 통해 문장 속에서 서로 영향을 주는 어휘의 의미들을 동시에 고려하여 어의 중의성 해소를 할 수 있다.

본 논문은 한국어 어휘 의미망인 코어넷을 활용한 비지도학습 방식의 한국어 어의 중의성 해소에 대한 연구이다. 의미 후보의 크게 나눔을 위해 동형이의어와 코어넷의 개념체계를 활용하였다. 이에 대한 자세한 설명은 2장에서 다루었다. 비지도학습 접근 방식으로는 마코프랜덤필드를 바탕으로 한 지식 기반 모델[1]을 코어넷의 개념체계를 활용하여 한국어에 적용하였다. 이에 대한 내용은 3장에 서술하였다. 4장에서는 이 방식을 평가하기 위해 직접 구축한 데이터에 대하여 설명하였다. 5장에서는 코어넷의 개념체계를 활용하여 의미 후보 구분을 한 것이 실제 응용에서 의미가 있음을 보이는 실험과 본 논문의 접근 방식의 성능에 대해 서술하였다. 6장에서는 결론과 향후 연구에 관해 서술하였다.

2. 코어넷과 개념체계를 통한 어의 중의성 해소

2.1 코어넷

코어넷[3]은 한국어의 명사, 형용사, 동사의 의미와 관계를 나타내며 중국어, 일본어, 영어의 개념과도 연결된 다국어 어휘 의미망이다. 한국어 기본 어휘 체계의 약 90% 이상을 커버한다고 알려져 있다[7]. 총 7만 3000여개의 의미가 등재되어 있으며 각 의미의 정의문, 예문과 같은 부가적인 자원이 포함되어 있다. 각 의미들은 한글 학회 우리말 큰 사전의 동형이의어와 다의어 번호를 통해 구분되어 있다.

2.2 코어넷의 개념체계

코어넷의 각 의미는 좀 더 보편적인 의미를 가지는 코어넷의 개념체계와 연결되어있다. 이 개념체계는 일본 NTT '어휘의미속성체계'를 기반으로 한국어 특징에 맞게 227개를 확장하여 총 2,954개의 개념으로 이루어져 있다. 각 개념이 상/하위 관계를 맺고 있는 최대 깊이 12인 트리 형태의 계층적인 구조이다. 또한 각 개념은 동사, 명사, 형용사를 아우르는 품사 독립적인 체계이다. 이 개념을 통해서 중국어 일본어와 연결되어 있고, 영어 WordNet의 의미와도 연결되어 있다. 그 예시는 그림 1과 같다. 깊이 7인 '경쟁'이란 개념이 있고, 이 개념 아래에 동사인 '겨루다_0_1'과 명사인 '경기_12_1', '결승전_0_0' 등의 의미가 연결된 식이다. 각 의미에서 어휘 뒤에 첫번째 숫자는 동형이의어 번호(vocnum)를 나타내고 두번째 숫자는 다의어 번호(semnum)를 나타낸다.

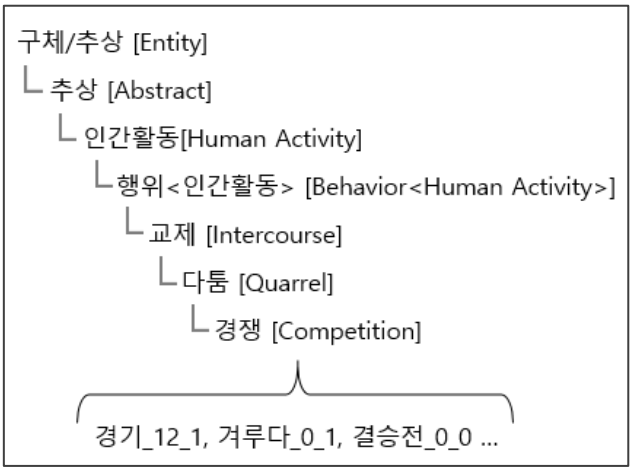


그림 1 코어넷의 개념체계 및 의미

2.3 어의 중의성 해소에 코어넷 개념체계 활용

어의 중의성 해소의 실질적인 응용을 위해서는 합리적인 수준에서 의미 구분을 할 필요성이 있다[2]. 본 연구에서는 분류 대상을 다른 한국어 연구처럼 동형이의어를 사용하되, 코어넷의 개념체계를 활용하였다. 코어넷 상에서 일반적으로 동형이의어 번호가 다른 의미들은 대부분 서로 다른 개념들에 연결되어 있다. 그러나 몇 가지 예외가 있다. '사과_1_0 : 사과참외의 준말', '사과_3_0 : 사과나무의 열매'는 서로 다른 동형이의어 번호를 가지지만 둘 다 '과일'이라는 개념에 연결되어 있다. 이런 경우 같은 후보로 군집화하여 진행하였다. 이렇게 같은 개념에 연결된 의미는 같은 후보로 판단하여, 후보 판별 시 개념을 활용할 수 있도록 하였다. 즉, '사과'라는 어휘에 대해서 표 1과 같은 식으로 의미 후보가 분류가 되고 각 후보에 대해서 적합성을 판단 할 때, '사죄', '과일' 등의 코어넷 개념체계를 활용할 수 있다.

이 결과 표 2와 같이 후보 대상 모호성이 평균 2.94개에서 2.66개로 줄어들었다. 모호성이란 전체 데이터의 의미 후보 개수의 총합을 전체 데이터 개수로 나눈 것이다. 4장에서 서술된 우리가 구축한 데이터에 대하여 측정하였다. 또한 이렇게 후보를 나는 것이 실제 활용에서 의미가 있음을 5장의 실험결과에서 보였다.

표 1 동형이의어와 코어넷 개념체계를 사용했을 때 '사과'의 의미 후보 분류 예시

후보	개념	(동형이의어, 다의어) 번호	정의문
0	사죄	(8,0)	- 잘못을 인정하고 용서를 빎
1	학문분야/학과	(6,1) (6,2)	- 도를 닦는 네 가지 과정 - 유학의 네 가지 학과
2	과일	(1,0) (3,0)	- '사과참외'의 준말 - 사과나무의 열매

표 2 후보 선택 대상별 선택 모호성

분류 대상	다의어	동형이의어	동형이의어+개념
모호성	5.00	2.94	2.66

3. 접근 방식

본 논문의 주 접근 방식은 3.2장에서 서술한 마코프랜덤필드 기반 방식이다. 다른 접근 방식으로 3.1장에 서술한 TF-IDF 벡터 유사도 방식을 활용하였다. 이 방식은 마코프랜덤필드 기반 방식의 모델을 구현하는 데 필요한 개념에 대한 빈도수 값을 구할 때 활용하였다. 이에 대한 자세한 내용은 3.2장에 서술하였다. 각 접근방식의 상세내용은 다음과 같다.

3.1 TF-IDF 벡터 유사도

각각의 후보 의미와 연결된 개념 아래에 있는 모든 의미의 정의문+예문의 TF-IDF 벡터와, 어의 중의성 해소를 하고자 하는 어휘가 포함된 문장의 TF-IDF 벡터의 코사인 유사도(Cosine Similarity)를 측정한다. 이때 가장 큰 값을 가지는 후보를 선택한다. 의미 후보 중 여러 개의 개념이 연결된 것이 있으면 해당 개념 중 하나만 맞추면 맞는 것으로 하였다.

예를들면 다음과 같다. '결승전 [경기]에서 연장전 끝에 한화가 이겼다.'라는 문장에서 '경기'라는 어휘의 중의성 해소를 할 때 '체육 운동으로 승부를 겨룸' 등의 뜻을 가지는 의미 후보가 있다. 이 의미 후보는 코어넷에서 그림 1과 같이 '경쟁'이라는 개념과 연결되어 있다. '경쟁' 개념 하위에 있는 '경기_12_1' 뿐만 아니라 '겨루다_0_1', '결승전_0_0' 등의 의미의 정

의문과 예문까지 활용하여 TF-IDF 벡터를 생성하고 입력 문장과 비교하는 방식이다.

TF-IDF 벡터의 차원은 전체 문서 집합에 나타난 서로 다른 어휘의 개수와 같고 벡터의 원소는 각각의 어휘에 대한 값을 나타낸다. 그 값은 다음과 같이 표시된다.

$$\text{tfidf}(t, d, D) = \text{tf}(t, d) \times idf(t, D)$$

D는 전체 문서 집합을 나타내고 본 연구에서는 한글학회 우리말 큰 사전에 등재된 각각의 의미 하나의 정의문과 예문을 문서 하나로 설정하였다. 단어 빈도 $\text{tf}(t, d)$는 어휘 t가 특정 문서 또는 문장 d에 나타난 회수이다. 역문서 빈도 $idf(t, D)$는 문서 집합 D 에 어휘 t가 출현한 문서의 수의 역수 값에 로그를 취한 값이다. 두 문장 TF-IDF 벡터의 코사인 유사도는 같은 어휘가 서로 많이 등장할수록 높아진다는 원리를 활용한 것이다.

3.2 마코프랜덤필드 기반 방식

마코프랜덤필드(Markov Random Field)란 확률변수들의 집합으로 이루어진 비방향성 그래프 모델이다. 그래프에서 정점은 각각의 확률변수를 나타낸다. 각 확률변수는 간선으로 연결된 다른 정점을 나타내는 확률변수에만 의존적이다. 이 모델은 자연어처리의 여러 문제를 해결하는 데 사용되고 있다[1,8]

본 논문에서는 어의 중의성 해소 문제를 마코프랜덤필드를 통해 접근한 [1] 연구를 코어넷의 개념체계를 활용하여 한국어에 적용하였다. 이 방식에서 문장 속의 중의성 후보 대상인 어휘를 해당 모델의 정점, 즉 확률변수로 택한다. 간선은 의존구조 분석결과 직접 연결된 두 어휘에만 생성한다. 이렇게 구성된 마코프랜덤필드에 최대 사후 확률 추론(Maximum a posterior inference)을 통해 결합 추론을 하여 문장 속에 모든 중의성 후보 대상 어휘의 적합한 의미를 결정한다. 자세한 원리와 방식은 다음과 같으며 [1]을 기반으로 한국어에 어떻게 적용하였는지를 설명하였다.

3.2.1 모델의 작동 원리

먼저 그래프 모델을 사용하여 문장 속의 모든 중의성 해소 대상 어휘에 대해 동시에 의미를 추론하는 이유는 다음과 같다. 어휘의 의미는 기본적으로 같은 문맥 속에 다른 어휘의 의미에 영향을 받는다.

'대상 수상'

이라는 문장에서 '대상'은 1) 최고 상, 2) 객관의 사물 등의 의미가 있고 '수상'은 a) 상을 받음, b) 내각의 우두머리 등의 의미가 있다. '대상' 이 1)의 의미로 쓰인 것을 알기 위해서는 '수상'이란 어휘만으로는 알 수 없고 '수상'이 a)의 의미로 쓰인 것을 알아야 파악할 수 있다. 따라서 문장 속 모든 어휘의 의미를 한 번에 결합 추론하는 그래프 모델이 사용되었다.

두 번째로, 문장 속에 모든 어휘가 서로 직접 영향을 미치지는 않는다. 그 예시는 그림 2와 같다. '배를 먹은 뒤 배가 아팠다' 라는 문장에서 첫 번째 '배' 라는 어휘에는 '먹다' 라는 어휘만 영향을 미치고 두번째 배는 '아프다' 라는 어휘만 영향을 미친다고 볼 수 있다. 영향을 미치는 어휘를 선택하는 방식에는 여러 가지가 있을 수 있지만 이 모델에서는 그림 2처럼 의존구조 분석 결과 직접 연결이 되어있는 어휘를 서로 영향을 미치

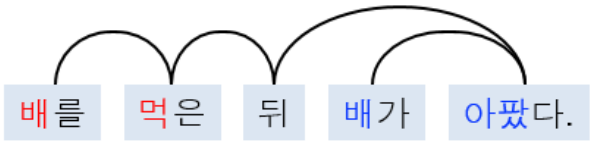


그림 2 문장의 의존구조 분석 결과

는 대상으로 선택하여 이 경우에만 마코프랜덤필드의 간선을 생성해 주었다.

3.2.2 모델 상세

먼저 문장이 주어지면 문장에서 모든 일반명사(NNG), 동사(VV), 형용사(VA) 중 코어넷에 등재된 어휘를 중의성 해소 대상 어휘로 선택한다. 이 과정에서 ETRI 언어분석기를 활용하였다. , 즉 마코프랜덤필드의 정점들을 $\mathbf{X} = \{x_1, x_2, \ldots, x_n\}$ 로 표현한다. x_i 값들은 각각 m_i 개의 개념 값을 가질 수 있다. x_i가 가질 수 있는 후보 개념들을 $s_i^1, s_i^2, \ldots, s_i^{m_i}$ 로 표현한다. 예를 들면 그림 2의 문장 같은 경우 x_1= '배' 가 가질 수 있는 개념을 나타내는 확률변수, x_2= '먹다' 가 가질 수 있는 개념을 나타내는 확률변수를 의미한다. x_1 의 후보 s_1^1=탈 것(본체(물)), s_1^2=과일 등이 되는 식이다.

이 방식도 3.1 방식과 마찬가지로 의미 후보가 여러 개의 개념과 연결되어 있으면 그 중 하나만 맞추면 알맞게 중의성 해소를 한 것으로 보았다.

이 마코프랜덤필드 모델의 포텐셜 함수 값으로 정점의 포텐셜 함수 $\psi(x_i)$ 와 간선의 포텐셜 함수 $\psi(x_i, x_j)$가 있다.

$$\psi(x_i = s_i^a) \propto \log(frequency(s_i^a) + e)$$

먼저 어휘 자체가 어떤 개념을 가질지 나타내는 정점의 포텐셜 함수는 위와 같다. $frequency(s_i^a)$ 는 해당 개념이 얼마나 자주 나타나는지를 의미한다. 이 값은 3.1장에서 서술한 TF-ID 유사도 방식에서 역치값을 0.14로 설정하여 우리가 구축한 데이터에 대해 정밀도(Precision) 0.951, 재현율(Recall) 0.287인 성능으로 위키피디아 전문의 약 10%에 해당하는 170만여 어절에 대해서 측정하였다. 0인 경우에 대비해 e 를 더하고 최종 모델에서 정점의 포텐셜이 너무 큰 영향을 미치는 것을 방지하기 위하여 log 를 적용하였다.

$$\psi(x_i = s_i^a, x_j = s_j^b) \propto Relatedness(s_i^a, s_j^b)$$

연관이 있는 두 어휘가 동시에 특정 개념들을 가질 경우를 나타내는 간선 포텐셜 함수는 위와 같이 두 개념간의 관련성에 비례한다. 이 모델에서 간선은 ETRI 언어분석기를 활용해 의존구조 분석결과 직접적으로 연결된 경우만 간선을 생성해 주었다. 이 간선 집합을 E 라 할 때 $\{x_i, x_j\} \in E$ 이다. 관련성은 아래 두가지 방식으로 측정하여 각각 실험하였다.

$$(1)\ Relatedness(s_i^a, s_j^b) = 1/(shortestpath(s_i^a, s_j^b) + 1)$$
$$(2)\ Relatedness(s_i^a, s_j^b) = \log(frequency(s_i^a, s_j^b) + e)$$

(1) 의 경우는 코어넷의 개념들간의 관계를 활용하여 두 개념간의 최단 경로의 역수를 취하였다. 이 방식은 [1]에서 사용한 방식과 같다.(2)의 경우는 $frequency(s_i^a)$ 를 구할 때와 같은 방식으로 두 개념이 한 문장에서 동시에 등장하는 횟수를 세었다.

최종적으로 이 모델의 포텐셜 함수는 다음과 같다.

$$\psi(\mathrm{X}) = \prod_{x_i \in X} \psi(x_i) \prod_{\{x_i, x_j\} \in E} \psi(x_i, x_j)$$

S를 각각의 어휘에 대해서 선택된 개념들 이라고 할 때, 위와 같은 포텐셜 함수를 가지는 모델에 아래와 같은 최대 사후 확률 추론을 통해서 중의성 해소를 한다. 이를 구현하기 위하여 [9]을 사용하였다.

$$\arg\max_S \psi(\mathrm{X} = \mathrm{S})$$

3.2.3 모델의 작동 예시

그림 3에 나타나 있는 문장은 우리의 데이터 중 하나이다. '자고산 서쪽 [사면]은 낙동강의 동쪽 강변과 닿아 있다' 라는 문장에서 '사면' 에 대해 중의성 해소를 하는 경우이다. 이 문장에서 '사면' 은 의존구조 분석결과 '서쪽' 과 '닿다' 에 영향을 받는다. '서쪽' 과 '닿다' 는 해당 모델에서 각각 '방위' 와 '연결' 이라는 개념을 가지는 의미 후보로 선택되었다. 이 때 사면의 의미 후보 중 '비스듬한 바닥' 이란 뜻의 후보는 '면(面)' 이라는 개념과 연결되어있다. 면(面) 이란 개념이 서쪽-'방위' 와 닿다-'연결' 이라는 개념과 가지는 관련성이 다른 후보들에 비해 높기 때문에 이 의미 후보가 선택되었고 알맞게 선택된 경우이다. 빈도수와 비례하는 정점 포테셜 값은 '죄를 용서하여 형별을 면제함' 의 뜻을 가진 후보가 높지만 관련성까지 고려하여 알맞은 후보를 선택되었다. 그림 3의 관련성 포텐셜 함수 값은 MRF co-occur 기준으로 기재되었다.

4. 데이터

먼저 위키피디아 알찬글 문서에서 임의로 문장을 추출하였다. 각 문장에서 일반명사, 동사, 형용사 중 하나이면서 코어넷에 등재된 어휘 중 임의로 하나를 선택하는 식으로 구축하였다. 이렇게 선택된 하나의 문장과 문장 속의 어휘로 이루어진 데이터에 대해서 3명의 사람이 알맞은 의미 후보를 선택하였다. 이 방식으로 총 470개의 데이터가 구축되었으며 구체적인 통계는 표 3과 같다.

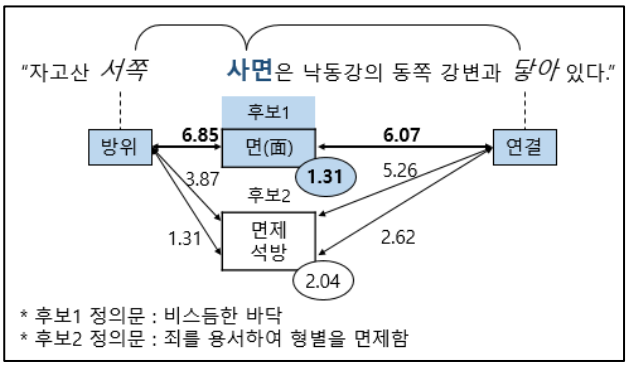


그림 3 마코프랜덤필드 모델의 작동 예시

구축한 데이터 중 중의성이 있는 어휘, 즉 후보가 2개 이상인 데이터는 총 215개 였다. 표 3에서 '# 어휘' 는 데이터에서 중의성 해소 대상이 되는 서로 다른 어휘의 개수를 의미한다. '# 의미' 는 데이터에서 어휘는 같아도 서로 다른 후보 의미로 쓰였으면 따로 세어서 계산한 것이다.

표 3 데이터 통계

	# 데이터	# 어휘	# 의미
모든 데이터	470	322	328
후보 2개 이상인 데이터	215	144	150

5. 실험 및 결과

5.1 성능

표 4 접근 방식별 정확도(Accuracy)

방식	RANDOM	TF-IDF	MRF SP	MRF co-occur
모든 데이터 성능	73.6	88.5	91.3	91.1
후보 2개 이상인 데이터 성능	42.2	74.9	80.9	80.5

우리가 구축한 데이터에 대하여 정확도를 측정한 결과는 위와 같다. 정확도는 다음과 같이 측정하였다.

$$\text{정확도} = (\text{시스템이 맞춘 개수})/(\text{데이터 개수}) * 100$$

RANDOM은 후보 중 임의로 하나를 선택하는 방식이며 5회 시행하여 평균을 내었다. TF-IDF는 3.1장에서 서술한 TF-IDF 벡터 유사도를 기반으로 한 방식이다. MRF SP와 MRF co-occur은 3.2장에서 서술한 마코프랜덤필드 기반의 방식이다. 차이는 두 개념간의 관련성 포텐셜함수 값을 다른 방식으로 구한 것인데, MRF SP는 3.2장의 (1)방식인 개념간의 최단 경로의 역수, MRF co-occur은 (2)방식인 두 개념의 동시 출현 빈도수로 설정한 경우이다.

마코프랜덤필드 기반 방식은 우리 데이터의 중의성이 있는 어휘에 대해서 80.9%의 정확도를 보였다. 임의로 선택한 베이스라인보다 많이 높은 정확도이고 정의문 및 예문의 유사도만을 비교한 TF-IDF 방식보다 약 6%의 성능향상을 보여주었다. 이를 통해 문장 속에 서로 영향받는 어휘의 의미들을 동시에 고려하여 추론하는 것이 의미가 있음을 보였다.

5.2 코어넷을 활용한 어의 중의성 해소 적용 결과

문장에서 관계추출(Relation Extraction)을 하는 문제에 본 연구의 어의 중의성 해소를 적용하였을 때 성능향상을 보였다. 이를 통해 코어넷의 개념체계를 활용하여 의미 구분을 한 후 어의 중의성 해소를 한 것이 실제 활용에서 의미가 있음을 알 수 있다.

구체적인 적용방식은 다음과 같다. Convolutional Neural Network 기반의 관계추출 기법[10]을 본 연구팀에서 한국어를 대상으로 구현한 모델에 어의 중의성 해소를 적용하였다. 관계추출 대상이 되는 문장을 토큰화한 후 각 토큰에 대한 임베딩 벡터가 이 모델의 입력으로 들어간다. 임베딩 벡터는 의미적 연관성이 높은 토큰들은 유사한 실수 벡터값으로 생성한다는 장점이 있다. 그런데 단어를 토큰화 할 때 형태소 단위로만 토큰화를 하면 중의성이 있는 단어들은 구분하지 못한다. 어의 중의성 해소를 통해 이 각각의 토큰들에 의미를 태깅하여 형태는 같아도 서로 다른 의미를 가지는 어휘가 임베딩 시 구분될 수 있도록 하였다.

표 5에서 형태소만으로 임베딩 하였을 때 '시장' 과 가까운 단어는 '물건을 사고 파는 장소' 라는 뜻과 관련된 단어들만 나오는 반면 의미를 태깅한 후에는 '시장' 의 두 가지 의미와 각각 관련된 단어들이 나오도록 임베딩이 된 것을 확인할 수 있다. 또한 표 6 에서 보듯이 의미 태깅 후에 임베딩을 한 것이 형태소 단위로만 토큰화 해서 임베딩 하는 것보다 관계 추출의 F1-score 성능이 약 7% 향상되었다.

표 5 임베딩 토큰 단위에 따른 유사한 토큰들

토큰 단위	토큰	유사한 토큰들
형태소	시장	투자, 유통, 수익, 수출, 자산, 대기업,
형태소 + 의미 태깅	시장-0 (물건을 사고 파는 장소)	시장, 산업-0, 업계-0, 경쟁력, 중소기업-0, 사업-4
	시장-1 (지방 자치 단체 장)	교육감-0, 기초자치단체장, 새누리당, 박순자, 고진화, 박영선_(1960년)

표 6 입력 임베딩 단위에 따른 관계추출 F1-score

임베딩 토큰 단위	형태소	형태소+의미 태깅
관계추출 F1-score	0.474	0.544

6. 결론 및 향후 연구

본 연구에서는 한국어 어휘 의미망인 코어넷의 개념체계를 활용하여 비지도학습 방식의 어의 중의성 해소를 진행하였다. 비지도학습이기 때문에 의미가 부착된 말뭉치 학습 데이터가 없어도 되며 마코프랜덤필드 모델과 의존구조 분석을 통해 문맥 속의 다른 어휘의 의미까지 고려하여 중의성 해소를 진행한다는 특징을 가지고 있다. 또한, 이러한 접근 방식과 의미 후보 분류에 있어서 보편적이며 계층적인 의미를 가지는 코어넷의 개념체계를 활용하였다는 데에서 의미가 있다. 이 방식을 통해 우리가 구축한 데이터에서 중의성이 있는 어휘에 대하여 80.9%의 정확도를 보였고, 코어넷의 개념체계를 활용하여 의미 후보 구분을 한 후 어의 중의성 해소를 하는 것이 실제 응용에서 의미가 있음을 보였다.

코어넷을 대상으로는 의미가 태깅된 말뭉치가 없어서 해당 모델의 함수 값 중 하나인 개념의 빈도수를 구할 때 TF-IDF 베이스라인 모델에서 재현율을 줄이고 정밀도를 높인 설정으로 구하였다. 정밀도를 0.951수준까지 높인 상태로 진행하긴 했지만 470개라는 비교적 적은 데이터에 대해서 측정된 것이고 좀 더 정확한 데이터를 활용할 필요성이 있다. 이와 관련해 한국어에는 약 1000만 어절 의미가 태깅된 세종말뭉치가 구축되어 있다. 이는 표준 국어 대사전의 의미 번호를 바탕으로 구성되어 있다. 반면 코어넷은 한글 학회 우리말 큰 사전의 의미 번호를 사용하여 의미가 구분되어 있다. 향후에는 코어넷에서 세종말뭉치를 잘 활용할 수 있는 방안을 연구하여 준지도학습(Semi-Supervised Learning) 등의 방향으로 성능향상을 이룰 수 있을 것으로 기대한다.

사사

이 논문은 2017년도 과학기술정보통신부의 재원으로 한국연구재단 바이오의료기술개발사업의 지원을 받아 수행된 연구임(NRF-2015M3A9A7029725)

참고문헌

[1] Chaplot, Devendra Singh, Pushpak Bhattacharyya, and Ashwin Paranjape. "Unsupervised Word Sense Disambiguation Using Markov Random Field and Dependency Parser." AAAI. 2015

[2] Navigli, Roberto, Kenneth C. Litkowski, and Orin Hargraves. "Semeval-2007 task 07: Coarse-grained english all-words task." Proceedings of the 4th International Workshop on Semantic Evaluations. Association for Computational Linguistics, 2007.

[3] Key-Sun Choi, et al. "Korean-Chinese-Japanese Multilingual Wordnet with Shared Semantic Hierarchy." LREC. 2004

[4] 신준철 and 옥철영. "한국어 어휘의미망 (UWordMap)을 이용한 동형이의어 분별 개선." 정보과학회논문지

제43권, 제1호, pp.71-79, 2016
[5] Agirre, Eneko, Oier Lopez de Lacalle, and Aitor Soroa. "Random walks for knowledge-based word sense disambiguation." Computational Linguistics 40.1, pp.57-84, 2014
[6] Iacobacci, Ignacio, Mohammad Taher Pilehvar, and Roberto Navigli. "Embeddings for Word Sense Disambiguation: An Evaluation Study." ACL (1). 2016.
[7] 한국과학기술원 전문용어언어공학연구센터, “다국어 어휘의미망 제1권: 어휘의미망 구축론”, KAIST Press, 2005.
[8] Jung, Sung-Young, et al. "Markov random field based English part-of-speech tagging system." Proceedings of the 16th conference on Computational linguistics-Volume 1. Association for Computational Linguistics, 1996.
[9] Ankan, Ankur, and Abinash Panda. "pgmpy: Probabilistic Graphical Models using Python." Proceedings of the 14th Python in Science Conference (SCIPY 2015). 2015.
[10] Zeng, D., et al. "Relation Classification via Convolutional Deep Neural Network." In Proceedings of COLING, pages 2335-2344. 2014.

Highway BiLSTM-CRFs 모델을 이용한 한국어 의미역 결정

배장성*O, 이창기*, 김현기+

강원대학교*, 한국전자통신연구원+

jseffort88@gmail.com, leeck@kangwon.ac.kr, hkk@etri.re.kr

Korean Semantic Role Labeling with Highway BiLSTM-CRFs

Jangseong Bae*O, Changki Lee*, Hyunki Kim+

Kangwon National University*, ETRI+

요 약

Long Short-Term Memory Recurrent Neural Network(LSTM RNN)는 순차 데이터 모델링에 적합한 딥러닝 모델이다. Bidirectional LSTM RNN(BiLSTM RNN)은 RNN의 그래디언트 소멸 문제(vanishing gradient problem)를 해결한 LSTM RNN을 입력 데이터의 양 방향에 적용시킨 것으로 입력 열의 모든 정보를 볼 수 있는 장점이 있어 자연어처리를 비롯한 다양한 분야에서 많이 사용되고 있다. Highway Network는 비선형 변환을 거치지 않은 입력 정보를 히든레이어에서 직접 사용할 수 있게 LSTM 유닛에 게이트를 추가한 딥러닝 모델이다. 본 논문에서는 Highway Network를 한국어 의미역 결정에 적용하여 기존 연구 보다 더 높은 성능을 얻을 수 있음을 보인다.

주제어: 자연어처리, 의미역 결정, 딥러닝, Highway Network

1. 서론

의미역은 서술어에 의해 기술되는 행동이나 상태에 대한 명사구의 의미 역할을 말하며 의미역이 부여된 각 명사구를 논항 이라고 한다. 의미역 결정은 각 서술어의 의미와 그 논항들의 의미역을 결정하여 “누가, 무엇을, 어떻게, 왜” 등의 의미 관계를 찾아내는 자연어처리의 한 응용이며 정보 추출, 질의 응답과 같은 다양한 자연어처리 시스템의 성능 향상을 위한 입력 정보로 사용될 수 있다. 예를 들어 의미역으로부터 시간 및 공간 정보, 사건의 주체와 같이 문장이 가지는 의미 등을 파악해 질의 응답 시스템이 필요로 하는 정보를 제공할 수 있다. 최근 의미역 결정 연구에는 Recurrent Neural Network(RNN)와 같은 딥러닝 모델을 이용한 연구가 주로 이루어 지고 있다[1,2,3].

딥러닝은 비선형의 히든레이어가 여러 층으로 쌓인 인공 신경망으로, 입력 자질들을 여러 비선형 변환기법의 조합을 통해 높은 수준의 표현으로 나타낼 수 있는 장점이 있다. Long Short-Term Memory Recurrent Neural Network(LSTM RNN)는 기존 RNN 모델의 그래디언트 소멸 문제(vanishing gradient problem)[4]를 해결한 딥러닝 모델이다. LSTM RNN은 음성 인식, 기계 번역, 자연어이해 등 다양한 분야에서 우수한 성능을 보이고 있으며[5,6], 순차 데이터(sequential data) 모델링에 적합한 구조로 이루어져 있다. Highway Network[7,8]는 비선형 변환을 거친 정보를 사용하는 LSTM 유닛에 비선형 변환을 거치지 않은 정보를 일부 선택하여 볼 수 있게 설계된 딥러닝 모델이다. 본 논문에서는 Highway Network를 한국어 의미역 결정에 적용하여 기존 연구보다 더 좋은 성능을 나타냄을 보인다.

본 논문의 구성은 다음과 같다. 2장에서는 관련연구를 소개하고, 3장에서는 한국어 의미역 결정 모델에 대해 설명하고, 4장에서는 실험 및 결과를 분석한다. 5장에서는 결론에 대해 기술한다.

2. 관련연구

최근 의미역 결정 연구는 딥러닝을 이용한 연구가 주로 이루어 지고 있다. [1]에서 사용한 Feed Forward Neural Network 모델은 출력 레이블을 결정하기 위해 현재 입력 단어를 포함한 고정된 크기의 입력 정보를 사용하는 단점이 있다. [2]의 연구는 Bidirectional Long Short-Term Memory Recurrent Neural Network(BiLSTM RNN) 모델을 이용한 연구로 LSTM 구조를 통해 멀리 떨어져 있는 단어의 정보를 유지할 수 있는 장점이 있고, Bidirectional 방법을 이용하여 문장에 나타나는 모든 단어의 정보를 사용하였다. [3]의 연구는 이미지 인식에서 뛰어난 성능을 보이고 있는 Convolutional Neural Network(CNN) 모델을 이용하여 의미역 결정에 사용되는 입력을 재구성하고, 재구성된 입력을 LSTM RNN의 입력으로 사용한다. [3]은 기존 단어 단위 연구가 아닌 알파벳 단위의 연구로서 입력 단어의 Out of vocabulary 문제에서 자유롭고 새로운 단어 표현을 얻는 장점이 있다. [7,8]은 비선형 변환을 거친 입력 정보를 사용하는 LSTM RNN 모델에 비선형 변환을 거치지 않은 정보를 사용할 수 있게끔 LSTM 유닛에 게이트를 추가한 딥러닝 모델이다. 본 논문에서는 [7,8]의 모델을 한국어 의미역 결정에 적용한다.

3. Highway BiLSTM-CRFs 모델

RNN은 순차 데이터 모델링에 적합한 형태로 디자인 되어 있으며 RNN을 입력 순서에 따라 언폴드(unfold)한 구조는 그림 1과 같다. 입력 단어 열 $x = \{x_1, x_2, \dots, x_T\}$와

히든레이어 유닛 열 $h=\{h_1,h_2,\dots,h_T\}$, 출력 단어 열을 $y=\{y_1,y_2,\dots,y_T\}$라 할 때 RNN은 식 (1)과 같이 정의된다.

$$h_t = f(UEx_t + Vh_{t-1} + b_h) \quad (1)$$
$$P(y_t|x) = y_t{}^T g(Wh_t + b_y)$$

U,W,V는 가중치 행렬이며 E는 단어 및 자질의 가중치 행렬이다. $f(z)$는 Sigmoid 혹은 Tanh 함수이고 $g(z)$는 Softmax 함수이다.

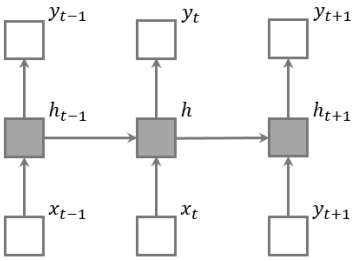


그림 1. RNN 모델

RNN은 입력 데이터의 열이 길어지면 신경망 구조가 깊어져 에러 전파(error propagation)가 어려워지는 그래디언트 소멸 문제가 발생하는데, 이 문제를 해결한 LSTM RNN은 식 (2)와 같이 정의된다.

$$i_t = \sigma(W_{ix}Ex_t + W_{ih}h_{t-1} + W_{ic}c_{t-1} + b_i) \quad (2)$$
$$f_t = \sigma(W_{fx}Ex_t + W_{fh}h_{t-1} + W_{fc}c_{t-1} + b_f)$$
$$c_t = f_t \odot c_{t-1} + i_t \odot \tanh(W_{cx}Ex_t + W_{ch}h_{t-1} + b_c)$$
$$o_t = \sigma(W_{ox}Ex_t + W_{oh}h_{t-1} + W_{oc}c_t + b_o)$$
$$h_t = o_t \odot \tanh(c_t)$$
$$P(y_t|\boldsymbol{x}) = y_t{}^T g(W_{yh}h_t + b_y)$$

위 식에서 σ는 Sigmoid 함수이고, $\odot$는 벡터 간의 element-wise product를 나타낸다. i,f,o,c는 각각 input gate, forget gate, output gate, memory cell 벡터 이며 각 벡터의 크기는 히든레이어의 벡터 크기와 같다. 가중치 행렬의 아래 첨자는 연결된 각 노드를 표시해 준다. 예를 들어 W_{hi}는 히든레이어와 input gate간의 가중치 행렬이다. 그림 2는 LSTM 유닛의 구조를 나타낸다.

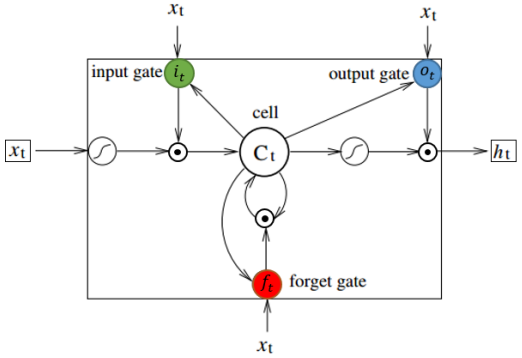


그림 2. LSTM 유닛 구조

그림 3은 LSTM RNN 구조를 나타내며 빨강, 파랑, 초록으로 표시된 부분이 LSTM RNN의 각 게이트(gate)를 나타낸다. LSTM의 입력은 4장의 표 1의 자질들이 projection layer를 거친 후 concatenate되어 하나의 벡터로 만들어지고, 만들어진 1개의 벡터가 LSTM의 입력으로 들어가게 된다.

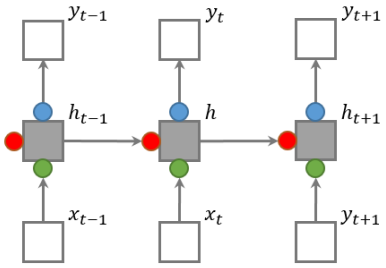


그림 3. LSTM RNN 모델

Highway Network는 비선형 변환을 거치지 않은 입력 정보를 LSTM 유닛이 사용할 수 있게 LSTM 유닛에 새로운 게이트를 추가한 모델이다. 추가된 게이트r_t와 변경된 h_t의 수식은 식 (3)과 같다.

$$r_t = \sigma(W_{rx}Ex_t + W_{rh}h_{t-1} + W_{rc}c_{t-1} + b_r) \quad (3)$$
$$h_t = r_t \odot o_t \odot \tanh(c_t) + (1 - r_t) \odot W_{hx}Ex_t$$

r_t는 비선형 변환을 거치지 않은 정보와 비선형 변환을 거친 정보를 얼마나 사용할지 정하는 역할을 하게 된다. 본 논문에서는 Highway Network를 BiLSTM RNN 모델에 적용하여 문장 전체의 정보를 사용한다. 또한 현재의 의미역 태그를 결정하기 위해 인접한 의미역 태그 정보를 활용하고자 한다. 이를 위해 출력 레이블의 인접성 정보를 바탕으로 현재 레이블을 추측할 수 있는 Conditional Random Field(CRFs)를 이용하여 output layer를 식 (4)와 같이 확장하였다.

$$s_{word}(y_t,t) = y_t{}^T(W_{yh}h_t + b_y) \quad (4)$$
$$s_{sent}(\boldsymbol{x},\boldsymbol{y}) = \sum_{t=1}^{T}\{[A]_{y_{t-1},y_t} + s_{word}(y_t,t)\}$$
$$log\,P(\boldsymbol{y}|\boldsymbol{x}) = s_{sent}(\boldsymbol{x},\boldsymbol{y}) - log\sum_{y'} exp(s_{sent}(\boldsymbol{x},\boldsymbol{y}'))$$

식 (4)에서 $[A]_{y_{t-1},y_t}$는 의미역 태그 y_{t-1}에서 y_t로 전이될 확률을 의미하고, $s_{sent}(\boldsymbol{x},\boldsymbol{y})$는 의미역 태그 열$y$의 점수이다. $log\,P(\boldsymbol{y}|\boldsymbol{x})$를 구하기 위해 forward 알고리즘을 이용하며, 최적의 태그 열을 구하기 위해 Viterbi search 알고리즘을 적용한다. 그림 4는 Highway BiLSTM-CRFs 모델을 나타낸다. **그림 4의 주황색 네모 상자는 비선형 변환을 거치지 않은 정보와 비선형 변환을 거친 정보를 얼마나 사용할지 정하는 r_t 게이트를 나타낸다.** 또한 모델의 출력 레이블 간의 의존성(전이확률)이 추가된 것을 알 수 있다. Highway BiLSTM-CRFs 모델의 학습을 위해 Stochastic Gradient Descent(SGD)를 이용하여 $-log\,p(y|x)$를 최소화 하였고, Back-Propagation Through Time(BPTT) 알고리즘을 이용하였다.

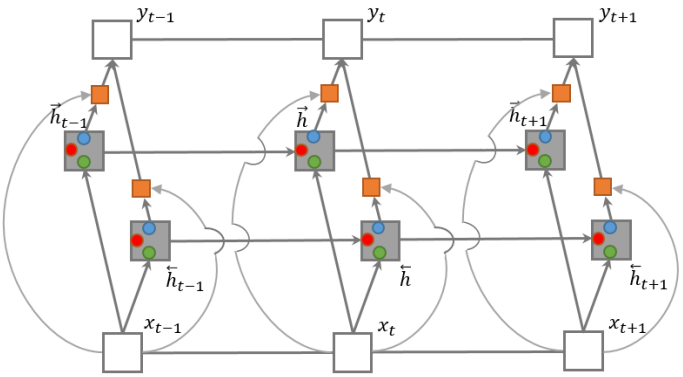


그림 4. Highway BiLSTM-CRFs 구조

4. 실험

본 논문에서는 Highway BiLSTM-CRFs 모델을 기존 연구와 비교하기 위하여 기존 연구에서 사용한 Korean PropBank[9]를 학습 말뭉치로 사용하였으며, 기존 연구와 동일한 학습데이터, 평가 데이터를 구성하였다. 표 1은 본 논문에서 사용하는 한국어 의미역 결정 자질과 그 예제를 나타낸다. 표 1의 우측 열은 문장 '한편 외국 자동차는 총 7만 2천 362대가 팔려 점유율이 39.8퍼센트로 떨어졌다.' 의 '372대가' 어절의 레이블을 결정할 때 사용되는 자질 예제이다.

표 1 한국어 의미역 결정 자질 예제

현재 어절에 포함된 형태소들의 어휘 및 품사 정보	362/SN, 가/JKS, SN, JKS, SN|NNB|JKS
현재 어절의 앞뒤 어절에 포함된 형태소들의 어휘, 품사 정보	2/SN, 팔리/VV, 천/NR, 어/EC, SN, VV, NR, EC, SN|NR, VV|EC,
서술어의 어휘 및 품사 정보	팔리/VV
서술어와 현재 어절의 위치 관계 및 거리 정보	PREV(위치 관계), DIST=1(텍스트 거리)

실험에 사용한 한국어 word embedding(단어 표현)은 word2vector[10]를 이용하여 구한 것을 사용하였다. feature embedding은 랜덤으로 초기화한 값을 사용하였고, 또한 Projection layer와 히든레이어에 Dropout[11]을 적용하였다. 성능 지표는 정확도와 재현율의 조화평균인 F1 지표를 사용하였으며, 본 논문에서 제시하고 있는 성능은 의미역 결정 문제의 논항 인식 및 분류(Argument Identification and Classification)에 해당한다. 평가의 단위는 어절 단위이며, Micro average를 사용한다.

표 2는 모델 별 한국어 의미역 결정 실험 결과이다. BiLSTM-CRFs 모델이 78.17의 성능을 보였고, Highway BiLSTM-CRFs 모델이 78.84로 BiLSTM-CRFs 모델보다 0.67% 더 높은 성능을 보였다. 앞선 두 모델의 성능차이를 통해 비선형 변환을 거치지 않은 입력 정보를 조절 하는 게이트가 성능 향상에 도움이 되는 것으로 유추할 수 있다. 추가적으로 각 모델에 히든레이어를 한 층 더 쌓아 실험을 진행하였는데, 실험 결과 Stacked BiLSTM-CRFs에서는 성능 향상이 있었으나 Stacked Highway BiLSTM-CRFs 모델에서는 경미한 성능 하락이 있었다. 또한 Highway Network와 유사하게 입력 정보를 가중치 연산 없이 히든레이어의 입력으로 사용하는 Residual Network[12]를 BiLSTM-CRFs에 적용하였으나, 실험 결과 가장 낮은 성능을 보였는데 Highway Network와 달리 가중치 연산이 없고, 입력 차원의 크기와 히든레이어의 크기가 같아야 하는 제한으로 인해 단어 정보 및 여러 자질 정보를 사용하는 의미역 결정에는 맞지 않는다고 생각한다.

표 2. 한국어 의미역 결정 실험 결과(AIC)

모델	F1
BiLSTM-CRFs (base)	78.17
Stacked BiLSTM-CRFs (2 layers)	78.57
Highway BiLSTM-CRFs	**78.84(+0.67)**
Stacked Highway BiLSTM-CRFs (2 layers)	78.77
Residual BiLSTM-CRFs	77.73(-0.44)

5. 결론

본 논문에서는 BiLSTM-CRFs 모델에 Highway Network를 적용하였고, 한국어 의미역 결정에서 기존 연구보다 좋은 성능을 얻었다. 이를 통해 비선형 변환을 거치지 않은 입력 정보가 한국어 의미역 결정 성능 향상에 도움이 됨을 알 수 있었다. 또한 Residual Network를 적용한 결과 입력 정보를 가중치 없이 히든레이어의 입력으로 사용하는 것이 한국어 의미역 결정의 성능 향상에는 도움이 되지 않는다는 것을 알 수 있었다.

감사의 글

본 연구는 미래창조과학부 및 정보통신기술진흥센터의 정보통신·방송 연구개발 사업의 일환으로 하였음. [2013-0-00131, (엑소브레인-1세부) 휴먼 지식증강 서비스를 위한 지능진화형 WiseQA 플랫폼 기술 개발]

참고문헌

[1] 배장성, 이창기, 임수종. 딥 러닝을 이용한 한국어 의미역 결정. 한국컴퓨터종합학술대회 논문집. 690-692. 2015.

[2] 배장성, 이창기. Stacked Bidirectional LSTM-CRFs를 이용한 한국어 의미역 결정, 정보과학회논문지, 제44권 제1호, 2017.

[3] Jie Zhou and Wei Xu, End-to-end learning of semantic role labeling using recurrent neural networks, Proceedings of the Annual Meeting of the Association for Computational Linguistics, pp. 1127-1137, 2015.

[4] YAO, Kaisheng, et al. Spoken language understanding using long short-term memory neural

networks. In: Spoken Language Technology Workshop (SLT), IEEE. 189-194. 2014.
[5] Sutskever, Ilya, Oriol Vinyals, and Quoc V. Le, "Sequence to sequence learning with neural networks", Advances in neural information processing systems, pp. 3104-3112, 2014.
[6] 박천음, 이창기. Bidirectional LSTM-CRF 모델을 이용한 멘션탐지, 한글 및 한국어 정보처리 학술대회, 2015.
[7] Yu Zhang, Guoguo Chen, Dong Yu, Kaisheng Yaco, Sanjeev Khudanpur, and James Glass. 2016. High- way long short-term memory rnns for distant speech recognition. In 2016 IEEE International Confer- ence on Acoustics, Speech and Signal Processing (ICASSP). pages 5755-5759.
[8] Luheng He, et al. Deep Semantic Role Labeling: What Works and What's Next. ACL 2016
[9] Martha Palmer, Shijong Ryu, Jinyoung Choi, Sinwon Yoon, and Yeongmi Jeon, Korean Propbank [Online]. Available: http://catalog.ldc.upenn.edu/LDC2006T03.
[10] Tomas Mikolov et al. Distributed Representations of Words and Phrases and their Compositionality. In Proceedings of NIPS, 2013.
[11] G.E Dahl, et al. Improving deep neural networks for LVCSR using rectified linear units and dropout. In:Acoustics, Speech and Signal Processing (ICASSP), International Conference on IEEE. p. 8609-8613. 2013.
[12] He, Kaiming, et al. Deep residual learning for image recognition. arXiv:1512.03385 (2015).

Layer Normalized LSTM CRF를 이용한 한국어 의미역 결정

박광현, 나승훈
전북대학교
khpark231@gmail.com, nash@jbnu.ac.kr

Layer Normalized LSTM CRFs for Korean Semantic Role Labeling

Kwang-Hyeon Park, Seung-Hoon Na
Chonbuk National University

요 약

딥러닝은 모델이 복잡해질수록 Train 시간이 오래 걸리는 작업이다. Layer Normalization은 Train 시간을 줄이고, layer를 정규화 함으로써 성능을 개선할 수 있는 방법이다. 본 논문에서는 한국어 의미역 결정을 위해 Layer Normalization이 적용 된 Bidirectional LSTM CRF 모델을 제안한다. 실험 결과, Layer Normalization이 적용 된 Bidirectional LSTM CRF 모델은 한국어 의미역 결정 논항 인식 및 분류(AIC)에서 성능을 개선시켰다.

1. 서 론

딥러닝이 뛰어난 성능을 보인다는 것은 증명이 되었지만, 여전히 모델이 복잡해질수록 Train 시간이 오래 걸리는 작업이다.

Train 속도를 높이는 방법에는 learning rate를 높이는 방법이 있지만, Gradient vanishing 혹은 Gradient exploding 문제를 야기한다.

learning rate의 값이 벗어나지 않을 정도로 크면 속도가 향상되기 때문에, Gradient vanishing 혹은 Gradient exploding 문제가 발생하지 않으면서 learning rate 값을 크게 설정할 수 있는 모델을 design 할 수 있도록 Internal covariate shift 개념을 이용한 Batch Normalization방법이 소개 되었다[1].

그러나 Mini-batch의 mean/variance를 이용하는 Batch Normalization의 효과는 mini-batch 크기에 따라 다르며, Train할 때 와 Test할때의 계산량이 변하는 문제가 발생하는데, 이를 해결한 방법으로 Layer Normalization 방법이 소개 되었다[2].

본 논문에서는 한국어 의미역 결정을 Layer Normalization이 적용 된 Bidirectional LSTM CRF를 이용해 성능이 개선 됨을 보인다.

2. 관련 연구

텍스트의 의미를 이해하는 것은 기계번역, 정보 추출, 정서 감지, 요약 등과 같은 많은 실제 응용 프로그램에서 중요한 역할을 한다. 의미론적 역할 레이블(SRL)은 각 동사의 의미론적 역할을 할당하는 중요한 NLP 작업이다. PropBank 및 FrameNet과 같은 리소스의 출현으로 SRL은 상당한 발전을 이루었지만, 여전히 SRL 시스템은 많은 작업과 사전 지식이 필요하였다. 이를 해결하기 위해 수작업으로 feature를 만드는 대신, 깊은 신경망을 이용해 자동으로 feature를 학습하는 방법에 대해서 많은 연구가 이루어 지고 있다[3-5]. 딥러닝 기법 중 순차 입력열에 특화되어 있는 LSTM기반 방법에 출력 노드간의 의존성을 모델링 하는 CRF를 결합한 방식인 LSTM CRF는 품사태깅, 개체명 인식 등에서 가장 우수한 성능을 보여주고 있다[8-9].

의미역 결정에서 FFNN(Feed Forward Neural Network)을 이용해 feature를 설계하는데 많은 시간과 노력이 들어가는 기계학습과 비교해 비슷한 성능을 보인다는 연구 결과를 보였고[3], [4]는 술어-논항 사이의 의존 관계 정보를 포함하고 있기 때문에 성능 향상에 큰 도움이 되는 구문 분석 정보 없이 Bidirectional LSTM CRF를 성능을 개선 시켰고, [5]는 Bidirectional LSTM으로 구성된 hidden layer를 한층 더 쌓은 Stacked Bidirectional LSTM-CRFs를 이용해 성능이 개선 됨을 보였다. 이 밖에, [6]은 동일 어휘임에도 분석 결과가 달라지는 경우를 해결하기 위해 의미역을 결정하는데 중요한 역할을 하는 술어의 의미 정보를 보다 명확하게 하기 위해 FrameNet의 의미 그룹 정보와 PropBank의 predicate senses 정보를 함께 사용하여 성능이 개선 됨을 보였고, [11]에서는 문장 구조가 달라도 동일 의미 정보를 지니는 경우가 있기 때문에 구문정보만으로는 한계가 있어 이를 해결 하기 위해 능동태와 수동태 정보, 자동사와 타동사 정보 등을 자질로 추가하여 성능이 개선 됨을 보였다. 또한, [7]에서는 서술어와 논항 사이의 dependency path를 이용해 성능이 개선 됨을 보였고, [12]에서는 의미 정보를 활용하는 방안으로 동형이의어 수준의 의미 애매성 해소, 고유 명사에 대한 개체명 인식 등의 정보를 사용하여 성능이 개선 됨을

보였다.

3. Layer Normalizaed LSTM CRF를 이용한 한국어 의미역 결정

그림 1은 본 논문에서 제안하는 Layer Normalized LSTM CRF기반 의미역 태깅의 뉴럴 모델을 도식화하여 보여준다.

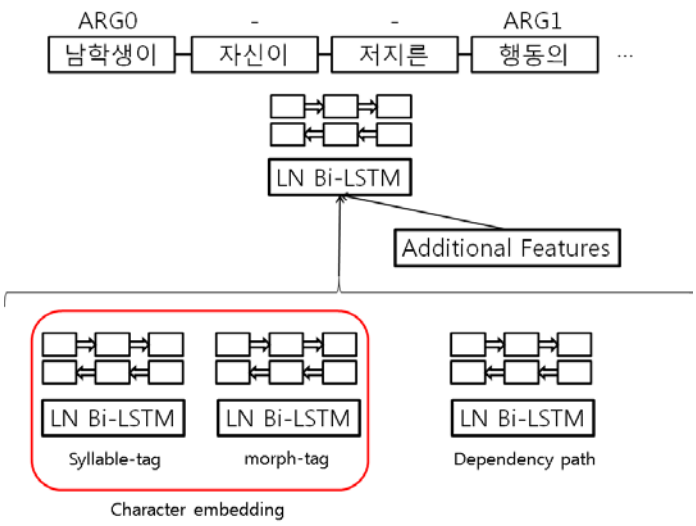


Figure 1. Layer Normalized LSTM CRF를 이용한 의미역 결정을 위한 뉴럴 모델 구조

그림 1에서 보다시피, 각 단어마다 LSTM의 입력표상 representation)이 정의되는데, 이 때 입력표상은 1) 문자 LSTM기반 단어표상 (word representation), 2) Dependency path, 3) additional features로 구성된다.

3.1 Batch Normalization & Layer Normalization

현재 layer의 입력은 이전 layer들의 변화에 영향을 받게 되는데 이전 layer의 파라미터 변화로 인해 현재 layer의 입력분포가 변하는 현상을 Covariate shift라고 한다. Covariate shift를 줄이는 방법 중 하나는 입력을 mean 0, variance 1로 바꿔주는 것(Whitening)이다. 하지만 전체 데이터를 기준으로 mean/variance를 학습시마다 계산하면 계산량이 많이 필요한데, 이때 나온 방법이 Batch Normalization 이다. Batch Normalization은 신경망에서 학습시 평균과 분산을 조정하여 Covariate shift를 줄이는 방법이다. 그림 2는 Batch Normalization 방법을 보여준다.

Input: mini-batch에 대한 입력값 x:
$B = \{x_{1...m}\}$; 및 학습 할 parameters γ, β
Output: $\{y_i = BN_{\gamma,\beta}(x_i)\}$

$\mu_B \leftarrow \frac{1}{m}\sum_{i=1}^{m} x_i$ //mini-batch mean

${\sigma_B}^2 \leftarrow \frac{1}{m}\sum_{i=1}^{m}(x_i - \mu_B)^2$ //mini-batch variance

$\hat{x_i} \leftarrow \frac{x_i - \mu_B}{\sqrt{{\sigma_B}^2 + \varepsilon}}$ //normalize

$y_i \leftarrow \gamma\hat{x_i} + \beta \equiv BN_{\gamma,\beta}(x_i)$ //scale 및 shift

Figure 2. Batch Normalizing Transform[1]

Batch Normalization은 전체 데이터에 대해 mean/variance를 계산하는 대신 지금 계산하고 있는 mini-batch에 대해서만 mean/variance를 구한 다음 inference를 할 때에는 data 전체에 대해서 mean/variance를 계산한 후, 정규화를 시키게 된다. 이 정규화가 입출력 값의 범위를 제한할 수 있기 때문에 linear transform을 사용하는데 이 transform에 있는 scale과 shift 파라미터 γ, β를 학습하면서 더욱 정교해지게 된다.

Batch Normalization을 사용함으로써, 더 큰 learning rate를 사용하여 학습속도를 향상시키고, covariate shift문제를 줄이고, 더 큰 weight가 더 작은 gradient를 유도하기 때문에 parameter growth가 안정화 되는 효과를 볼 수 있다. 하지만, batch size에 따라 Batch Normalization의 효과가 변하고, Train할때와 Test할때의 계산량이 다르다는 문제가 생기는데, 이를 해결한 방법이 Layer Normalization이다.

그림 3은 Layer Normalization 방법을 보여준다.

$$\mu = \frac{1}{H}\sum_{i=1}^{H} z_i \quad (1)$$

$$\sigma = \sqrt{\frac{1}{H}\sum_{i=1}^{H}(z_i - \mu)^2} \quad (2)$$

$$LN(\mathbf{z};\boldsymbol{\alpha},\boldsymbol{\beta}) = \frac{(\mathbf{z}-\mu)}{\sigma} \odot \boldsymbol{\alpha} + \boldsymbol{\beta} \quad (3)$$

$$\mathbf{f}_t = LN(\mathbf{W}_{xh_f}\mathbf{x}_t) + LN(\mathbf{W}_{hh_f}\mathbf{h}_{t-1}) + \mathrm{b}_{h_f} \quad (4)$$

$$\mathbf{i}_t = LN(\mathbf{W}_{xh_i}\mathbf{x}_t) + LN(\mathbf{W}_{hh_i}\mathbf{h}_{t-1}) + \mathrm{b}_{h_i} \quad (5)$$

$$\mathbf{o}_t = LN(\mathbf{W}_{xh_o}\mathbf{x}_t) + LN(\mathbf{W}_{hh_o}\mathbf{h}_{t-1}) + \mathrm{b}_{h_o} \quad (6)$$

$$\mathbf{g}_t = LN(\mathbf{W}_{xh_g}\mathbf{x}_t) + LN\left(\mathbf{W}_{hh_g}\mathbf{h}_{t-1}\right) + \mathrm{b}_{h_g} \quad (7)$$

$$\mathbf{c}_t = \sigma(\mathbf{f}_t) \odot \mathbf{c}_{t-1} + \sigma(\mathbf{i}_t) \odot \tanh(\mathbf{g}_t) \quad (8)$$

$$\mathbf{h}_t = \sigma(\mathbf{o}_t) \odot \tanh(\mathbf{c}_t) \quad (9)$$

위 식(1) 과 식(2)는 Layer Normalization에서 mean과 variance를 구하는 식이며 z_i 는 벡터 $\mathbf{z}$ 의 i 번째 원소를 나타낸다. Batch Normalization에서는 batch size인 m이 사용되었지만, Layer Normalization에서는 한 layer 에서의 hidden units수를 나타내는 H가 사용 되었다. Batch Normalization과 달리 Layer Normalization mini-batch 크기에 어떠한 제약도 받지 않는다. 또한, RNN에서 Batch Normalization을 적용하면 sequence 각 시간 단계에 대해 별도의 statistics를 계산하고 저장해야 하고, Test sequence가 Train sequence보다 긴 경우 문제가 발생하는데 Layer Normalization은 정규화 조건이 현재 시간 단계에서 Layer에 대한 합계 입력에만 의존하기 때문에 이러한 문제가 없다.

식(3)-(9)는 LSTM에서의 Layer Normalization 수식을 나타내는데, 식(3)에서 gain $\boldsymbol{\alpha}$와 bias $\boldsymbol{\beta}$는 scale과 shift를 위한 파라미터로, batch normalization과 마찬가지로 non-linearity 이전에 적용되며, $\boldsymbol{\alpha}$는 1, $\boldsymbol{\beta}$는 0으로 초기화 하였다.

$\odot$는 두 벡터 사이의 element-wise 곱셈을 나타내며, 식(4)-(8)에서 $\mathbf{W}_{hh}$ 는 hidden 과 hidden 사이의 가중치,

$\mathbf{W}_{xh}$ 는 x와 hidden 사이의 가중치를 나타내고 식(4)-(8)의 LN 함수에는 생략되었지만 각각 gains α_i 와 biases β_i 파라미터도 포함되어 있으며, 식(8)-(9)의 σ 는 sigmoid 함수를 의미한다.

3.2 입력

본 논문에서는 문자 LSTM기반 단어표상, dependency path, additional feature를 사용하였다. 문자 LSTM기반 단어표상으로는 형태소-태그 단위 문자, 음절-태그 단위 문자를 사용하였다. 형태소-태그 단위 문자는 단어와 품사태그를 합친 프랑스/NNP 형태로 구성 하였고 음절-태그 단위 문자는 띄어쓰기 정보를 활용하기 위해 프/B-NNP 랑/I-NNP 스/I-NNP 형태로 구성하였다. 이 문자들로부터 입력 단어 표상을 얻기 위해 형태소-태그 단위 문자 와 음절-태그 단위 문자 각각을 Bi-LSTM을 적용하여 마지막 상태 벡터를 결합한 후 MLP를 적용하여 입력 단어 표상을 얻어내었다.

Dependency path는 서술어-논항 사이의 dependency 관계로 "부시 검사는 남학생이 자신이 저지른 행동의 중대성을 인식하지 못하고 있는 것 같다고 말했다." 에서 서술어 "말했다." 와 논항 "같다고" 사이의 dependency path는 [quot, aux] 가 된다. Dependency path의 표상은 Bi-LSTM을 적용 하고, 마지막 상태 벡터와 입력 단어 표상과 결합한 뒤 MLP를 적용하여 사용하였다. 다음 그림 5는 dependency path의 예시를 보여준다.

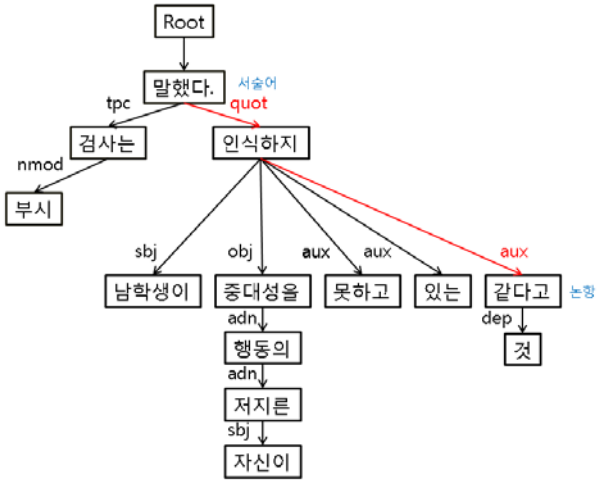


Figure 5. 말했다(서술어)와 같다고(논항) 사이의 dependency path

Additional feautre는 어절태그, 서술어의 어휘 및 품사정보, 서술어와 현재 어절 사이의 위치 정보, head의 위치 가 사용되었다. 다음 표 1은 additional feautre의 예시를 보여준다.

Table 1. Additional feature

Feature	Example
어절태그	남학생/NNG|이/JKS ➔ NNG~JKS
서술어의 어휘 및 품사정보	인식/NNG|하/XSV|지/EC ➔ 인식/NNC
서술어와 현재 어절 사이의 위치 정보	문장 "자신이 저지른 행동의 중대성을 인식하지" 에서 서술어 "인식하지" 와 현재 어절 "자신이" 의 위치 정보는 PREV 4 가 된다.
head의 위치	tree상의 parent(head가 문장내에 없으면 root로 표시)

3.2 출력

문자들의 임베딩 벡터(character embedding vector)와 자질 임베딩 벡터(feature embedding vector)를 결합하여 Layer Normalized Bidirectional LSTM을 적용하여 은닉 상태 벡터를 구하고, 출력층에 전달되어 각 태그별로 확률을 계산하게 되는데 출력층의 인접 노드들간의 의존성 모델링을 위해 CRF를 추가하였다.

4. 실험

4.1 실험 셋팅

본 논문에서는 의미역 결정 평가를 위해 Korean Propbank의 Newswire 말뭉치만을 사용하였고, Tree 구조의 말뭉치를 변환하는 도중 말뭉치에 오류가 있거나, 변환에 실패하는 문장은 제외 하였다.

전체 문장 23659 문장 중 20110 문장은 학습데이터로, 1183 문장은 개발셋, 2366 문장은 평가셋으로 사용하였다.

사용 된 의미역의 수는 None label 'O'를 포함하여 27개로 다음 표 2는 사용 된 의미역 태그를 나타낸다.

Table 2. 의미역 태그

labels	ARG0, ARG1, ARG2, ARG3, ARG5, ARGA, ARGM, ARGM-TMP, ARGM-LOC, ARGM-EXT, ARGM-CAU, AUX, ARGM-DIS, ARGM-INS, ARGM-MNR, ARGM-PRD, ARGM-ADV, ARGM-PRP, ARGM-CND, ARGM-DIR, ARG0-INS, ARGM-NEG, ARG0-DIS, AUX-DIS, ARG1-EXT, ARG0-TMP, O

4.1 실험 결과

Layer Normalization이 적용 된 LSTM CRF의 성능 향상을 알아보기 위해 LSTM CRF모델과 비교 하였다.

다음 표 3은 기존의 연구 결과와 LN Bi-LSTM CRF 모델의

실험 결과를 나타낸다. 표 3의 실험결과는 기존연구와 평가셋이 다르기 때문에 완전하지 않은 비교이다.

Table 3. 실험 결과 (AIC, F1)

	Dev	Test
Bi-LSTM CRF Model[4]		78.17%
Stacked Bidirectional LSTM-CRF Model[5]		78.57%
Our Bi-LSTM CRF Model	79.98%	77.86%
Our LN Bi-LSTM CRF Model	80.55%	78.46%

LN: Layer Normalization

표 4에 나온 결과는 F1으로 실험한 결과 중 제일 높게 나온 성능을 표기 하였다. Layer Normalization이 성능에 영향을 주는지 더욱 정밀하게 확인 해 보기 위해, 파라미터를 동일하게 하고 실험 한 5번의 결과의 평균을 낸 결과는 다음과 같다.

Table 4. 실험 결과 평균 (AIC, F1)

	Dev	Test
Bi-LSTM CRF Model	80.14%	77.57%
LN Bi-LSTM CRF Model	80.56%	78.10%

5. 결론

본 논문에서는 Layer Normalized LSTM CRF를 이용해 Layer Normalization이 의미역 결정 성능 향상에 도움이 되는지 실험 하였다. 실험 결과, 제안 방법을 이용한 Layer Normalized LSTM CRF모델은 한국어 의미역 결정 테스크상에서 성능이 개선됨을 보였다.

향후 연구로는 의미역 결정에 Attention mechanism을 적용하여 Attention mechanism이 의미역 결정에서 성능을 개선 시킬 수 있는지 평가 하고자 한다.

6. 참고문헌

[1] Sergey Ioffe, Christian Szegedy, Accelerating Deep Network Training by Reducing Internal Covariate Shift, 2015, arXiv

[2] Jimmy Lei Ba, Jamie Ryan Kiros, Geoffrey E. Hinton, Layer Normalization, 2016, arXiv

[3] 배장성, 이창기, 임수종, 딥 러닝을 이용한 한국어 의미역 결정, 2015, KCC

[4] 배장성, 이창기, Bidirectional LSTM CRF를 이용한 End-to-end 한국어 의미역 결정, 2015, KIISE

[5] 배장성, 이창기, Stacked Bidirectional LSTM-CRFs를 이용한 한국어 의미역 결정, 2017, KIISE

[6] 박태호, 차정원, 형태 의미 정보를 이용한 한국어 의미역 결정, 2017, KCC

[7] Michael Roth, Mirella Lapata, Neural Semantic Role Labeling with Dependency Path Embeddings, 2016, arXiv

[8] 나승훈, 민진우, 문자 기반 LSTM CRF를 이용한 개체명 인식, 2016, KCC

[9] 민진우, 오효정, 나승훈, 식품 도메인 개체명 인식을 위한 문자 기반 LSTM CRF, 2016, KCC

[10] 박광현, 나승훈, 문자 기반 LSTM CRF를 이용한 한국어 의미역 결정, 2017, KCC

[11] 박태호, 차정원, CRFs 기반의 한국어 의미역 부착 성능 향상을 위한 자질 선택, 2016, 정보과학회지

[12] 임수종, 임준호, 이충희, 김현기, 의미 프레임과 유의어 클러스터를 이용한 한국어 의미역 인식, 2016, 정보과학회논문지

SC-GRU encoder-decoder 모델을 이용한 자연어생성

김건영○, 이창기

강원대학교

uhi7074@gmail.com, leeck@kangwon.ac.kr

Natural Language Generation Using SC-GRU Encoder-Decoder Model

Geonyeong Kim○, Changki Lee

Kangwon National University

요 약

자연어 생성은 특정한 조건들을 만족하는 문장을 생성하는 연구로, 이러한 조건들은 주로 표와 같은 축약되고 구조화된 의미 표현으로 주어지며 사용자가 자연어로 생성된 문장을 받아야 하는 어떤 분야에서든 응용이 가능하다. 본 논문에서는 SC(Semantically Conditioned)-GRU기반 encoder-decoder모델을 이용한 자연어 생성 모델을 제안한다. 본 논문에서 제안한 모델이 SF Hotel 데이터에서는 0.8645 BLEU의 성능을, SF Restaurant 데이터에서는 0.7570 BLEU의 성능을 보였다.

주제어: 자연어생성, 딥러닝, 머신러닝, GRU

1. 서론

자연어 생성은 특정한 조건들을 만족하는 문장을 생성하는 연구이다. 이러한 조건들은 주로 표와 같은 축약되고 구조화된 의미 표현으로 주어지며 사용자가 자연어로 생성된 문장을 받아야 하는 어떤 분야에서든 응용이 가능하다.

의미 표현	inform(name='colibri mexican bistro'; type=restaurant)
출력	colibri mexican bistro is a nice restaurant

표 1. 자연어 생성 예제

표 1은 자연어 생성 예제를 보여준다. 위 행은 입력으로 들어가는 의미표현으로, inform은 출력의 대화 의도(dialogue act)이고 소괄호 안의 정보들은 출력이 가져야 하는 슬롯들을 의미한다.

전통적인 자연어 생성 방법은 방대한 양의 언어 표현 지식과 템플릿을 손으로 구축하여 규칙 기반에 의존하는 방식이었다. 그러나 기계학습이 발전하면서, 통계 기반 방법[6]을 거쳐 현재에 들어와서는 딥러닝 기반 방법[1,3,4]이 지배하고 있다.

본 논문에서는 SC(Semantically Conditioned)-LSTM[1]의 DA(Dialogue Act) cell을 GRU(Gated Recurrent Unit)[2]에 적용한다. DA cell은 GRU 내부 마지막 히든 레이어 갱신에 one-hot 벡터로 된 슬롯 정보를 추가로 주는 구조이다. 대화 의도는 그대로 1로 고정하고 슬롯은 출력이 생성될 때마다 레이어와 시그모이드 함수를 통하여 점차 1에서 0이 되도록 조정한다. 이 조정을 통제하기 위해 DA Cell 내부정보들을 이용한 목적함수를 사용한다.

전체적인 모델로는 encoder-decoder모델을 활용한다. 의미표현은 단어표현(word embedding)의 형태로 일반 GRU encoder로 들어가 압축(encoding)이 된다. 압축된 의미표현은 SC-GRU decoder로 들어가고, 이 decoder는 자연어로 된 출력을 생성한다. 출력들은 beam search 알고리즘에 의해 생성되며 잘못된 슬롯들이 등장하는걸 방지하기 위해 생성된 문장의 스코어와 슬롯 에러율을 더하여 re-ranking한다.

2. 관련 연구

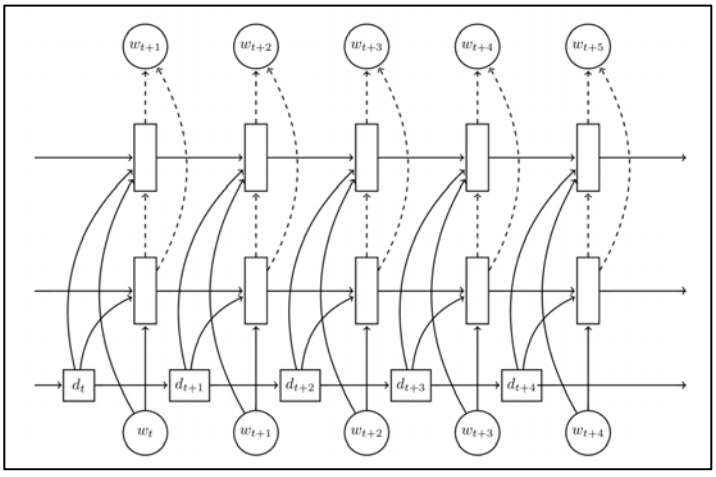


그림 1. Stacked SC-LSTM[1]

본 논문의 모태가 되는 [1]은 인공 신경망을 이용한 대표적인 자연어 생성 논문으로, SC-LSTM을 이용해 샌프란시스코에 있는 호텔과 레스토랑 데이터를 가지고 실험하였다.

그림 1은 [1]에서 사용한 모델로, 의미 표현을 one-hot 형태의 벡터로 주는 단일 decoder 모델이다. 본 논문에서 제안하는 모델은 일반 GRU를 이용한 encoder구조를 추가하여 의미 표현의 추상적인 압축을 가능하게 한 점에서 [1]의 SC-LSTM 모델과 다르다.

[3]은 encoder-decoder 기반 RA(Refinement Adjustment)-LSTM을 제안하고, [1]과 같은 데이터와 tv, laptop 도메인 데이터를 실험하여 현존 최고 성능을 내었다. 본 논문의 encoder와 달리 RA-LSTM에서 encoder는 슬롯들을 정, 역방향으로 encoding하고 decoder의 매 출력시간마다 attention mechanism을 사용하여 의미 표현을 압축한다. Decoder는 encoder에서 압축된 의미표현을 참고하여 이전 시간 출력을 재정의하고 다음 시간 입력으로 주는 refinement cell과 one-hot형태의 의미 표현을 LSTM의 출력에 더해주는 adjustment cell로 이루어져 있다. Attention mechanism과 refinement, adjustment cell을 통해 RA-LSTM은 추가적인 목적함수의 제약없이 매 시간 마다 one-hot형태의 의미 표현을 0에서 1로 조정할 수 있으며, 이는 본 논문과 구별된다.

[4]에서는 Wikipedia의 인물 페이지들이 항상 인물의 정보가 축약된 표를 가지고 있는 점을 이용하여, 표를 보고 본문의 첫 문장을 생성하는 연구를 시도하였다.

[5]는 [1], [4]에서 사용한 데이터를 GRU기반 sequence-to-sequence모델과 신경망 기계 번역에 쓰이는 여러 기법들을 써서 비교 실험하였다.

위와 다르게 [6]은 통계 기반 모델로, 의미 표현의 대화 의도와 슬롯 별로 클래스를 두었고, 클래스 마다 언어 모델링(language modeling)을 하여 특징을 잘 살린 출력을 생성할 수 있었다.

3. 데이터

의미 표현	inform SLOT_NAME hotel stratford SLOT_ADDRESS 242 powell street SLOT_PHONE 4153977080 SLOT_POSTCODE 94102 </s>
출력	the SLOT_NAME is located at SLOT_ADDRESS . , SLOT_POSTCODE . the phone number is SLOT_PHONE </s>

표 2. SF(San Francisco) Hotel 데이터

의미 표현	?select SLOT_NEAR russian hill or marina cow hollow </s>
출력	sorry would you like a restaurant near SLOT_NEAR </s>

표 3. SF Restaurant 데이터

본 논문에서 사용한 실험 데이터는 [1]에서 사용한 샌프란시스코 호텔, 레스토랑 관련 데이터이다. 이 데이터들은 표 1의 형태이지만 전처리를 하여 표 2, 3과 같은 형태로 만들고 탈 어휘화(delexicalize)를 통해 출력의 슬롯 값들은 슬롯 이름으로 대체하였다.

실제 decoding시에는 탈 어휘화된 문장을 생성하고 출력이 완성된 다음 후처리를 통해 슬롯 이름들을 슬롯 값으로 바꾼다.

[1]에서 실험한 SC-LSTM은 표 2, 3과 같은 입력이 사용되지 않고 슬롯 이름과 대화 의도만 one-hot형태로 decoder에 주어진다.

표 4는 실험에 쓰인 데이터의 의미 표현을 구성하는 정보들이다. Act type은 대화 의도이고 그 아래 칸들은 슬롯들이다. 이진 슬롯은 네, 아니오로 구성되며 앞에 별표가 오는 슬롯들은 상관 없음 값을 포함한다.

		SF Restaurant	SF Hotel
act type		inform, inform_only, reject, confirm, select, request, reqmore, goodbye	
shared		name, type, *pricerange, price, phone, address, postcode, *area, *near	
specific		*food *goodformeal ***kids-allowed**	***hasinternet** ***acceptscards** ***dogs-allowed**

bold=binary slots, *=slots can take "don't care" value

표 4. 의미 표현 구성 정보[1]

4. 모델

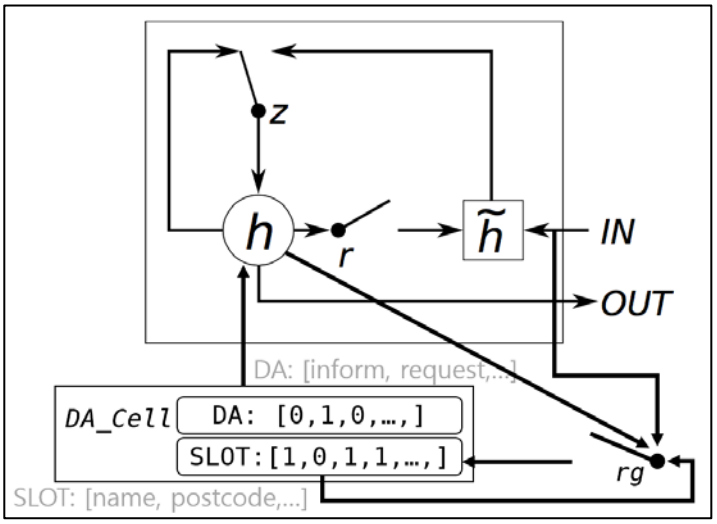


그림 2. SC-GRU 도식[2]

그림 2는 decoder로 쓰인 SC-GRU의 도식이다. 기존 GRU에 DA_Cell이 붙은 구조로 이전 시간 슬롯 이름들이 시간에 따라 reading gate(rg)에 의해 1에서 점차 0으로 바뀌는 구조다. SC-GRU의 자세한 수식은 아래와 같다.

$$(1)\ z_t = \sigma(W_{zx}x_t + U_z h_{t-1} + W_{zc}c + b_z)$$
$$(2)\ r_t = \sigma(W_{rx}x_t + U_r h_{t-1} + W_{rc}c + b_r)$$
$$(3)\ \hat{h_t} = \tanh(W_{hx}x_t + U_h(r_t \odot h_{t-1}) + W_{hc}c + b_h)$$
$$(4)\ rg_t = \sigma(W_{rgx}x_t + U_{rg1}h_{t-1} + W_{rgc}c + U_{rg2}SLOT_{t-1} + b_{rg})$$

(5) $SLOT_t = SLOT_{t-1} \odot rg_t$

(6) $h_t = (1 - z_t) \odot h_{t-1} + z_t \odot \widehat{h_t}$
$+\tanh(W_{hSLOT}[DA; SLOT_t] + b_{hSLOT})$

x 는 한 단어의 벡터 표현(word embedding)이고, Encoder는 일반 GRU를 사용하며, 의미표현 x를 입력으로 받는다. 의미 표현이 압축된 형태인 c는 의미 표현을 역방향으로 넣어 맨 마지막 GRU의 hidden state에 가중치를 곱하여 얻는다. SC-GRU 모델은 GRU로 이루어진 encoder를 사용함으로써 단순히 decoder에서 one-hot벡터로 된 의미 표현만 보는 것보다 높은 추상화가 가능해진다.

Decoder의 입력으로 이전 시간에 생성된 단어가 들어간다. DA와 $SLOT$은 모두 one-hot 형태의 벡터이다. DA는 대화의도로써 출력되는 문장이 일관된 의도를 갖게 하기 위해 1로 고정한다. $SLOT$은 의미 표현에서 등장하는 슬롯으로, 등장한 슬롯은 1로 나머지는 0으로 초기값을 주고 출력 시간마다 rg에 의해 값이 조절된다.

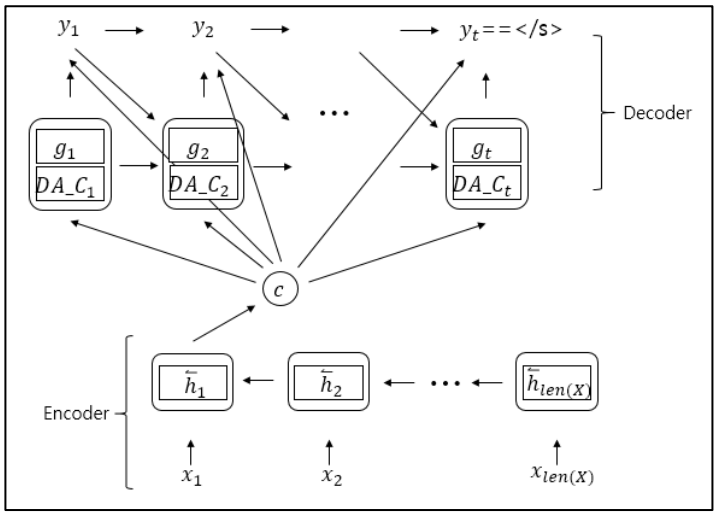


그림 3. SC-GRU encoder-decoder

그림 3은 자연어 생성을 위해 쓰인 SC-GRU encoder-decoder 모델의 전체 모습이다. [1]에서 사용한 SC-LSTM 모델과 다르게 encoder가 추가된 형태이다. Encoder의 일반 GRU를 통해 의미표현을 encoding한 c를 만들고, Decoder의 SC-GRU를 이용해 조건에 맞는 문장을 생성한다. 이때 decoder는 문장의 끝인 </s>를 만들면 멈춘다.

(7) $P(y_{t+1}|y_t, y_{t-1}, \dots, y_1, c, DA, SLOT_t) =$
$softmax(tanh(W_{yy}y_t + W_{yh}h_t + W_{yc}c + b_y))$

수식 (7)은 y_{t+1}의 확률 수식을 나타내고 있다. Decoder의 hidden state와 이전 시간 출력의 word embedding, 마지막으로 의미 표현의 압축형태인 c를 이용하여 출력을 결정한다.

(8) $\mathcal{L}(\theta) = -\sum_t R_t \log y_t$
$+ \|SLOT_T\| + \sum_{t=1}^{T-1} \eta\xi^{\|SLOT_{t+1} - SLOT_t\|}$

수식 (8)은 SC-GRU encoder-decoder 모델의 목적 함수이다. R은 실제 정답을 의미하며 1 혹은 0이다. 수식의 첫 항은 크로스 엔트로피로 $P(R)$과 $P(y)$의 차이를 줄여준다. 두 번째 항은 $SLOT$값이 꾸준히 줄어들게 해준다. 세 번째 항에서 η는 10^{-4}, ξ는 100의 값을 가지며, 이 세 번째 항은 시간에 따른 $SLOT$값의 변화를 최소화한다. 이러한 목적 함수의 정의로 SC-GRU는 $SLOT$의 값을 조절한다.

(9) $SLOT\_ERR = \frac{p+q}{n}$

(10) $\mathcal{L}(\theta)_{all} = \mathcal{L}(\theta) + \lambda \cdot SLOT\_ERR$

수식 (7)과 빔서치를 통해 문장을 생성한 후 여러 개의 문장 후보들을 슬롯 에러율을 이용하여 re-ranking한다. 수식 (9)는 슬롯 에러율의 정의를 보여준다. p는 입력에는 등장하지만 출력에는 없는 슬롯의 개수, q는 출력에는 등장하지만 입력에는 없는 슬롯의 개수, n은 입력에 등장하는 슬롯의 개수이다. 구해진 슬롯 에러율을 문장 스코어 $\mathcal{L}(\theta)$과 합하여 새로운 스코어 $\mathcal{L}(\theta)_{all}$를 구한다. 본 실험에서 λ는 0.5로 고정하였다.

5. 실험 및 결과

본 논문에서 제안한 모델을 기존의 모델들과 비교하기 위해 [1]의 저자가 github[7]에 올려놓은 성능 표와 성능 측정기를 활용하였다.

본 실험에서 사용한 하이퍼 파라미터는 표 5와 같다. 단어 표현(word embedding)은 google의 word2vec[8]을 사용해 사전 학습 시킨 후 사용했으며, 모든 히든 레이어의 차원을 일치시켜 학습하였다. Dropout[9]은 과적합(over-fitting)을 방지하는 기술로 입력, 출력, GRU 내부에 적용되었다.

하이퍼 파라미터	값(차원, 확률)
단어 표현 (word embedding)	200
히든 레이어	[30, 50, 80, 100, 120, 150]
Dropout[9]	[0.0, 0.2, 0.5]

표 5. 하이퍼 파라미터

표 6, 7은 성능을 나타낸다. 성능 측정에는 스코어가 가장 높은 문장 5개를 원문과 비교하는 Top-5 BLEU 스코어를 이용하였다. SC-GRU enc-dec가 본 논문에서 실험한 모델이다. 비교 실험에 사용된 기존 모델을 설명하자면, 의미표현을 보고 규칙 기반으로 문장을 생성한 모델이 hdc(handcrafted)이며, knn은 k-nearest neighborhood, N-gram은 [6]에서 실험한 class-LM이다. 그 아래부터는

신경망 모델의 성능으로, Enc-Dec는 DA_Cell이 들어가지 않은 LSTM기반 encoder-decoder모델이고 H-LSTM은 SC-LSTM과 비슷하나 목적 함수로 DA_Cell을 조절하지 않고 해당 슬롯이 출력에 등장하면 즉시 0으로 값을 바꿔버리는 휴리스틱한 방법을 쓴 모델이다.

모델	BLEU	SLOT_ERROR
HDC[1,7]	0.4260	0.00%
KNN[1,7]	0.5943	0.60%
N-gram[1,7]	0.6422	8.73%
Enc-Dec[1,7]	0.7398	2.78%
H-LSTM[1,7]	0.7466	0.74%
SC-LSTM[1,7] (baseline)	0.7525	**0.38%**
RA-LSTM[3]	**0.7789 (+0.0264)**	**0.16% (-0.22%)**
SC-GRU enc-dec(Our)	0.7570 (+0.0045)	3.51% (+3.13%)

표 6. SF Restaurant 성능

모델	BLEU	SLOT_ERROR
HDC[1,7]	0.5406	0.00%
KNN[1,7]	0.6745	1.75%
N-gram[1,7]	0.7700	5.87%
Enc-dec[1,7]	0.8549	4.69%
H-LSTM[1,7]	0.8504	2.67%
SC-LSTM[1,7] (baseline)	0.8482	3.07%
RA-LSTM[3]	**0.8981 (+0.0499)**	**0.43% (-2.64%)**
SC-GRU enc-dec(Our)	0.8645 (+0.0163)	2.12% (-0.95%)

표 7. SF Hotel 성능

SF Hotel의 경우, SC-GRU enc-dec모델이 슬롯 에러율도 줄고 BLEU값도 올랐으나, SF Restaurant의 경우 BLEU값만 상승이 있었고 슬롯 에러율은 높아졌다. 이는 word embedding과 GRU를 이용한 의미 표현의 추상적인 압축이 의미가 있음을 나타낸다. 다만 데이터에 따라 슬롯 에러율이 높아지는데 이는 식 (8)에서 의미 표현이 가지는 정보를 고려하지 않고 *SLOT*값만을 기준으로 제약을 주므로 생기는 한계라 볼 수 있다. 이러한 제약에서 벗어나 의미 표현만으로 *SLOT* 값을 조절하는 RA-LSTM은 BLEU값이 높음에도 불구하고 낮은 슬롯 에러율을 가진다.

데이터 셋에 따라 성능차이가 나는데 이는 KNN성능에서 그 이유를 찾을 수 있다. 데이터 내에서 가까운 거리에 있는 문장을 찾아 답을 주는 KNN방법은 데이터 내에서 같은 문장이 반복될수록 높은 성능을 보인다. 따라서 hotel데이터가 restaurant데이터에 비해 같은 문장이 반복되는 쉬운 데이터라 생각할 수 있다.

데이터	생성된 문장 (의미표현 / 생성문장)
SF Hotel	inform **SLOT_NAME** the carriage inn **SLOT_DOGSALLOWED** no </s>
	SLOT_NAME does not allow dogs </s>
SF Hotel	?request **SLOT_NEAR** </s>
	what area are you looking for </s>
SF Restaurant	inform **SLOT_NAME** taiwan restaurant **SLOT_PRICERANGE** cheap **SLOT_FOOD** taiwanese **SLOT_GOODFORMEAL** lunch </s>
	SLOT_NAME serves **SLOT_PRICERANGE** **SLOT_FOOD** food and is good for **SLOT_GOODFORMEAL** </s>
SF Restaurant	?select **SLOT_PRICERANGE** expensive or cheap </s>
	do you want a **SLOT_PRICERANGE** restaurant </s>

표 8. 생성 문장 예시

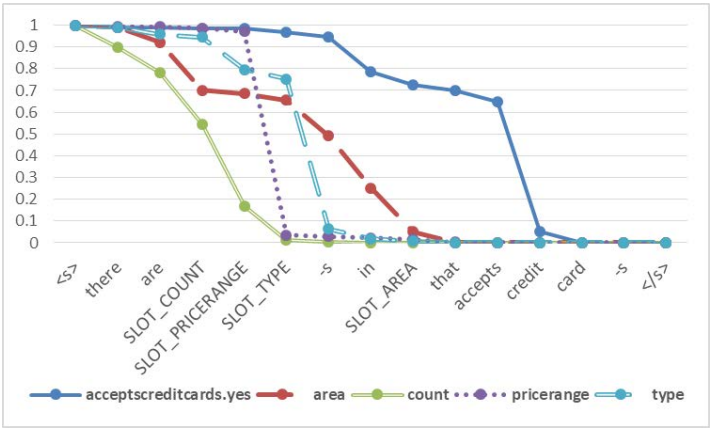


그림 4. SF Hotel의 DA_Cell 변화

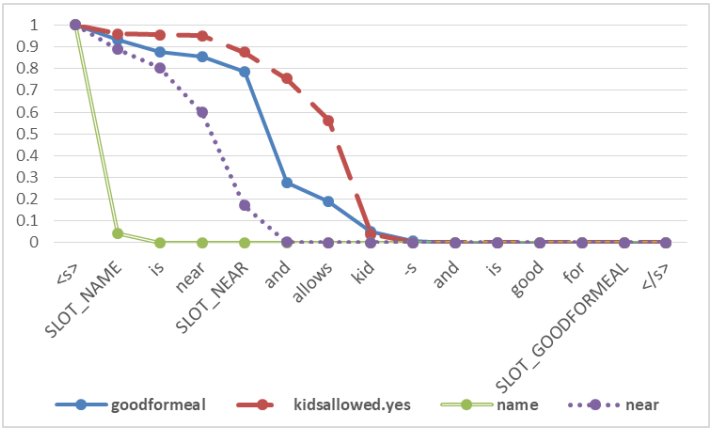


그림 5. SF Restaurant의 DA_Cell 변화

표 8은 생성된 문장의 예시를 보여주고, 그림 4와 5는 DA Cell 내부 값들의 변화를 보여준다. 그림 4의 경우 모든 출력에 대해서 DA_Cell이 잘 작동하는걸 볼 수 있다. 예를 들어 호텔이 카드를 받는다는 슬롯인 acceptcreditcards.yes 슬롯의 경우 맨 끝 card가 생성되기 전까지 값이 유지된다. 반면 레스토랑 데이터인 그림 5의 경우 다른 슬롯은 잘 동작하나 goodformeal 슬롯은 해당 슬롯이 생성되기 전에 값이 0으로 바뀌어 버렸다. 그러나 값이 0이 됨에도 불구하고 끝에서 생성이 된 걸로 보아 의미표현을 GRU로 압축한 값 c가 충분히 유의미함을 보여준다.

6. 결론

본 논문에서는 SC-GRU 기반 encoder-decoder를 이용한 자연어 생성 모델을 제안하였다. 실험결과, 본 논문에서 제안한 SC-GRU enc-dec모델이 베이스 라인보다 우수한 성능을 보였으며, One-hot 형태의 벡터와 성긴(dense) 형태의 벡터로 된 정보를 같이 사용하면 성능이 향상됨을 알 수 있었다. 그러나 현재 최고 성능을 보여주는 RA-LSTM에 비해 낮은 성능과 높은 슬롯 에러율을 가지는 점에서 의미를 고려하지 않은 의미 표현의 제약은 한계가 있음을 보였고 향후에는 이러한 문제를 해결하기 위해 연구할 예정이다.

감사의 글

이 논문은 2016년도 정부(미래창조과학부)의 재원으로 한국연구재단의 지원을 받아 수행된 기초연구사업임 (No. NRF-2016R1C1B1014124)

참고문헌

[1] Tsung-Hsien Wen, Milica Gasic, Nikola Mrksic, Pei-Hao Su, David Vandyke, Steve Young, “Semantically Conditioned LSTM-based Natural Language Generation for Spoken Dialogue Systems.” CoRR, Vol.abs/1508.01745, 2015.

[2] K. Cho, B. van Merrienboer, D. Bahdanau, and Y. Bengio. “On the properties of neural machine translation: Encoder-decoder approaches.” arXiv preprint arXiv:1409.1259, 2014.

[3] V.-K. Tran, L.-M. Nguyen, “Natural Language Generation for Spoken Dialogue System using RNN Encoder-Decoder Networks”, CoNLL 2017

[4] Remi Lebret, David Grangier, Michael Auli, “Neural Text Generation from Structured Data with Application to the Biography Domain.” CoRR, Vol.abs/1603.07771, 2016.

[5] 김건영, 이창기 “Sequence-to-sequence 모델을 이용한 자연어생성”, 한국정보과학회 학술발표논문집, Vol.2017, No.06, pp.624-626, 2017.

[6] Alice H. Oh and Alexander I. Rudnicky, “Stochastic language generation for spoken dialogue systems.”, Proceedings of the 2000 ANLP/NAACL Workshop on Conversational systems-Volume 3, pp.27-32, 2000.

[7] https://github.com/shawnwun/RNNLG

[8] Tomas Mikolov, Kai Chen, Greg Corrado, Jeffrey Dean, “Efficient Estimation of Word Representations in Vector Space”, CoRR, Vol.abs/1301.3781, 2013.

[9] Nitish Srivastava, Geoffrey Hinton and Alex Krizhevsky, Ilya Sutskever, Ruslan Salakhutdinov, “Dropout: A Simple Way to Prevent Neural Networks from Overfitting, Journal of Machine Learning Research, Vol.2014, pp.1929-1958, 2014”.

● 포스터 발표

구텐베르그 프로젝트 텍스트 데이터를 활용한 시각화 및 용례 검색

김동성, 신연수, 이지안, 유지민○
이화여대 인문대학
dsk202@ewha.ac.kr, dorn21@ewha.ac.kr, 1501299@ewhain.net, kekekele2007@ewhain.net

Text Visualization and Concordance Search Using Gutenberg Project Text Data

Dongsung Kim,Yeonsu Shin, Jian Lee, Jimin Yu○
College of Art & Science, Ewha Womans University

요 약

본 연구는 거시적 빅데이터 인문학과 미시적 언어 텍스트 검색 시스템을 구축하고, 이를 통해서 언어를 통한 문화의 역동적 변화를 시간적 순서에 따라 살펴보고자 한다. 연구의 최종적인 목표는 문화도 생물체처럼 변화하는 존재라 여기고 그 구성요소들을 연구한다는 뜻인 '문화체학(文化體學; Culturomics)'과 같은 '인문학 + 정보과학 + 사회과학' 등등의 다학문간의 융합적 연구에 있다. 이 시스템을 통해서 인류 역사의 기록인 텍스트 빅데이터를 통한 인문학적 성찰을 시각화하고 있다. 이러한 구글의 업적은 인문학과 정보기술의 융합을 통해서 인문학 자체의 지평을 넓히고, 사회과학을 변형시키고, 산업과 상아탑 사이의 관계를 재조정하는데 있다[1].

주제어: 구글 엔그램, 텍스트 시각화, 용례 검색, 구텐베르그 프로젝트

1. 서론

현재 빅데이터를 통한 언어 데이터의 시각화 그리고 문화의 역동적 변화에 대한 연구인 문화체학(文化體學; Culturomics)에 대한 접근은 현재 인문학과 융합된 연구들의 시대적 흐름이다. 특히 인문학 연구 관점에 볼 때 텍스트를 통한 문화 자체의 동향 및 추세(trend)를 포괄적으로 이해하는 것은 필수불가결한 요소이다.

본 연구는 거시적 빅데이터 인문학과 미시적 언어 텍스트 검색 시스템을 구축하고, 이를 통해서 언어를 통한 문화의 역동적 변화를 시간적 순서에 따라 살펴보고자 한다. 연구의 최종적인 목표는 문화도 생물체처럼 변화하는 존재라 여기고 그 구성요소들을 연구한다는 뜻인 '문화체학' 과 같은 '인문학 + 정보과학 + 사회과학' 등등의 다학문간의 융합적 연구에 있다.

2. 관련 연구

정보과학 분야에서는 인문학의 추상적 내용들은 정량화하는 여러 연구가 진행되고 있다. 그 중에서 특히 구글은 전 세계 많은 서적 텍스트를 디지털화하고 이를 기반으로 Ngram에 기반을 두고 텍스트 정보를 검색하는 Google Ngram Viewer[1] 서비스를 구축했다. 이 시스템을 통해서 인류 역사의 기록인 텍스트 빅데이터를 통한 인문학적 성찰을 시각화하고 있다. 이러한 구글의 업적은 인문학과 정보기술의 융합을 통해서 인문학 자체의 지평을 넓히고, 사회과학을 변형시키고, 산업과 상아탑 사이의 관계를 재조정하는데 있다[1]. [2]는 Google Ngram Viewer 서비스를 통한 텍스트 출현빈도에 기반을 두고 문화체학을 설명했다.

대용량 텍스트 용례 검색을 위한 시스템으로 [3], [4]는 IMS Corpus Workbench (이하 CWB)를 개발했다. CWB는 천만에서 20억 단어로 구성된 텍스트를 다양한 복잡한 조건의 검색조건에 맞춰서 검색할 수 있는 시스템이다.

이러한 시스템적 연구를 통해서 문화체학의 연구는 질적으로 국어를 대상으로 관련 시스템을 구축하고 이를 통해서 핵심어 연구[5], 특정 트렌드 연구[6], 개념어 연구[7] 등의 양적 연구를 통한 언어 내부적 표현의 변화등의 연구들로 발전하고 있다.

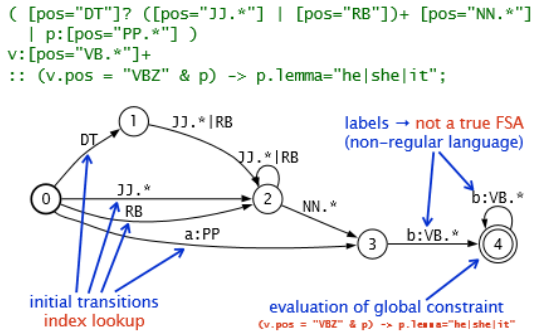
3. 시스템 구성도

CWB는 특정 구성 방식으로 텍스트 데이터를 구성하고 이를 바이너리(binary) 형식으로 변환하는 컴파일(compile) 과정을 거친다. 이러한 작업은 단어를 특정 인덱스로 변환해서 이를 활용한 효과적인 검색 방식으로 전환하기 위함이다. 특히 허프만 인코딩(Huffman Encoding)과 같이 효율적 검색 알고리즘도 구현되어 있다. 또한 인덱스를 바이너리 형식으로 전환해서 기계 친화적인 구조로 바꾸어 놓는다.

1) https://books.google.com/ngrams

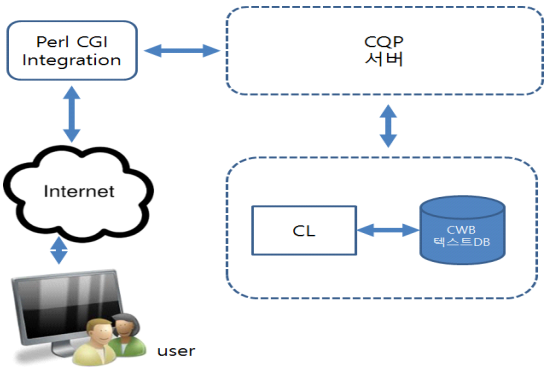
CWB의 장점은 크게 두 가지 인데, 하나는 대용량 텍스트 데이터의 효율적 처리이고, 다른 하나는 복잡한 검색 구조가 활용 가능하다는 것이다. CWB가 처리 가능한 데이터는 천만에서 20억 단어로 이루어진 대용량 텍스트이다. 또한 그림 1과 같이 정규 언어(Regular Language)가 아닌 복잡한 유한 상태 오토마타(Finite State Automata)도 처리가 가능하다.

그림 1 검색 FSA예 [4]



구축된 시스템 개요는 그림 2와 같다.

그림 2 시스템 개요도



전체 시스템은 CQP 서버에 의해서 구동되는데, CWB 형식으로 만들어진 데이터는 CL 모듈에 의해서 검색이 가능한다. 전체적으로 웹 인터페이스는 Perl 프로그래밍 언어인 Perl CGI로 구현된다.

4. 처리 과정

현재 시험 제작 원형으로 기 구축된 내용은 웹상에서 저작권이 없거나 무료인 서적들을 전자정보로 구축한 Gutenberg Project에서 전체 5만여 권 중 약 1%에 해당하는 500여권 소설 텍스트를 대상으로 용례 검색 및 연도별 시계열 그래프를 구성했다. 텍스트는 대략 5천만 어절로 구성되었으며, 텍스트에 대한 색인은 문장, 단어, 품사, 기본형이 가능하게 했다. 텍스트는 원시 코퍼스로 이를 자연어처리 시스템을 활용해서 품사, 문장 분리, 기본형 추출을 했다.[2)]

자료의 시대별 분포는 표 1과 같다.

표 1 연대별 소설 분포

연대	%
1950년대 이전	61
1950~1960	23
1970~1990	1
1990~2010	9
2010~2017	1

Gutenberg Project가 저작권이 무료이거나 없는 텍스트의 경우를 대상으로 하기 때문에 1950년대 이전 데이터가 주류를 이루고 있으며, 특히 1920년대 데이터가 가장 많다.[3)] 그러나 전체 텍스트를 대상으로 하면 더 다양한 연대의 텍스트가 수집될 것이다. 표 2와 같은 다양한 소설 장르들이 기 구축된 자료에 포함되어 있다.

표 2 소설 세부 장르별 구성

세부 장르	%	세부 장르	%
모험	2	학교	8
범죄	2	유머	9
탐정	7	영화	5
판타지	6	미스터리	0.4
고딕풍	1	과학	46
공포	5	서부극	8.8

그림 3과 같이 문장 구분을 <s>...</s> 태그로 구분하고 각 열은 텍스트에서 사용된 단어 자체, 단어의 품사, 단어의 기본형으로 구성했다. 각각은 검색은 단어, 품사, 기본형을 모두 조합해서 가능하게 했다.

그림 3 코퍼스 작성의 예

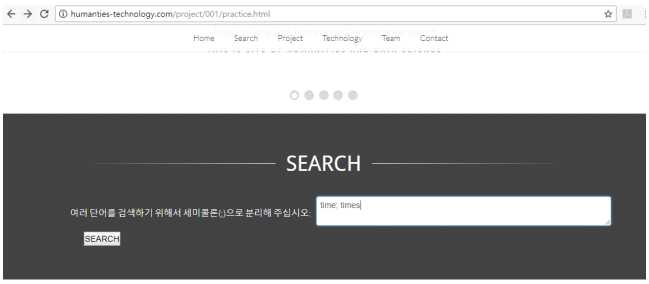
```
#     word        pos     lemma

0     A           DET     a
1     fine        ADJ     fine
2     example     NN      example
3     .           PUN     .

4     Very        ADV     very
5     fine        ADJ     fine
6     examples    NN      example
7     !           PUN     !
```

2) 사용된 자연언어 처리 시스템은 Stanford POS Tagger, NLTK sentence splitter, WordNet 등등이다.
3) 저작권 문제 등으로 인해서 1920년대 데이터에 집중되어 있다. 더 많은 데이터를 수집하면 이 문제는 해결되리라 생각된다.

5. 데모 서비스

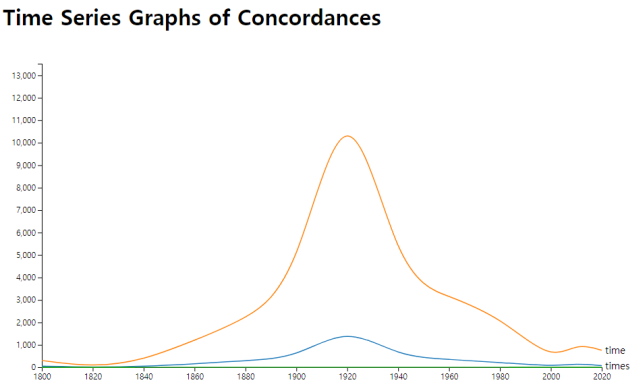
현재 그림 4와 같이 기 구축되어서 데모 웹사이트를 통해서 서비스고 있다.[4] 여러 개의 검색어를 ";(세미콜론)"으로 구분해서 입력할 수 있다. 그림 4는 "time, times"와 같이 두 개의 다른 단어를 입력한 것이다.

그림 4 두 개 검색어 입력



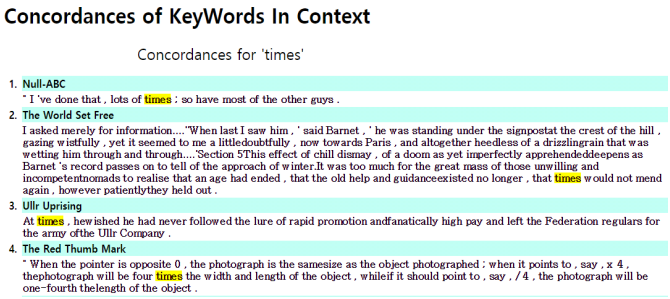
결과물은 텍스트 시각화와 용례 검색 화면에서 나탄다. 텍스트 시각화를 위해서 d3.js가 활용되었다.

그림 5 텍스트 시각화



현재 수집된 데이터는 1920년대에 집중되어 있기 때문에 1920년대에 많은 양의 데이터가 나타난다. 향후 영문 구텐베르그 프로젝트 텍스트 데이터가 활용되면 다양한 결과가 발견될 것이다. 그림 5에서 나타난 결과는 time이 times보다 10배 이상의 더 많은 용례가 발견되는 것을 보여준다.

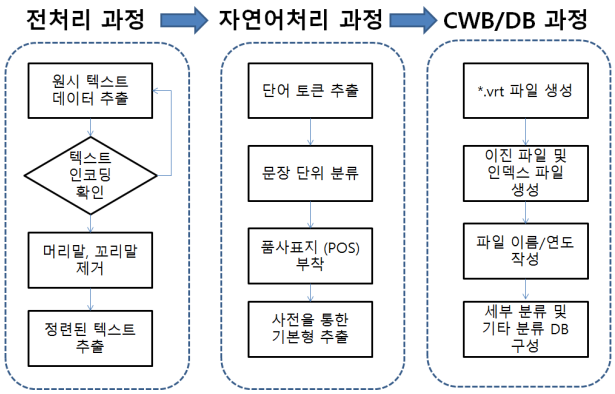
그림 6 times 용례 검색 결과



현재 데모 웹 서비스는 전체 용례 검색 결과를 보여주는 대신에 20개만 나타낸다. Php, MySQL 기반 웹 서비스가 CWB에서 개발되어 있기 때문에, 이를 적용하면 용례 결과 전체를 보여줄 것이다. 여러 용례 검색 결과 중 문장 단위 검색 결과만 보여준다. 문장 단위를 선택한 이유는 여러 검색 결과 중 가장 적절하다는 연구자들의 판단에서이다. 이 부분도 웹 서비스를 적용하면 더 다양한 검색 내용을 보여 줄 수도 있다.

Gutenberg Project에서 수집된 파일들을 정련하는 과정은 다음과 같다. 우선 텍스트 전처리(preprocess) 과정, 자연어처리 과정을 거치고, CWB 및 기타 파일 이름 및 연도, 세부 분류 DB를 구성하는 과정을 거쳐야 한다. 전처리 과정에서는 텍스트 인코딩이 무엇인지 확인하고 사용이 가능한지를 확인 작업해야 한다. 그리고 텍스트 내부의 머리말 및 꼬리말을 제거해야 한다. 파일의 메타데이터 및 법적 내용이므로 텍스트 자체와 연관성이 없는 것을 제거하기 위함이다. 다음으로는 자연처리 과정으로 단어 토큰을 분류하고 문장 단위로 텍스트가 구성되어 있지 않기 때문에 문장 단위로 텍스트를 분류한다. 이를 기반으로 품사표지를 부착하고 사전을 활용해서 기본형을 추출한다. 그림 7은 전체 작업 공정도이다.

그림 7 작업 공정도



4) http://humanties-technology.com/project/001/practice.html

6. 결론

이 연구는 언어 자료 구축과 연관되어서 구글 엔그램 뷰어와 유사한 구조인 텍스트 시각화를 보여준다. 구글 엔그램은 텍스트 용례 검색과 같이 미시적 텍스트 처리가 없는 반면에 본 시스템은 용례 검색도 가능하게 했다. 이를 위해서 CWB를 사용해서 전체 시스템을 구축했다.

향후 한국어 텍스트도 처리를 하기 위해서 노력하고 있으며, 한국어 인코딩 문제를 해결하고 있다. 또한 군집어 분석, 핵심어 추출과 같은 2단계 텍스트 분석 작업도 연구 중에 있다.

참고문헌

[1] 에이든·미셸 (2015) 빅데이터 인문학: 진격의 서막, 김재중 번역, 사계절.

[2] 문상호 (2015) 엔그램 뷰어를 이용한 인문학 빅데이터 사례 연구, Asia-pacific Journal of Multimedia Services Convergent with Art, Humanities, and Socialogy, 5(6), 57-65.

[3] Evert, S. and A. Hardie (2011) Twenty-first century corpus workbench. Proceedings of the Corpus Linguistics. Univ. of Birmingham, UK.

[4] Evert, S. (2008) Inside the IMS corpus workbench. Presentation at IULA, Univ. of Pompeu Fabra, Barcelona, Spain.

[5] 최재웅·김일환·홍정하·이도길 (2015) 핵심어로 본 시대상의 변화, 새국어생활, 26(4), 36-76.

[6] 조수곤·조재희·김성범 (2015) 텍스트마이닝을 활용한 대통령 취임 연설문의 트렌드 연구, 41(5), 435-460.

[7] 김영회·윤상길·최운호 (2011) 대한매일신보 국문논설의 언론 관련 개념 분석, 한국언론학보, 55(2), 77-102.

형식형태소가 한국어 단어 벡터 생성에 미치는 영향

윤준영○, 김도원, 민태홍, 이재성
충북대학교 소프트웨어학과
junyoung292@cbnu.ac.kr, dowon.0914@daum.net, mintaehong@cbnu.ac.kr, jasonlee@cbnu.ac.kr

Grammatical morphemes' effect on Korean word vector generation

Junyoung Youn○, Dowon Kim, Tae Hong Min, Jae Sung Lee
Dept. of Computer Science, Chungbuk National University

요 약

단어 벡터는 단어 사이의 관계를 벡터 연산으로 가능하게 할 뿐 아니라, 상위의 신경망 프로그램의 사전 학습 데이터로 많이 활용되고 있다. 한국어 어절은 생산적인 조사나 어미 때문에 효율적인 단어 벡터 생성이 어려워 대개 실질형태소만을 사용하여 한국어 단어 벡터를 생성한다. 본 논문에서는 실질형태소와 형식형태소를 모두 사용하되, 형식형태소를 적절하게 분류하여 단어 벡터의 성능을 높이는 방법을 제안한다. 자체 구축한 단어 관계 테스트 집합으로 추출 성능을 평가해 본 결과, 제안한 방법으로 형식형태소를 사용할 경우, 성능이 향상되었다.

주제어: 단어 벡터 생성, 신경망 프로그램, 한국어 어절 표현, 형식형태소

1. 서론

단어 벡터 (word vector)는 자연어의 단어를 다차원의 실수 벡터로 압축하여 표현한 것으로, 단어들의 특징을 잘 표현하여, 각 단어 사이의 여러 가지 관계를 벡터 연산으로도 찾아 낼 수 있다[1-4]. 예를 들어 의미적 관계인 <king> - <man> + <woman> = <queen> 이라든지 문법적 관계인 <write> - <wrote> + <eat> = <ate> 등의 관계를 계산할 수 있다. 또한, 이러한 벡터 표현은 신경망 프로그램의 사전학습(pre-training)의 결과로 사용되어 보다 복잡한 자연어처리, 기계번역, 개체명인식 프로그램 등의 입력 속성으로 활용되고 있다. 이에 따라 품질 좋은 단어 벡터의 개발이 다른 응용프로그램의 성능 향상에 중요한 요소가 되고 있다[5-9].

한국어는 형태소 발달 언어(morphological rich language)로서 띄어쓰기 단위가 어절이며, 영어 등에서의 띄어쓰기 단위인 단어와는 다르게 여러 형태소를 함께 포함하고 있어 비교적 복잡하다. 이런 복잡성 때문에 한국어 어절 벡터를 한 단위로 계산하려면 영어보다는 훨씬 더 많은 학습데이터가 필요하다[10]. 뿐만아니라, 한국어 언어처리 응용프로그램에서도 어절 단위가 아닌 형태소 단위로 처리하는 프로그램들이 많다. 이런 이유로 한국어에 대한 단어 벡터는 어절을 먼저 형태소 단위로 분리한 후, 이를 벡터로 표현한 형태소 벡터를 주로 사용해 왔다[11, 12].

본 논문에서는 이러한 형태소 벡터 특히 내용어에 해당되는 형태소 벡터들을 효율적으로 학습하기 위한 형식형태소의 역할을 분석한다. 이를 위해 한국어 어절을 실질형태소와 형식형태소로 나눈 후, 다양한 형태로 변형하여 학습데이터로 만들었다. 이를 Word2Vec[13]의 학습데이터로 사용하여 단어 벡터를 생성하고, 평가는 유추관계를 평가할 수 있는 100개의 단어 유추 관계 쌍(1쌍은 4개의 단어로 구성)을 새로 구축하여 수행하였다.

2. 단어 벡터의 생성 방법

단어를 벡터로 표현하는 방법으로는 가장 간단한 방법은 one-hot 표기 방법이다. 이 방법은 단어 개수만큼의 차원을 두고, 그 단어에 해당되는 차원만을 1로 표현하고 나머지는 0으로 하는 방법이다. 이 방법은 매우 큰 차원의 벡터가 필요할 뿐만아니라, 각 단어 사이가 모두 독립적으로 처리되어 그 벡터로는 단어의 비교가 불가능하다. 또 다른 방법은 단어를 축소하여(혹은 확대할 수도 있음) 각 차원을 실수로 표현한 것이다. 이 논문에서는 전자는 "one-hot 표기 벡터", 후자를 "단어 벡터"라고 칭한다.

단어 벡터의 학습은 여러 가지가 있을 수 있지만, 현재 Word2Vec[1-3],이나 GloVe[4]가 많이 쓰이고 있다. Word2Vec의 경우, 대상 단어를 매우 큰 차원의 one-hot 벡터로 표기하고, 그 단어의 문맥 단어들을 다시 one-hot 벡터의 합으로 표기한 후, 연산을 통해 그 벡터와 일치할 수 있도록 계산하는 과정에서 단어 벡터를 만들어 낸다. 그림 1은 Word2Vec의 CBOW모델로, 문맥 단어들의 one-hot벡터들을 더하여 신경망의 입력 벡터를 만들고, 변환 과정을 거친 후 다시 대상 단어의 one-hot 벡터를 생성해내는 과정을 나타낸다. 변환 과정은 입력 벡터에 대해 α행렬의 곱으로 선형변환 후 차원이 축소된 은닉층 벡터를 만들고, 이를 다시 β행렬의 곱으로

선형변환하는 과정을 나타낸다. 이를 수식으로 나타내면 (1)과 같다. 여기에서 w는 단어, C는 문맥을 나타낸다 [14].

$$\hat{y} = softmax_{\beta}(\sum_{w \in C} \alpha_w) \quad (1)$$

이 과정에서 만들어진 은닉층은 필요에 따라 여러 차원으로 만들어 낼 수 있으며, 대개 차원이 축소되어 표현되고, 이 벡터를 이용하여 단어 연산을 할 경우, 단어의 의미적 관계나 문법적 관계를 찾아 낼 수 있다[1-3].

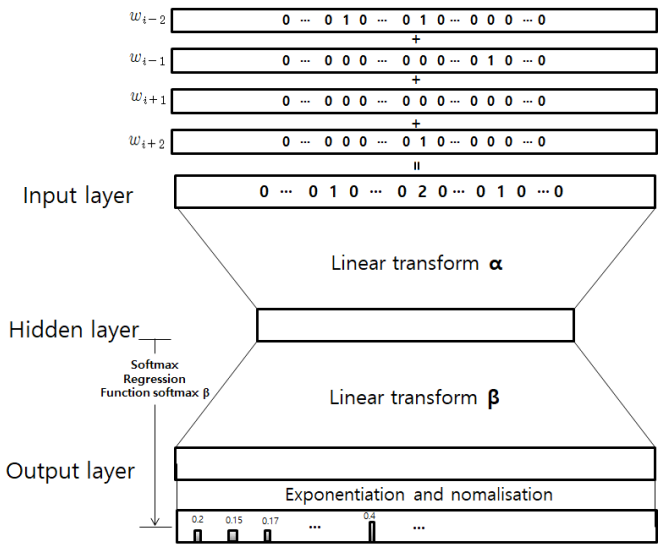


그림 1. Word2Vec의 CBOW모델 학습과정[14]

3. 한국어 단어 벡터 생성

앞 절에서 설명했듯이, Word2Vec의 입력으로는 one-hot 표기의 벡터가 사용된다. 영어의 경우, 비교적 단어 변형이 적어 충분히 많은 어휘에 대해 one-hot 벡터로 표현이 가능하고, 그 외 자주 나타나지 않는 단어에 대해서는 미등록어(unk)의 한 one-hot 벡터로 표기하여 처리한다[1-3].

한국어의 경우, 띄어쓰기 단위인 어절이 매우 생산적이어서, 어절의 종류가 영어 단어보다 훨씬 많아 큰 차원의 one-hot 벡터 표기가 필요하다. 이를 계산하는데는 많은 비용(메모리, 시간)이 들어 현재로서는 실용성이 적다. 따라서, 현재 대부분의 한국어 단어 벡터는 어절 단위보다는 형태소 단위로 분리한 후, 주로 실질형태소만을 취하여 벡터를 생성하고 있다.

하지만, 한국어에서 형식형태소는 단어의 의미를 명확히 해주는 역할을 하고 있어, 이를 반영하면, 더 정확한 실질형태소 기반의 단어 벡터를 생성할 것이다. 이 논문에서는 이를 검증하기 위해 형식형태소를 포함하여 단어 벡터를 생성하고, 그 효과를 측정한다. 한국어 어절을 형태소로 분리하고 띄어쓰기를 고려하여 다음과 같은 방법을 제안한다.

어절은 미리 형태소분석이 되어 있다고 가정하고, 형태소와 태그 정보를 이용하여 분리한다. 형태소는 실질형태소(어휘형태소, Lexical morpheme: Lmor)와 형식형태소(문법형태소, Grammatical morpheme: Gmor)의 두 그룹으로 처리하고, 그 두 그룹의 형태소에 해당되는 태그를 각각의 태그에 대해 실질형태소 태그(Ltag), 형식형태소 태그(Gtag)로 표시한다. 또, 형식형태소를 보다 분포 의미적으로 분석하기 위해 그 단어 벡터를 클러스터링한 번호(C(X))를 사용한다. 이 내용을 정리한 것이 표 1이다.

표 1. 어절 표기에 사용된 약칭

Lmor	실질형태소 (명사, 형용사, 동사)-부사제외
Gmor	형식형태소 (어미, 조사)
Ltag	실질형태소 품사태그
Gtag	형식형태소 품사태그
C(X)	X에 대한 단어 벡터 클러스터 번호
<bln>	띄어쓰기 표시

본 논문에서는 4가지 방법으로 어절을 표기하고 그 성능을 측정한다. 각 방법을 표 1의 표기법에 따라 예와 함께 표현하면 다음과 같다.

예: 나는 학교에서

Mbase: 실질형태소만으로 생성 (기준모델)
단위 형식: Lmor/Ltag
예: 나/np 학교/nng

Mall: 기준모델에 형식형태소 및 태그 포함
단위 형식: Lmor/Ltag Gmor/Gtag
예: 나/np 는/jx <bln> 학교/nng 에서/jkb

Mtag: 기준모델에 형식형태소 태그 포함
단위 형식: Lmor/Ltag Gtag
예: 나/np jx <bln> 학교/nng jkb

Mcls: 기준모델에 형식형태소 클러스터 번호 포함
단위 형식: Lmor/Ltag C(Gmor/Gtag)
예: 나/np C12 <bln> 학교/nng C20

Mcls에서는 형식형태소의 클러스터 번호를 이용하는데, 이는 Mall의 "형식형태소 및 태그" 보다는 덜 세밀하고, Mtag의 "형식형태소 태그" 보다는 좀 더 세밀한 정보를 포함하기 위한 것이다. Mcls의 클러스터는 초기의 연구로서 비교적 간단한 사전 학습을 통해 구하였다. 즉, Word2Vec의 학습데이터를 사전학습 모델인 다음

과 같은 Mpre 모델 형식으로 구성한 후, 클러스터를 구하였다.

Mpre: 실질 형태소 태그와 형식형태소 및 태그
단위 형식: Ltag Gmor/Gtag
예: np 는/jx <bln> nng 에서/jkb

이 클러스터 결과를 이용하여 Mcls 모델의 C(Gmor/Gtag) 번호를 생성한다.

3. 실험

3.1 실험 데이터 및 평가 함수

학습에 사용한 데이터는 세종 형태소 품사분석 말뭉치로 약 1000만 어절이다[15]. 전처리로 ETRI 형태소 분석기를 이용하여 미리 형태소 분석한 결과를 사용하였다. 평가를 위해 같은 유추 관계에 있는 단어 쌍 100개를 수집하여 평가 셋으로 사용하였다. 표2는 평가 데이터의 일부 예이다.

단어 벡터의 평가 방법은 주로 단어유사도 평가와 유추관계 평가가 사용되고 있다[10, 11]. 단어유사도 평가는 의미유사도, 단어관련도 등으로 세분화해서 평가하고, 유추관계는 의미유추와 문법유추의 두 가지로 세분화하여 평가한다. 본 논문에서는 평가 데이터를 새로 구축해야 하는 관계로 단어유추관계 평가만을 시행하였다[1]. 또, 이 논문은 형태소분석을 수행한 후, 실질형태소만을 평가하는 것이므로 문법유추는 제외하고 의미유추만을 사용하여 평가하였다.

의미유추 평가를 위해, 평가 셋에 나타난 단어 관계식(A-B+C=D)을 계산하고, 계산된 단어(D) 벡터와 가장 유사한 단어를 모든 단어 벡터와 비교하여 유사도 순으로 상위 10개를 나열하고, 정답이 나온 순위를 점수로 계산하였다. 즉, 각 단어 i에 대한 순위(0부터 9)를 $rank_i$라 하면 n개의 데이터에 대한 점수(score)는 다음 식(2)와 같다. 즉, 각 순위점수의 평균에 10을 곱한 것으로 최고점은 100점이 된다.

$$score = 10 \times \frac{1}{n} \times \sum_{i=1}^{n} (10 - rank_i) \qquad (2)$$

표 2. 평가 셋의 일부 예

A	B	C	D
원	한국	미국	**달러**
아들	남자	여자	**딸**
할머니	여자	남자	**할아버지**
병원	의사	경찰	**경찰서**
농구공	농구	야구	**야구공**
알파벳	미국	한국	**한글**
하늘	위	아래	**땅**
공자	유교	불교	**석가모니**
왕	남자	여자	**여왕**
서울	한국	일본	**도쿄**

3.2 실험 결과

실험을 위해 Word2Vec[13]의 하이퍼 파라미터를 조절하였으며, 윈도우 사이즈는 11, 최소 빈도 컷(cutoff)은 5에서 높은 성능을 보였으며, CBOW와 Skip-gram을 비교해 본 결과 Skip-gram이 우수하여 이 파라미터 값을 사용하였다. 또, 클러스터를 구하기 위한 사전 학습 모델인 Mpre 모델은 클러스터 크기를 200으로 하여 학습하였고, 이 결과를 Mcls 모델의 클러스터 번호 함수로 사용하였다.

실험 결과는 표 3과 같고, 이를 그래프로 표현한 것이 그림 1이다. 결과에서 보듯이 전체적인 성능은 40점에서 45점 사이로 다른 한국어 연구(유추관계 약 67%[10], 유사도 약 61점[11])에 비해 그리 높지 않은 편이다. 그 이유는 [10]은 약 6억7천여 단어, [11]는 약 5억7천여 단어를 사용하여 본 연구의 데이터보다 학습데이터가 더 크기 때문이며, 단어 벡터 생성 방법의 차이(GloVe[4], Skip-gram[1-3]), 유사도 계산 공식의 차이(cosine add, cosine multiply)[16] 등에서 기인한 것으로 판단된다. 하지만, 본 논문은 주로 실질형태소와 형식형태소의 역할을 상대적으로 비교하기 위한 것이므로, 다른 연구 결과와의 자세한 성능 비교 및 분석은 생략한다.

각 모델을 최고 성능을 보인 차원(표3의 굵은 글씨)을 중심으로 비교해 보면, 일반적으로 많이 사용하는 Mbase 모델을 기준으로 보아 Mall모델은 매우 성능이 낮았다. 이 Mall모델은 특별한 고려 없이 형식형태소를 형식형태소 태그와 함께 사용한 경우로 형식형태소의 추가가 오히려 성능을 하락시켰다. 형식형태소 태그만 추가한 Mtag모델은 Mbase와 거의 비슷한 성능을 보였다. 형식형태소를 적절히 클러스터링하여 추가한 Mcls 모델은 기준 모델 Mbase보다 성능이 향상되어 최고점에서 1.0점 더 좋은 성능을 보였다. 이는 적절하게 형식형태소를 분류하여 사용할 경우, 한국어 단어 벡터 생성에 효과적임을 보여준다.

1) 다른 연구에서 한국어 단어 유추 평가 데이터가 개발되어 있었지만, 실험 시 평가 데이터를 얻을 수 없어, 본 연구에서 새로 구축한 데이터를 사용하였다.

표 3. 각 차원에서의 모델의 성능

차원	100	200	300	400
Mbase	42.4	**44.0**	39.0	41.6
Mall	40.4	**42.1**	40.7	42.1
Mtag	**43.8**	42.5	42.2	41.1
Mcls	44.2	**45.0**	39.5	41.2

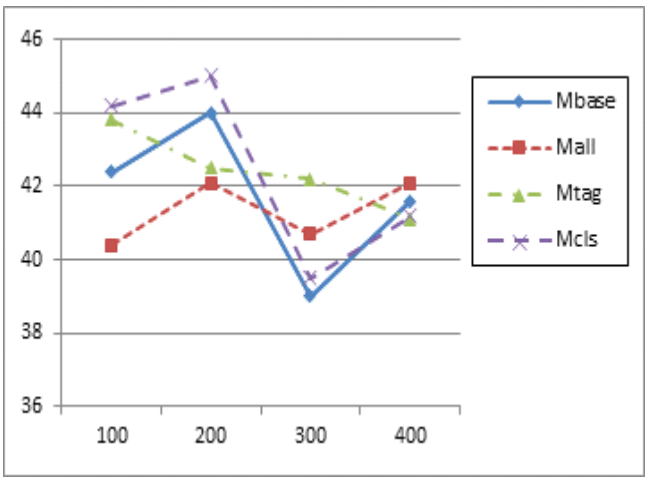


그림 2. 차원별 각 모델의 성능 변화

5. 결론

한국어 단어 벡터 생성 시 주로 한국어 어절 중 실질 형태소만을 분리하여 단어 벡터 학습에 사용한다. 본 논문에서는 한국어 단어 벡터 학습시 형식형태소를 추가하여 그 효과를 분석하였다. 즉, “형식형태소 및 태그”, “형식형태소 태그”, “형식형태소의 클러스터”를 각각 추가하여 학습하였고, 형식형태소를 적절히 클러스터링하여 그 정보를 사용할 경우, 단어 벡터의 품질이 높아짐을 알 수 있었다. 향후에는 충분히 큰 학습데이터와 평가 데이터를 구축하여 실험하고, 더 다양한 계산 방식으로 이를 평가하여, 품질 좋은 한국어 단어 벡터를 생성하기 위한 방법을 연구할 계획이다.

사사(Acknowledgement)

이 논문은 2017년도 정부(교육부)의 재원으로 한국연구재단의 지원을 받아 수행된 기초연구사업임(No. 2017R1D1A3B03035676)

참고문헌

[1] Mikolov, Tomas, Kai Chen, Greg Corrado, and Jeffrey Dean, "Efficient estimation of word representations in vector space," ICLR workshop, 2013.

[2] Mikolov, Tomas, Wen-tau Yih, and Geoffrey Zweig, "Linguistic regularities in continuous space word representations," In Proceedings of HLT-NAACL, 2013.

[3] Mikolov, Tomas and J. Dean, “Distributed representations of words and phrases and their compositionality.” In proceedings of NIPS, 2013.

[4] Pennington, Jeffrey, Richard Socher, and Christopher D. Manning, “GloVe: Global vectors for word representation,” In Proceedings of the Conference on Empirical Methods in Natural Language Processing, pages 1532-1543, 2014.

[5] Bahdanau, Dzmitry, Kyunghyun Cho, and Yoshua Bengio, “Neural machine translation by jointly learning to align and translate,” arXiv preprint arXiv:1409.0473, 2014.

[6] Luong, Minh-Thang, Hieu Pham, and Christopher D. Manning, “Effective approaches to attention-based neural machine translation,” arXiv preprint arXiv:1508.04025, 2015.

[7] Mikolov, Tomas, Quoc V. Le, and Ilya Sutskever. “Exploiting similarities among languages for machine translation.” arXiv preprint arXiv:1309.4168, 2013.

[8] Rush, Alexander M., Sumit Chopra, and Jason Weston, “A neural attention model for abstractive sentence summarization,” arXiv preprint arXiv:1509.00685, 2015.

[9] Siencnik, Scharolta Katharina, “Adapting word2vec to named entity recognition,” In proceedings of NODALIDA 2015, Vilnius, Lithuania, no. 109, pp. 239-243, 2015.

[10] Yang, Hejung, Young-In Lee, Hyun-jung Lee, Sook Whan Cho, and Myoung-Wan Koo, “A Study on Word Vector Models for Representing Korean Semantic Information,” In proceedings of KSSS, Vol 7, no. 4, 2015.

[11] 최상혁, 설진석, 이상구, “한국어에 적합한 단어 임베딩 모델 및 파라미터 튜닝에 관한 연구”, 제28회 한글 및 한국어 정보처리 학술대회 논문집, 2016.

[12] Kang, Myung Yun, Bogyum Kim, and Jae Sung Lee, “Word Sense Disambiguation Using Embedded Word Space,” Journal of Computing Science and Engineering, Vol 11, no. 1, 2017.

[13] Word2Vec web page, https://code.google.com/p/word2vec/

[14] Wilson, Benjamin, “An overview of word2vec,” presentation file, Berlin ML Meetup, 2014.

[15] 국립국어원, 21세기 세종계획 최종성과물 (2011년 12월 수정판), 2011.

[16] Levy, Omer and Yoav Goldberg, “Linguistic

regularities in sparse and explicit word representations," In proceedings of the eighteenth conference on computational natural language learning, 2014.

한-베 기계번역에서 한국어 분석기 (UTagger)의 영향

원광복, 옥철영

울산대학교
nqphuoc@gmail.com, okcy@ulsan.ac.kr

Effect of Korean Analysis Tool (UTagger) on Korean-Vietnamese Machine Translations

Quang-Phuoc Nguyen, Cheol-Young Ock
University of Ulsan

Abstract

With the advent of robust deep learning method, Neural machine translation has recently become a dominant paradigm and achieved adequate results in translation between popular languages such as English, German, and Spanish. However, its results in under-resourced languages Korean and Vietnamese are still limited. This paper reports an attempt at constructing a bidirectional Korean-Vietnamese Neural machine translation system with the supporting of Korean analysis tool – UTagger, which includes morphological analyzing, POS tagging, and WSD. Experiment results demonstrate that UTagger can significantly improve translation quality of Korean-Vietnamese NMT system in both translation direction. Particularly, it improves approximately 15 BLEU scores for the translation from Korean to Vietnamese direction and 3.12 BLEU scores for the reverse direction.

Keywords: Korean-Vietnamese Machine Translation, Neural Machine Translation, WSD, Homological analysis.

1. Introduction

Using the computer to translate text from a language into another is referred to machine translation (MT) that has been a desire from the early 1950s. Various approaches have been investigated to build a quality MT system, such as dictionary-based, rule-based, statistical, and neural network. Currently, with the advent of robust deep learning method, the neural machine translation (NMT) has attracted much attention to become a dominant paradigm in MT area with remarkable improvements in comparison with rule-based and statistical approaches [1–3]. Recently, NMT systems have achieved adequate results when translating between several popular languages such as English, German, French, and Spanish.

The MT systems for under-resourced languages Korean and Vietnamese also need to be investigated to serve the development of bilateral cooperation between South Korea and Vietnam. Since 2014, South Korea has been Vietnam's biggest investor by foreign direct investment, whereas Vietnam ranks third among hosting FDI from South Korea following by the United States and China [4]. Furthermore, according to the statistic of South Korea Immigration Service in July 2017[1], Vietnamese is the top second foreigner community in Korea with over 160 thousand people. Having a system that can translate between Korean and Vietnamese languages is necessary to help Korean as well as Vietnamese people easily understand each other.

In this paper, we describe our research to build a high-quality Korean-Vietnamese (Kr-Vn) MT system using the dominant NMT approach. However, Korean is a morphologically complex language that does not have clear optimal word boundaries for MT. This causes a major problem of translating into or from Korean. In this paper, we apply our Korean analysis toolkit – UTagger[2] to NMT. UTagger can simultaneously analyze Korean morphology, determine the correct sense of multiple meanings words (WSD), and tag POS for each word in sentences. The method to apply UTagger to NMT system is described in section 4.2.

Besides, parallel corpus plays an important role in NMT as it is the training data set for the translation mode. To build Kr-Vn NMT system, we firstly build a Kr-Vn parallel corpus by collecting Kr-Vn sentence pairs manually from diversified resources. Our current parallel corpus has the size of over 280 thousand sentence pairs. The detail of this corpus is stated in section 4.1.

Based on the collected parallel corpus and the tool UTagger we build a bidirectional Kr-Vn NMT system. Experiment results demonstrated Korean analysis – UTagger could significantly improve translation quality of Kr-Vn NMT system in both translation direction. Particularly, it improved approximately 15 BLEU scores for the translation from Korean to Vietnamese direction and 3.12 BLEU scores for the reverse direction.

[1] http://www.immigration.go.kr → 통계자료실 → 통계월보

[2] http://nlplab.ulsan.ac.kr/doku.php?id=utagger

2. Related Work

Lee et al. [5] addressed the problems of sparse corpora, ambiguities of homophones, and multiple word expression in Kr-Vn statistical machine translation (SMT). To solve these problems, in preprocessing step, they tagged the training corpus with name entity. Then they used MOSES toolkits to train their translation model. The experiment results showed that the method could improve the translation quality approximately 0.8 BLEU scores for Korean to Vietnamese direction and nearly 0.6 scores for vice versa direction.

Nguyen et al. [6] proposed a method to analyze Korean morphology for Korean side in training corpus. Korean is a morphologically complex language that does not have clear optimal word boundaries causes a major problem of translating into or from Korean. After applying morphological analyzing to Kr-Vn SMT system, the translation improved about 3.3 BLEU scores.

Korean words *eojeol* (어절) usually contain one or more function words such as postposition (조사) or ending (어미). The form of these function words is changed depending on their final consonant (받침). Lee et al. [7] standardized the form of ending and postposition in training corpus before training the SMT model. The experiment results showed that this method could improve the translation quality approximately 1 BLEU score for Vietnamese to Korean direction. However, for vice versa direction the results were reduced.

Further research, Cho et al. [8] proposed a method to extract words and phrases inside brackets, parentheses, or quotes so that these words and phrases can be translated individually. The experiments were carried out on Kr-Vn SMT showing it is effective.

Most of the proposed Kr-Vn MT systems belong to SMT approach that has been proved underperform NMT [1-3]. In this paper, we develop a Kr-Vn MT system based on NMT approach with the reinforcements of Korean morphological analysis closely related to the studies of Nguyen et al. [6]. Moreover, we apply even WSD to our NMT system.

3. Neural Machine Translation

As a data-driven based method, NMT require a parallel corpus to train the translation model which is used to find a target sentence *y* by maximizing the condition probability of *y* given a source sentence *x*. Neural translation model is a sequence-to-sequence framework consisting of an *encoder* and a *decoder* recurrent neural network (RNN) [9-10].

The *encoder* RNN reads a variable-length source sentence as a sequence of vectors $x = (x_1, \dots, x_{T_x})$ and then encode it into a fixed-length vector *c by*

$$c = q(\{h_1, \dots, h_{T_x}\}) \quad (1)$$

$$h_t = f(x_t, h_{t-1})$$

where h_t is a hidden state of the RNN at time *t; q* is a nonlinear activation function; *f* can be a logistic sigmoid function or a long short-term memory unit.

The *decoder* RNN decodes the vector *c* into a variable-length target sentence $y = (y_1, \dots, y_{T_y})$ by the joint probability

$$p(y) = \prod_{t=1}^{T} p(y_t | \{y_1, \dots, y_{t-1}\}, c)$$
$$p(y_t | \{y_1, \dots, y_{t-1}\}, c) = g(y_{t-1}, s_t, c) \quad (2)$$

where q is a nonlinear activation function and s_t denotes the decoding hidden state of the RNN at time *t.* In this case, the output word y_t is predicted at time *t* depended on all the preciously predicted words.

4. Data Preparation

In this section, we describe the training and test data set. After these data sets were built and standardized, then they were individually analyzed by the characteristics of each language.

- Korean: morphology analyzing, POS tagging, WSD tagging
- Vietnamese: word segmentation, POS tagging

4.1 Parallel Corpus Building

Besides the learning model, the parallel corpus is one of an essential component of NMT system as it is used as a training data set for the translation model. A good NMT requires a parallel corpus with a vast number of sentences pairs. Building such kind of large corpus takes many time, efforts and high cost. The public parallel corpora are available for only popular language pairs. However, for Kr-Vn, there is no such kind of parallel corpus available for researchers.

In this work, we have to build Kr-Vn parallel corpus by collecting manually from diversified resources. We extracted phrase and sentence example of Kr-Vn pairs from Naver dictionary[3], the most popular and accurate dictionary for Kr-Vn. We extracted definition statement of Kr-Vn pairs from National Institute of Korean Language's Learner Dictionary[4]. We downloaded and aligned Kr-Vn sentence pairs from articles on the multilingual magazines "Watchtowers and Awake![5]", and "Rainbow[6]" that include many categories (economy, entertainment, health, science, social, political, and technology). After normalizing (remove long sentences, remove duplicates, re-correct the splitting of sentences), we obtained over 280 thousand sentence pairs as showed in Table 1 where Token denotes the

[3] http://vndic.naver.com
[4] https://krdict.korean.go.kr
[5] https://www.jw.org/en/publications/magazines/
[6] https://www.liveinkorea.kr

eojeol (어절) for Korean and the syllable for Vietnamese.

Table 1. Korean-Vietnamese parallel corpus

	Training Set		Test Set	
	Korean	Viet.	Korean	Viet.
#Sentence	280,134		1,000	
#Token	1,153,991	2,238,848	8,597	15,856
#Word	2,174,331	1,717,941	16,224	11,852
#Vocabulary	45,563	29,642	2,214	2,195

4.2 Korean Analysis – UTagger

Once training parallel corpus have been collected, this section illustrates the processes of analyzing Korean sentences including morphological analyzing, POS tagging, and WSD. These processes are parts of our open tool UTagger, which is available in both online using and downloadable application through internet.

4.2.1. Morphology Analysis and POS Tagging

Unlike English, which has each word clearly segmented, or Vietnamese is devoid of morphology language, Korean is a morphologically complex language that does not have clear optimal word boundaries causes a major problem of translating into or from Korean. In Korean, the spacing unit delimited by a whitespace is called an *eojeol* (어절) which consists not only the content word but also one or more function word(s) such as *josa* (조사 - postposition) or *eomi*(어미 - ending). Korean morphological analysis is to decompose an *eojeol* into morphemes with three processes as the following.

- Separating a *eojeol* into morphemes
- Recovering original form for changed phonemes
- Tagging the POS to each morpheme

UTagger is one of the most accurate morphological analysis tools for Korean. UTagger analyzes morphology using a pre-analyzed partial *eojeol* dictionary [11], which is firstly constructed from the Sejong tagged corpus with over eleven millions of eojeol and a set of rules for the irregular phoneme changes and compound nouns. After analyzing the morphology, UTagger conducts the POS tagging for each morpheme based on Hidden-Markov model with 48 kinds of POS in tag set.

The experiments are carried out on the Sejong corpus demonstrate that the accuracy reaches 96.76%, and the recall rate is 99.05%. The running time consumes 23 seconds in case of analyzing 11 million *eojeols* on the system with CPU i7 860(2.8GHz).

4.2.2. Word Sense Disambiguation (WSD)

It is commonly assumed that WSD should help to improve the quality of MT systems. Ambiguity causes the word choice problem in which a word in source language may have many different translations in the target language, and MT systems must choose a correction between them. If the sense of a source word is disambiguated in advance, the corresponding target word will be correctly chosen. With resolving sense ambiguity, WSD will be able to help MT systems to determine the correct translation of ambiguous words.

Recently, the integration of WSD into MT systems has been successful at the improving of translation quality between popular language pairs by different methods. Su et al. [12] used graph-based framework for collective lexical selection in Chinese-English MT. Neale et al. [13] used the word senses as contextual features in maxent-based translation models for English-Portuguese MT. Vintar and Fišer [14] integrated a WordNet-based unsupervised in English-Slovene MT.

In this research, we present an attempt at applying the Korean WSD to NMT systems by adding a distinct sense-code to each sense of Korean multi-sense words. Each sense-code consists digits that are defined in the Standard Korean Language Dictionary (SKLD) (표준국어대사전) as the representative of each sense of Korean multi-sense words. For instance, the sense-codes of Korean word “배” are defined in SKLD from 01 to 13 to represent 13 different senses as shown in Table 2. The adding one of such 13 sense-codes to the word “배” metamorphoses this word “배” into a different word. In this way, instead of inputting a single “배” into NMT system, a different word includes “배” and its added sense-codes is inputted into NMT system, this means that the ambiguity of the word “배” has been solved.

Table 2. Definition of Sense-codes of “배”

Code	POS	Meaning
01	noun	stomach, belly, abdomen, tummy
02	noun	boat, ship, vessel
03	noun	pear
04	noun	heavy rope, hawser
05	bound noun	trophy, cup
06	noun	worship, respect
07	noun	the root of the word ‘배하다’
08	noun	embryo
09	noun	double, two times, times
10	noun	a surname in Korea
11	suffix	a suffix means people of a group
12	noun	a combination of words ‘바’and ‘이’
13	adverb	very, much, so, extremely

Let us consider the Korean sentence “배를 먹고 배를 탔더니 배가 아팠다” with meaning “I had a stomachache after eating a pear and boarding the ship.” In this sentence, the word ‘배’ appears three times with three different meanings “pear”, “ship”, and “stomach”. Looking up the sense-codes for such different meanings in Table 2, we get the codes 03, 02, and 01 corresponding with meanings “pear,” “ship,” and “stomach” respectively. After adding the corresponding sense-codes to the word “배”, the mentioned sentence is metamorphosed into form

"배_03 를 먹고 배_02 를 탔더니 배_01 가 아팠다" where the word '배' and its added sense-codes are combined to a different word '배_01', '배_02' or '배_03' depending on its meaning. Since computer uses the blank spaces to separate words, there is no ambiguity of '배' in this sentence form.

To deal with Korean WSD, we have manually constructed a Korean lexical semantic network (LSN) – UWordMap [15] since 2002. The base knowledge used for constructing UWordMap is obtained from SKLD contains words in all POS and their sense-codes. UWordMap consists of a hierarchical structure network for nouns, a subcategorization of verbs and adjectives, and predicate connections between them. Currently, it has a vocabulary of about 366 thousand nouns, over 73 thousand verbs, nearly 17 thousand adjectives, and over 17 thousand adverb. It is not only useful for MT, but also for various fields such as information retrieval and semantic web by using its application-programming interface or online service[7].

Once morphology was analyzed into morphemes, UTagger use this UWordMap to identify the correct sense of each morphemes and tag the corresponding sense-codes for them [16]. The experiment on the Sejong corpus demonstrate that UTagger can identify the correct sense with accuracy 96.52%.

4.3 Vietnamese Analysis

4.3.1. Word Segmentation

Unlike English, Vietnamese is a monosyllable language that is one word is composed of one or more syllables. In Vietnamese, blank spaces are not only used to separate words, but they are also used to separate syllables. Furthermore, many of Vietnamese syllables are words by themselves, but can also be part of multi-syllable words. Hence, we cannot use the blank space to determine the word boundaries. In this research, to segment Vietnamese words in parallel corpus, we used the open-source tool vnTokenizer [17].

The tool uses the finite-state automata technique to build linear graphs corresponding to the phrases that are separated from the input sentence. Then it generates all segmentation candidates from the graphs by using the maximal-matching strategy. Finally, it chooses the most probable segmentation based on the bigram language model. To train and evaluate this tool, they used a corpus of the Vietnam Lexicography Center that contains manually spell-checked and segmented 507,358 words. The experiment results show this tool's accuracy is over 96%.

4.3.2. POS Tagging

To apply factor NMT architectures [18], we use the open source tool JVnTextPro [19] to conduct the POS tagging for Vietnamese side in the training parallel corpus. JVnTextPro that is based on Conditional Random Fields and Maximum Entropy was trained on a dataset consisting of 20.000 sentences with 18 kinds of POS from Vietnamese TreeBank[8]. The experiment results show this tool has 93.45% accuracy.

5. Experimental Result

5.1. System Architecture

We implement the Kr-Vn NMT system relying on the open source toolkit OpenNMT [20], which has been developed based on the jointly learning to align and translate method to NMT of Bahdanau et al. [21]. In convention NMT, the encoder RNN reads an input sequence $x = (x_1, \dots, x_{T_x})$ from left to right described in equation (1). Instead, this method uses a bidirectional RNN that consists of forward and backward RNNs. The forward RNN reads the input sequence from left to right and calculates a sequence of the forward hidden states $(\vec{h}_1, \dots, \vec{h}_{T_x})$. The backward RNN reads the sequence in the reverse order, producing a sequence of the backward hidden states $(\overleftarrow{h}_1, \dots, \overleftarrow{h}_{T_x})$. Then, the source annotations $\{h_j\}$ of each word x_j is computed by concatenating the hidden states of these two RNNs, where $h_j = [\vec{h}_j^T; \overleftarrow{h}_j^T]$ encodes information about the j-th word concerning all the other surrounding words.

In the decoder RNN, unlike the conventional equation (2), here the probability is conditioned on a distinct context vector c_i for each target word y_i.

$$p(y_i|\{y_1, \dots, y_{i-1}\}, x) = g(y_{i-1}, s_i, c_i) \tag{3}$$

The context vector c_i depends on a sequence of source annotations $(h_1, \dots, h_{T_x})$ computed as a weighted sum of these annotations h_i:

$$c_i = \sum_{j=1}^{T_x} \frac{\exp(e_{ij})}{\sum_{k=1}^{T_x} \exp(e_{ik})} h_j$$
$$e_{ij} = a(s_{i-1}, h_j) = v_a^T \tanh(W_a s_{i-1} + U_a h_j) \tag{4}$$

Where e_{ij} is an alignment model, which measure how well the inputs around position *j* and the output at position *i* match. $W_a \in \mathbb{R}^{n \times n}$, $U_a \in \mathbb{R}^{n \times 2n}$, and $v_a \in \mathbb{R}^n$ are the weight matrices.

5.2. Implementation

To train the Kr-Vn NMT system, we used the training parallel corpus of approximately 280 thousand sentence pairs as described in table 1. Before training the system, we had done the pre-processing on this parallel corpus with following steps.

- Conducting the word segmentation process for Vietnamese sentences
- Analyzing morphology for Korean sentences

[7] http://nlplab.ulsan.ac.kr/doku.php?id=uwordmap

[8] https://vlsp.hpda.vn/demo/?page=resources

- Tagging sense-code for Korean sentences
- Tagging POS for both Korean and Vietnamese sentences.

Then, the system was setup with parameters: word-embedding dimension as 500, hidden layer as 2x500 RNNs, input feed as 13 epochs. We carried out the translation system on both bidirectional Korean to Vietnamese and Vietnamese to Korean.

Because UTagger includes morphological analyzing, POS tagging, and WSD, the system applied UTagger refers to Morp-WSD-POS. To evaluate the effect of UTagger on this system, we compared with a baseline system that uses the same training parallel corpus and the same setting parameters but did not apply any Korean analysis process. The baseline system jus was applied the Vietnamese word segmentation. Furthermore, to evaluate the effect of WSD severally, we also carried out another system, Morp-WSD, that was applied the morphological analyzing and sense-code tagging for Korean side. The input forms for these systems are illustrated in Table 3.

We trained these systems on CPU core i7 (680) and 16GB RAM without GPU. The time consumption for each epoch was 308 minutes. Each system was run with 13 epochs, so we needed 4,004 minutes (~66.7 hours) to train each system for one-way direction.

We tested these systems with a testing set of 1,000 sentence pairs as described in table 1.

Table 3. Example of Training Input Forms

Baseline	Kr	배를 먹고 배를 탔더니 배가 아팠다 .
	Vn	Sau_khi ăn lê rồi lên tàu tôi bị đau_bụng .
Morp-WSD-POS	Kr	배_03|NNP 를|JKO 먹_02|VV 고|EC 배_02|NNP 를|JKO 타_02|VV 았|EP 더니|EC 배_01|NNP 가|JKS 아프|VA 았|EP 다|EF .|SF
	Vn	Sau_khi|N ăn|V lê|V rồi|C lên|V tàu|N tôi|P bị|V đau_bụng|A .|.
Morp-WSD	Kr	배_03 를 먹_02 고 배_02 를 타_02 았 더니 배_01 가 아프 았 다 .
	Vn	Sau_khi ăn lê rồi lên tàu tôi bị đau_bụng .

5.3. Evaluation

We used evaluation metrics BLEU, TER, and DLRATIO measure the translation quality. BLEU (Bi-Lingual Evaluation Understudy) [22] measures the precision of an MT system by comparing the n-grams of a candidate translation with those of the corresponding reference and counts the number of matches. In this research, we use BLEU metric with 4-gram. TER (Translation Error Rate) [23] is an error metric for MT that measures the number of edits required to change a system output into one of the references. DLRATIO [24] (Damerau-Levenshtein edit distance) measures the edit distance between two sequences.

Table 4 shows the results in translation from Korean into Vietnamese direction, whereas Table 5 shows the results in translation from Vietnamese into Korean direction of three systems mentioned in section 5.2.

Table 4. Korean to Vietnamese Translation Results

	BLEU	TER	DLRATIO
Baseline	18.45	60.13	47.54
Morp-WSD	27.90	56.65	45.03
Morp-WSD-POS	34.44	48.67	42.42

Table 5. Vietnamese to Korean Translation Results

	BLEU	TER	DLRATIO
Baseline	19.90	59.43	52.29
Morp-WSD	22.27	55.61	47.54
Morp-WSD-POS	23.02	54.01	47.56

The metrics in Table 4 showed that the UTagger to NMT systems could remarkably improve translation quality with 15.99 BLEU scores for the translation from Korean to Vietnamese direction. It also reduced the translation error with 11.46% TER and 5.3% DLRATIO in the same translation direction. However, in the reverse direction, the translation quality was just improved 3.12 BLEU scores, and translation error was reduced 5.42% according to TER and 4.73% according to DLRATIO as shown in Table 5.

The disproportionate improvement of translation performance in different translation direction can be easily explained that we just applied the morphological analysis and sense-code tagging for Korean side only. Hence, in the Korean to Vietnamese translation direction, the improvement is more significant than the reverse direction.

The next, we evaluate the effect of sense-code tagging (WSD) on NMT system. According to the BLUE scores in Table 4 and Table 5, WSD could improve the translation quality in both translation directions. In the Korean to Vietnamese translation direction, it is simple to understand that the WSD help the NMT system correctly select target words, so it improved 9.45 BLEU scores. In reverse direction Vietnamese to Korean, WSD improved 2.37 BLEU scores. This improvement can be explained that before tagging sense-code, Korean sentences had analyzed morphology. The Korean morphological analysis reduces the unknown word (out-of-vocabulary words) problem in the alignment model.

Overall, Korean analysis – Utagger could significantly improve translation quality of Korean-Vietnamese NMT system in both directions. With the promising results, we can say that the Korean analysis – Utagger makes the significant improvement of MT into or from Korean. It means Korean analysis – Utagger maybe effect to not only Korean-Vietnamese but also Korean and another language in NMT.

6. Conclusion

This paper has presented our work on building a bidirectional

Kr-Vn NMT system. In this work, we have collected a highly valuable Kr-Vn parallel corpus of over 281 thousand sentence pairs to train the neural translation model. For Korean side, we applied the analysis – UTagger, includes morphological analyzing, POS tagging, and WSD. For Vietnamese side, we conducted word segmentation process and POS tagging. Experiment results demonstrated Korean analysis – UTagger could significantly improve translation quality of Kr-Vn NMT system in both translation direction.

In the future, we will process the WSD for Vietnamese to improve the quality of translation from Vietnamese to Korean. Additionally, we plan to study the applying of syntactic and parsing attentional model to Kr-Vn NMT systems.

Acknowledgement

This work is supported by ICT R&D program of MSIP/IITP. [2013-0-00179, Development of Core Technology for Context-aware Deep-Symbolic Hybrid Learning and Construction of Language Resources].

References

[1] L. Bentivogli, A. Bisazza, M. Cettolo, and M. Federico, “Neural versus phrase-based machine translation quality: a case study”, *arXiv preprint arXiv: 1608.04631*, Aug. 2016.

[2] J. Crego et al., “SYSTRAN’s Pure Neural Machine Translation Systems”, *arXiv preprint arXiv: 1610.05540*, Oct. 2016.

[3] M. Junczys-Dowmunt, T. Dwojak, and H. Hoang, “Is neural machine translation ready for deployment? A case study on 30 translation directions”, *arXiv preprint arXiv: 1610.01108*, Oct. 2016.

[4] J. H. Oh, and J. S. Mah. “The Patterns of Korea’s Foreign Direct Investment in Vietnam”, *Open Journal of Business and Management*, Vol. 5, pp. 253-271, 2017.

[5] 이원기 et al., "개체명 인식과 단어 정렬을 이용한 통계적 기계번역의 성능 향상", *한국정보과학회 학술발표논문집*, pp. 615-617, 2017.

[6] Q. P. Nguyen, J. C. Shin, and C. Y. Ock, “Korean Morphological Analysis for Korean-Vietnamese Statistical Machine Translation”, present at *the 9th International Conference on Computer Research and Development (ICCRD)*, Vietnam, 2017.

[7] 이원기 et al., "한국어의 이형태 표준화를 통한 구 기반 통계적 기계 번역", 제28회 한글 및 한국어 정보처리 학술대회 논문집, 2016.

[8] 조승우 et al., "한베 통계기계번역의 성능 향상을 위한 내포문 추출 및 복원 기법", 제28회 한글 및 한국어 정보처리 학술대회 논문집, 2016.

[9] I. Sutskever, O. Vinyals, and Q. Le, “Sequence to sequence learning with neural networks”, *Advances in Neural Information Processing Systems*, pp. 3104-3112, 2014.

[10] K. Cho et al., “Learning phrase representations using RNN encoder-decoder for statistical machine translation”, *arXiv preprint arXiv: 1406.1078*, 2014.

[11] J. C. Shin and C. Y. Ock, “A Korean morphological analyzer using a pre-analyzed partial word-phrase dictionary”, *KIISE: Software and Applications*, vol. 39, pp. 415-424, 2012.

[12] J. Su et al., “Graph-Based Collective Lexical Selection for Statistical Machine Translation”, in *Proc. of EMNLP*, Portugal, 2015, pp. 1238–1247.

[13] S. Neale et al., “Word sense-aware machine translation: Including senses as contextual features for improved translation models”, in *Proc. of LREC*, Slovenia, 2016, pp. 2777–2783.

[14] Š. Vintar and D. Fišer, “Using wordnet-based word sense disambiguation to improve MT performance”, *Hybrid Approaches to Machine Translation*, Springer International, 2016, pp. 191–205.

[15] Y. J. Bae, C. Y. Ock, "Introduction to the Korean Word Map (UWordMap) and API," in *Proc. of 26th Annual Conf. on Human and Language Technology*, 2014, pp. 27-31.

[16] J. C. Shin and C. Y. Ock, “Improvement of Korean Homograph Disambiguation using Korean Lexical Semantic Network (UWordMap)”, *KIISE: Software and Applications*, vol. 43, pp. 71-79, 2016.

[17] L. H. Phuong et al., “A hybrid approach to word segmentation of Vietnamese texts”, in *Proc. of the 2nd Int. Conf. on Language and Automata Theory and Applications (LATA)*, Spain, 2008, pp. 240-249.

[18] G. M. Mercedes, L. Barrault, and F. Bougares, “Factored neural machine translation architectures”, In *Proc. of the International Workshop on Spoken Language Translation - IWSLT’16*, Seattle, USA, 2016.

[19] X. H. Phan, “JVnTextPro: A Java-based Vietnamese text processing tool,” http://jvntextpro.sourceforge.net, 2010.

[20] G. Klein et al., “OpenNMT: Open-Source Toolkit for Neural Machine Translation”, *arXiv preprint arXiv: 1701.02810*, Jan. 2017.

[21] D. Bahdanau, K. H. Cho, and Y. Bengio, “Neural machine translation by jointly learning to align and translate”, in *Proc. of ICLR*, 2015.

[22] K. Papineni, S. Roukos, T. Ward, and W. Zhu, “Bleu: A method for automatic evaluation of machine translation,” in *Proc. of ACL2002*, 2002, pp. 311–318.

[23] M. Snover et al., “A study of translation edit rate with targeted human annotation”, *In Proc. of Association for Machine Translation in the Americas*, Massachusetts, USA, 2006.

[24] G. V. Bard, “Spelling-error tolerant, order-independent pass-phrases via the Damerau-Levenshtein string-edit distance metric”, *In Proc. of the fifth Australasian symposium on ACSW frontiers-Volume 68*, Australian Computer Society, Inc.. , pp. 117-124, 2007.

단어 임베딩과 음성적 유사도를 이용한 트위터 '서치 방지 단어'의 자동 예측

이상아○
서울대학교 언어학과
visualjan@snu.ac.kr

Automatic Prediction of 'Anti-Search Variants' of Twitter based on Word Embeddings and Phonetic Similarity

Sangah Lee○
Seoul National University, Dept. of Linguistics

요 약

'서치 방지 단어'는 SNS 상에서 사용자들이 작성한 문서의 검색 및 수집을 피하기 위하여 사용하는 변이형을 뜻한다. 하나의 검색 키워드가 있다면 그와 같은 대상을 나타내는 변이형이 여러 형태로 존재할 수 있으며, 이들 변이형에 대한 검색 결과를 함께 수집할 수 있다면 데이터 확보가 중요하게 작용하는 다양한 연구에 큰 도움이 될 것이다. 본 연구에서는 특정 단어가 주어진 키워드로부터 의미 벡터 상의 거리가 가까울수록, 그리고 주어진 키워드와 비슷한 음성적 형태 즉 발음을 가질수록, 해당 키워드의 변이형일 가능성이 높을 것이라고 가정하였다. 이에 따라 단어 임베딩을 이용한 의미 유사도와 최소 편집 거리를 응용한 음성적 유사도를 이용하여 주어진 검색 키워드와 유사한 변이형들을 제안하고자 하였다. 그 결과 구성된 변이형 후보의 목록에는 다양한 형태의 단어들이 포함되었으며, 이들 중 다수가 실제 SNS 상에서 같은 의미로 사용되고 있음이 확인되었다.

주제어: 서치 방지 단어, 단어 임베딩, 음성적 유사도, 최소 편집 거리

1. 서론

최근 트위터나 블로그와 같은 SNS의 데이터를 이용하여 특정 주제에 대한 대중의 의견을 수집하고 참고하려는 움직임이 커지고 있다. 한편 개인이 SNS에서 자유롭게 피력한 의견들이 수집을 위한 검색에 노출될 것을 우려하여, 이에 반하는 전략들도 나타나기 시작하였다. 그 중 하나는 '서치 방지 단어'의 사용인데, '서치 방지 단어'란 SNS 상에서 검색을 피하기 위해 사용되는, 특정 단어의 변이형들을 말한다.

[표1] 서치 방지 트윗 예시

헬씽ㅋi 스벆은 휘핑크림을 재밌게 주는군요!

예를 들면 [표1]의 트윗에서 '헬씽ㅋi'와 '스벆'은 각각 '헬싱키'와 '스벅(스타벅스)' 대신에 사용된 서치 방지 단어라고 할 수 있다. SNS의 특성상 작성자의 신분이나 위치 등이 드러날 수 있으므로 외부 검색에 의한 접근은 원치 않으나, 해당 단어들을 언급하거나 그에 관한 글을 작성하고자 할 때 서치 방지 단어를 사용하는 것이 일반적이다. 보통은 한 가지 형태로 약속되기보다는 다양하게, 산발적으로 나타난다.

이러한 검색 방지의 유형은 크게 단어의 시각적 형태에 따른 것, 청각적 형태에 따른 것의 두 가지로 나누어 볼 수 있다. 먼저 시각적 형태에 주목한 경우에는 한글 자모를 조합한 모양이 비슷한 것들끼리 대치하는 소위 '야민정음'[1]과, 영문자나 숫자, 특수기호 등을 섞어 한글처럼 보이게 하는 경우[2] 등이 포함된다. 청각적 형태에 따른 변이형은 소리내어 읽었을 때 유사한 발음으로 실현되는 것[3]이며, 본 연구는 이 경우에 중점을 두어 진행되었다. 한편 형태 측면과는 다소 거리가 있으나, 집단 내에서 암묵적으로 약속된 대체어를 이용하는 경우[4]나 해당 단어의 초성만을 기재하는 경우[5] 역시 존재한다.

본 연구에서는 하나의 키워드를 기준으로 발생 가능한 변이형들을 자동으로 예측하여 얻고자 하였다. 이를 위하여 단어 임베딩을 이용한 단어 간의 의미적 유사도와 최소 편집 거리(Minimum Edit Distance)[1]를 응용하여 얻은 단어 간의 음성적 유사도를 함께 고려하였다. 최종적으로는 의미적 유사도와 음성적 유사도를 가중합하여 주어진 단어와 변이형 후보 사이의 유사도 점수를 계산하고, 이 점수가 높은 순서대로 변이형을 제안한다.

이러한 흐름에 따라 한 가지 키워드를 가지고 SNS 상에 나타난 대중의 의견을 검색할 때, 형태는 조금 다르

1) (예) 파이널판타지(파이널판타지)
2) (예) ㅇ1디야(이디야커피), ㅂ6탄쇼넨단(방탄소년단)
3) (예) 스따벆스(스타벅스), 탬인(태민), 정핝(정한)
4) (예) 넬(강다니엘)
5) (예) ㅂㅌ(방탄, 방탄소년단)

지만 같은 대상을 나타내는 변이형들을 언급한 문서들도 함께 검색되도록 한다. 이를 통해 보다 풍부한 검색 결과를 얻고, 이러한 데이터의 확장은 다양한 자연어처리 연구에 도움이 될 것이다.

2. 관련 연구

검색 키워드의 가능한 변이형을 제안하는 연구는 아직 충분히 이루어지지 않은 실정이다. 주어진 단어(문자열) 사이의 유사도를 측정하는 방법은 다양하게 제시되었으나, 이를 응용하여 발생 가능한 문자열의 변이를 예측하는 연구는 드문 편이다.

생물정보학 분야의 서열 정렬(sequence alignment) 방식과 편집 거리 방식을 함께 이용하여 오자, 탈자 등의 입력 오류를 허용하는 근사 한글 검색 시스템이 앞서 제안된 바 있다. 해당 연구에서는 같은 조음 위치에서 발음되는 평음, 경음, 격음을 한 가지 평음으로, 이중모음을 단모음으로 표준화하여, 다양한 형태로 변형된 욕설을 필터링하는 시스템을 구축하였다[2]. 또한 다수의 문자열에 대하여 서열 정렬을 수행하는 다중 서열 정렬(multiple sequence alignment) 방식을 채택한 연구도 존재한다[3]. [3]의 연구에서는 하나의 문자열로부터 파생된 변이형들이 포함된 문자열 집합이 주어졌을 때 그 중 대표 문자열을 정의하고, 문자열과 문자열 집합 간의 유사도 계산 방법을 제안하여 문자열 집합 내에 특정 문자열이 포함되어 있는지 아닌지를 출력하는 문제를 다루었다.

한편 스마트폰 가상 키패드 상에서 발생할 수 있는 입력 오류에 의한 유사 단어를 검색하는 문제도 다루어졌다[4]. 이 연구에서는 스마트폰 키패드의 종류에 따라 편집 거리 알고리즘에 사용되는 편집 비용을 수정하는 방식을 택하였다.

본 연구는 단어로부터 발생 가능한 변이형을 예측한다는 점에서는 이전 연구들과 공통적이나, 단어 사이의 편집 거리를 이용해 정의한 음성적 거리와 단어 임베딩에 기반한 의미의 유사성을 함께 고려하고자 하였다.

3. 서치 방지 단어의 자동 예측

3.1 어휘 사이의 의미 유사도

먼저 문맥에서 얻어지는 단어와 단어 사이 의미의 유사성을 이용하여 변이형을 예측한다. 형태가 다를지라도 같은 것을 뜻하는 단어들의 경우 공기하는 단어들이나 문맥과 관련하여 제공하는 정보가 서로 유사할 가능성이 높기 때문이다.

이러한 의미의 유사성을 수치화하고 연산하기 위하여 트위터에서 수집한 훈련 데이터에서 Word2Vec의 C-BOW 모델을 이용하여 단어 임베딩을 구축하였다[5]. 훈련 데이터는 각각 140자 이내로 작성된 10486개의 트윗으로 이루어져 있으며, 단어마다 갖는 벡터는 100차원의 자질로 구성되어 있다.

검색에 주로 사용되는 키워드는 일반명사나 고유명사인 경우가 많으므로, 시험 데이터에서 형태소 분석을 통해 품사가 명사인 요소들만을 필터링하여 유사도 계산의 대상으로 한다. 이 때 형태소 분석에는 KoNLPy 패키지 내 Twitter 모듈을 이용하였다[6]. 이렇게 걸러낸 명사들은 넓은 의미에서 변이형의 후보가 된다. 이 때 각각의 후보 단어들과 주어진 기본 키워드 사이의 의미 유사도는 100차원의 단어 임베딩끼리의 코사인 유사도를 이용한다. 시험 데이터에서, 훈련 데이터에는 존재하지 않았던 새로운 어휘가 나타난 경우, 똑같이 100차원의 자질을 가지되 균등 분포를 따르는 임의의 벡터를 생성하여 유사도를 계산할 수 있도록 하였다. 이렇게 계산된 의미 유사도는 0에서 1사이의 값을 갖게 된다.

3.2 어휘 사이의 음성적 유사도

다음으로는 단어와 단어가 음성적으로 유사한 정도를 척도의 하나로 이용하고자 하였다. 음성적 유사도는 최소 편집 거리 알고리즘에 한국어 자소의 음성적 특성[7]을 일부 적용하여 구현하였다.

먼저 최소 편집 거리는 두 문자열이 서로 얼마나 비슷한지를 나타내는 척도 중 하나로, 한 단어의 철자가 다른 단어와 같아지도록 수정하는 과정(철자의 삽입, 삭제, 대체) 각각에 비용을 부여하는 방식이다. 본 연구에서는 먼저 각각의 단어를 자소 단위로 분해하고, 최소 편집 거리 알고리즘을 적용하되 자소들의 특성에 따라 자소의 대체 비용에 각각 다른 값을 부여하는 규칙을 정의하였다. 기본 대체 비용을 1로 설정하고, 아래 [표2]의 기준들에 해당하는 자음, 모음의 그룹 내에서 발생하는 대체에는 0과 1 사이의 값을 정의하여 이용하였다.

[표2] 편집 대체 비용의 판단 기준

기준	자소 그룹	대체 비용
자음의 조음 위치	{ㄱ, ㄲ, ㅋ}, {ㄷ, ㄸ, ㅌ}, {ㅂ, ㅃ, ㅍ}, {ㅅ, ㅆ}, {ㅈ, ㅉ, ㅊ}	0.5
모음 발음상의 유사성	{ㅏ, ㅐ}, {ㅗ, ㅜ}, {ㅓ, ㅗ} 등	0.5
	{ㅔ, ㅐ}, {ㅖ, ㅒ} 등	0.2

자음의 조음 위치에 따른 유사성은 같은 조음 위치에서 발음되는 자음들 중 평음, 격음, 경음의 대립이 존재하는 경우를 상정하여 정의하였다. ‘ㄱ, ㄲ, ㅋ’, ‘ㄷ, ㅌ, ㄸ’ 등의 자음들은 임의로 대체하더라도 발음상의 차이가 덜하여 원래의 형태를 알기 쉬운 편이다.

모음 발음 시 혀의 위치, 입의 개폐 정도의 유사성은 주로 모음사각도 상의 거리에 기반하여 정의된다(‘ㅏ, ㅐ’, ‘ㅔ, ㅐ’ 등). 또한, 이중모음의 경우에도 발음의 유사성과 서치 방지의 일반적인 전략을 고려하여 대체하기 쉬운 모음의 목록을 구성하였다(‘ㅏ, ㅑ’, ‘ㅐ, ㅒ’ 등).

한편 종성에 쓰이는 겹받침 역시 서치 방지 단어의 생

성 전략이 될 수 있다. 발음이 같거나 비슷한 것을 이용하여 홑받침을 겹받침으로, 겹받침을 홑받침으로 대체하는 것이다. 따라서 음성적 유사도를 계산하기 이전에 이들을 각각의 자소로 분리하였다. 예를 들면 'ㅈ'를 'ㄴㅈ'로, 'ㅛ'를 'ㄹㅁ'로 변환하는 것이다.

이러한 방법으로 얻은, 주어진 키워드 w_i와 시험 데이터 내 명사 w_j 사이의 최소 편집 거리를 아래의 식 (1)을 통해 가공하여, 음성적 유사도는 0에서 1 사이의 값을 가지도록 하였다.

$$w_i, w_j\text{의 음성적 유사도} = 1/(w_i, w_j\text{의 최소 편집 거리} + 1) \cdots (1)$$

4. 실험 및 결과

본 연구에서는 각각 140자 이내로 작성된 9547개의 트윗으로 구성된 시험 데이터에 출현한 명사들을 대상으로, 주어진 키워드와의 의미 유사도와 음성적 유사도에 기반한 유사도 점수를 계산하여 실제로 변이형이 될 만한 단어들을 얻고자 하였다. 이 때 유사도 점수는 의미 유사도와 음성적 유사도에 똑같이 0.5의 가중치를 부여한 평균값이 된다.

K-pop 아이돌 그룹명인 '엑소'와 그 멤버 이름인 '세훈'을 기본 키워드로 부여하고, 시험 데이터에 쓰인 명사들 중 이들 키워드와 가장 유사한 20개의 명사를 추출하였다.

[표3] 유사도 기반 변이형 제안 결과

키워드='엑소' (유사도)	키워드='세훈' (유사도)
엑소 (1.0)	**세훈 (1.0)**
액소 (0.748)	**새훈 (0.799)**
엑스 (0.616)	**세혼 (0.745)**
예고 (0.573)	**셰훈 (0.725)**
엔시 (0.571)	**새혼 (0.711)**
앢소 (0.563)	**섀훈 (0.706)**
악수 (0.553)	**세후 (0.705)**
엓소 (0.564)	세운 (0.694)
엔씨 (0.552)	**세홍 (0.685)**
애교 (0.551)	**새후 (0.659)**
육수 (0.550)	**세후니 (0.652)**
백시 (0.549)	**새홍 (0.641)**
윾소 (0.547)	세준 (0.632)
야기 (0.531)	**새휸 (0.628)**
해고 (0.530)	쳰 (0.615)
엥셔 (0.530)	백현 (0.614)
악어 (0.530)	**쎄홍 (0.613)**
섹시 (0.528)	수호 (0.613)
센스 (0.527)	**오세훈 (0.611)**
에스 (0.526)	세호 (0.610)

위 [표3]에서 키워드인 '엑소'와 '세훈'의 실제 변이형인 명사들은 굵은 글씨로 표시하였다. 키워드가 '엑소'인 경우 제안된 상위 10개의 변이형 중에서는 4개의 명사, 상위 20개의 변이형 중에서는 5개의 명사가 실제로 사용되고 있음을 확인하였다. 또한 키워드가 '세훈'인 경우 제안된 상위 10개의 명사 중에서는 9개, 상위 20개 중에서는 14개의 변이형이 정답에 해당하는 것으로 확인되었다. 이러한 결과는 [표4]에 정리되어 있으며, 두 경우 모두 정답률은 상위 10개까지의 변이형을 채택했을 때 더 높게 나타났다.

또한 같은 키워드에 대하여 의미 유사도와 음성적 유사도 각각만을 가지고 단어 사이의 유사도를 계산하고, 이를 두 가지 유사도를 모두 사용한 경우와 비교하였다. 이 때 [표4]의 결과에 따르면 두 가지 유사도를 함께 적용했을 때의 정답률이 가장 높았으므로, 적절한 변이형을 제안하는 데 두 척도가 모두 기여한다는 것을 확인하였다.

[표4] 키워드 변이형 정답률

이용한 유사도	키워드	정답 개수 (%)	
		상위 10개	상위 20개
의미 유사도	엑소	1 (10%)	1 (5%)
	세훈	1 (10%)	1 (5%)
음성적 유사도	엑소	6 (70%)	7 (35%)
	세훈	9 (90%)	13 (65%)
의미 유사도 + 음성적 유사도	엑소	**7 (70%)**	**11 (55%)**
	세훈	**9 (90%)**	**15 (75%)**

음성적 유사도는 고려하지 않고 단어 임베딩에 의한 의미 유사도만을 이용한 경우에는 관련된 단어들을 효과적으로 추출하였으나, 정확히 같은 대상을 가리키는 변이형은 제안하지 못하였다. '엑소'와 '세훈'을 키워드로 설정했을 때, 두 경우 모두 해당 아이돌 그룹에 소속된 다른 멤버의 이름('준면', '찬열', '종인', '백현' 등)이 0.99 이상의 높은 코사인 유사도 값을 보이며 변이형으로 예측되었으나, 동일한 대상을 가리키는 변이형을 보다 정확하게 얻기 위해서는 음성적 유사도를 반영할 필요가 있다. [표3]에 제시된 '액소', '앢소', '엓소' 등의 형태는 의미 유사도만으로는 예측할 수 없기 때문이다.

그러나 단어의 의미를 고려하지 않은 채 청각적 형태의 유사성만을 변이형 예측의 단서로 삼는 것은 효과적인 방법이 아니다. 음성적 유사도만을 이용하여 변이형의 후보를 추출할 경우, '악수', '에고', '엑스' 등의 단어들이 0.5의 유사도를 가지고 높은 순위로 예측된다. 이들 단어는 자소들의 음성적 특징에 따라서는 제시된 키워드와 유사하지만 의미 면에서는 그 차원이 전혀 다르다. 따라서 특정 키워드의 변이형을 예측할 때 그 단어가 문맥 속에서 가지는 의미를 반영하여야 할 것이다.

한편 당초 상정하지 않았던 변이형이 높은 점수를 보이며 예측 결과에 포함되는 경우도 존재하였다. [표3]에 굵은 글씨로 표시되어 있는 '윾소', '세혼, 새혼, 세후, 새후, 세후니, 새휸, 쎄홍, 오세훈'[6]이 그것이다. 이들 변이

6) '세후, 새후'의 경우 '세후나(세훈아)', '새후니(새훈이)' 등의 형태소 분석 오류에 의한 것을 포함한다.

형은 학습 데이터와 시험 데이터를 수집하는 과정에서 미리 설정하지 않았으므로 우연히 텍스트 안에 포함된 것들이다. 이는 곧 미리 정답셋으로 구성한 '엑소, 액소, 엑쏘, 엑오, 앸오, 엑소', '세훈, 새훈, 세훙, 새훙, 셰훈, 섀훈'과 같은 형태 외에도 다양한 변이형이 추론될 수 있고, 정답인 변이형을 포함하리라고 보장할 수 없는 임의의 데이터에서도 동일한 연산이 가능하다는 것을 뜻한다.

이러한 분석을 통해, SNS 상에서 사용자들이 특정 주제에 해당하는 키워드를 언급하되 검색 결과에 노출되는 것을 피하기 위하여 어떤 변이형을 쓰는지 예측할 수 있을 것이다.

5. 결론

본 연구는 Word2Vec 모델에 기반한 단어 임베딩을 구성하여 단어 간의 의미 유사도를 구하고, 최소 편집 거리를 응용하여 단어 사이의 음성적 유사도를 계산하여, 주어진 단어로부터 '서치 방지 단어' 즉 해당 단어의 변이형의 후보들을 이끌어내는 것을 목표로 하였다.

의미와 음성적 형태를 동시에 고려한 점수가 높은 명사일수록 변이형이 될 가능성이 높으며, 이러한 변이형을 포함한 문서 즉 트윗은 주어진 키워드의 검색 결과로 함께 제안할 만하다. 해당 트윗은 검색된 키워드와 같은 대상을 가리키는 단어를, 자소의 결합으로 이루어진 형태는 조금 다를지라도, 포함하고 있을 가능성이 높다. 이에 따라 기존의 검색 방법으로는 쉽게 얻을 수 없었던 문서들을 검색 결과에 노출시키고, 특정 주제에 대한 자연어 데이터를 확장함으로써 다양한 연구에 도움이 될 것이다.

향후에는 음성적 유사도를 위한 자소 간의 대체 비용을 정의할 때, 조음 위치나 혀의 위치 이외에 자음 동화와 같은 추가적인 음운 지식을 적용해볼 수 있을 것으로 생각된다. 또한 트위터와 같은 SNS의 특성상 문서 작성자의 ID나 최근 관심사, 거주 지역과 같은 정보, 사용자 간 네트워크 정보 등도 변이형 제안에 중요한 요소로써 고려해볼 수 있을 것이다.

참고문헌

[1] Jurafsky and Martin, Speech and language processing : An introduction to natural language processing, computational linguistics, and speech recognition, Pearson Prentice Hall, 2009.

[2] 윤태진, 조환규, 정우근, "제한된 한글 입력환경을 위한 음소기반 근사 문자열 검색 시스템", 정보과학회논문지 : 소프트웨어 및 응용, 제37권, 제10호, pp. 788-801, 2010.

[3] 김성환, 조환규, "다중서열정렬을 이용한 변형 문자열 집합의 유사도 계산 기법", 정보과학회논문지 : 소프트웨어 및 응용, 제40권, 제1호, pp. 53-60, 2013.

[4] 송명길, 김학수, "다양한 스마트폰 키패드 환경에서 유사 단어 검색을 위한 수정된 편집 거리 계산 방법", 한국콘텐츠학회논문지, 제11권, 제12호, pp.12-18, 2011.

[5] Mikolov et al., Distributed representations of words and phrases and their compositionality, In Advances in neural information processing systems, pp. 3111-3119, 2013.

[6] 박은정, 조성준, "KoNLPy: 쉽고 간결한 한국어 정보처리 파이썬 패키지", 제 26회 한글 및 한국어 정보처리 학술대회 논문집, 2014.

[7] 이호영, "국어 음성학", 태학사, 1996.

문서 임베딩을 이용한 소셜 미디어 문장의 개체 연결

박영민[1], 정소윤[2], 이정엄[3], 신동수[3], 김선아[3], 서정연[1]

서강대학교 컴퓨터공학과[1], LG전자 소프트웨어센터[2], 현대자동차 융합기술개발팀[3]

pymnlp@gmail.com, soyun.jeong@lge.com, {lee.jeongeom, dshin, seona}@hyundai.com, seojy@sogang.ac.kr

Document Embedding for Entity Linking in Social Media

Youngmin Park[1], Soyun Jeong[2], Jeong-Eom Lee[3], Dongsoo Shin[3], Seona Kim[3], Junyun Seo[1]

Department of Computer Science and Engineering, Sogang University[1],

LG Electronics Software Center[2],

Convergence Technology Development Team, Hyundai Motor Company[3]

요 약

기존의 단어 기반 접근법을 이용한 개체 연결은 단어의 변형, 신조어 등이 빈번하게 나타나는 비정형 문장에 대해서는 좋은 성능을 기대하기 어렵다. 본 논문에서는 문서 임베딩과 선형 변환을 이용하여 단어 기반 접근법의 단점을 해소하는 개체 연결을 제안한다. 문서 임베딩은 하나의 문서 전체를 벡터 공간에 표현하여 문서 간 의미적 유사도를 계산할 수 있다. 본 논문에서는 또한 비교적 정형 문장인 위키백과 문장과 비정형 문장인 소셜 미디어 문장 사이에 선형 변환을 수행하여 두 문형 사이의 표현 격차를 해소하였다. 제안하는 개체 연결 방법은 대표적인 소셜 미디어인 트위터 환경 문장에서 단어 기반 접근법과 비교하여 높은 성능 향상을 보였다.

주제어: 개체 연결, 개체명 인식, 위키백과, 문서 임베딩

1. 서론

자연어 문장의 개체명 인식(Named Entity Recognition) 결과는 인명, 지명, 조직명 등의 태그를 갖지만 각 개체명이 구체적으로 어떤 개체를 의미하는지는 불분명 하다. 예를 들어 '*김기범은 반올림에 출연했다.*' 라는 문장에서 '*김기범*'을 인명으로 인식하였더라도 다양한 개체(야구선수, 배우, 가수 등…) 중 어느 개체에 해당하는지 모호하다. (그림 1)과 같이 이러한 모호성을 해결하는 작업을 개체 연결(Entity Linking)이라 한다.

기존의 개체 연구는 주로 입력 문장과 지식 베이스 문서에 동시에 출현하는 단어를 기반으로 한 단어 기반 접근법(Word-based Approach)을 사용하기 때문에 비정형 문장에서는 높은 성능을 기대하기 어렵다. 본 논문에서는 문서 임베딩(Document Embedding)과 선형 변환(Linear Transformation)을 이용하여 이러한 단점을 극복하고자 한다.

2. 관련 연구

기존의 개체명 연결은 개체와 지식베이스에 출현하는 단어, 하이퍼링크 등을 이용하여 두 대상의 문맥 유사도를 계산하는 방법들이 연구되었다[1-2]. 소셜 미디어의 문장은 길이가 짧아 문맥정보가 충분하지 않은 문제가 있다. 이러한 문제를 극복하기 위해 Shen의 연구[3]에서는 사용자의 성향을 모델링하였고, 정소윤의 연구[4]에서는 사용자 과거 문장과 최근 발생한 뉴스 주제를 모델링 하여 성능 향상을 이룬바 있다.

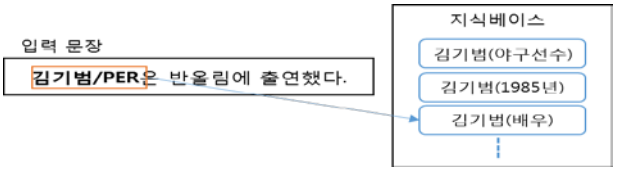


그림 1. 개체 연결의 예

3. 심층학습을 이용한 개체 연결

본 논문에서 제안하는 개체 연결 모델의 전체 구조는 (그림 2)에서 확인할 수 있다. 제안하는 개체 연결 모델은 2 개의 문서 임베딩 모델, 1 개의 선형 변환 모델 그리고 코사인 유사도로 구성된다. 개체 연결에서 연결 대

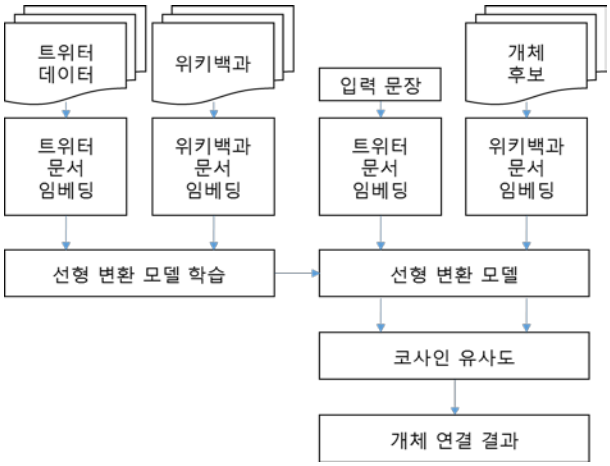


그림 2. 개체 연결 모델의 구성

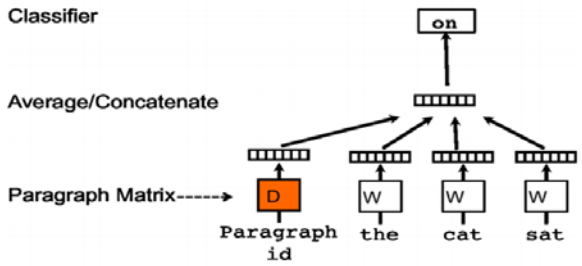


그림 3. Distributed Model의 구조

상이 되는 지식베이스의 개체로서 위키백과(wikipedia.org)의 문서를 사용하는 것을 위키화(wikification)라고 한다. 본 논문에서도 위키화를 대상으로 한다.

3.1 문서 임베딩

최근 단어 임베딩(Word Embedding)을 확장하여 자연어 문장이나 문서를 벡터 공간으로 표현하는 문서 임베딩(Document Embedding)[5]이 제안되었다. 문서 임베딩은 DM(Distributed Memory) 또는 DBOW(Distributed Bag of Word)로 구성되는데 본 논문에서는 DM 모델을 사용한다. DM 모델은 (그림 3)과 같이 각 문서에 단락 식별자(Paragraph id)와 해당 문서를 구성하는 단어 열을 입력으로 하고, 다음 단어가 Softmax 분류기에 의해 예측되는 신경망으로 구성된다. 이때 D와 W는 확률적 경사 강하법(Stochastic Gradient Descent)에 의해 학습된다. 그리고 테스트 단계에서 문서가 입력되면 Softmax 분류기와 W의 가중치(weight)를 고정한 상태에서 신경망 학습을 수행이 수행된다. 학습이 수행 된 후 단락 행렬 D가 입력 문서에 대한 문서 임베딩이 된다.

3.2 선형 변환

트위터와 같은 소셜 미디어의 문장과 위키백과 문서의 문장은 표현 방식에서 큰 차이가 있기 때문에 유사한 내용을 포함하더라도 문서 임베딩의 유사도가 낮게 나올 가능성이 높다. 본 논문에서는 위키백과 문서의 벡터 표현과 트위터 문장의 벡터 표현 사이에 선형적 사상이 가능하다는 가정을 하고 선형 변환을 수행하여 이러한 문제를 해결하고자 한다.

트위터 문장의 문서 임베딩을 T, 위키백과의 문서 임베딩 출력을 W 그리고 두 임베딩의 선형 변환 행렬을 M이라 하면 다음과 같은 식으로 표현된다.

$$TM=W \quad (1)$$

M을 학습하기 위한 T와 W를 수집하기 위해 문서 W는 위키백과 문서를 임베딩의 입력으로 사용하고, T는 해당 위키백과 문서의 제목을 포함하는 트워터 문장을 수집하여 임베딩의 입력으로 사용한다. 또한 M을 학습하기 위해 확률적 경사 강하법을 사용한다.

3.3 개체 연결을 위한 코사인 유사도

각 모델의 학습이 끝난 후 입력 문장에 대해서 개체 연결을 수행할 때 입력 문장의 문서 임베딩T_t를 선형 변환하여 위키백과 문서 임베딩 W_t로 변환한다. 그 후 각 개체 후보 문서들의 문서 임베딩 W_i에 대해서 아래의 코사인 유사도를 계산하여 가장 유사도가 높은 개체 i를 선택하게 된다.

$$\cos(W_t, W_i) = \frac{W_t \cdot W_i}{||W_t||\,||W_i||} \quad (2)$$

4. 실험

실험은 한국어 위키백과 문서와 한국어 트위터를 대상으로 수행하였다. 위키백과 문서의 문서 임베딩을 학습하기 위해 본문 내용이 200 어절 이상으로 구성된 약 32만개의 문서를 수집하였다. 트위터 문서의 문서 임베딩을 학습하기 위해 임의의 트위터 문장을 약 38만개 수집하였다. 위키백과 지식베이스는 동명이인 문서가 존재하는 인명을 대상으로 구축하였고, 트위터에서 임의의 300명 사용자가 작성한 문장 중 지식베이스의 인명이 출현한 문장 248개를 수집하여 테스트 문장으로 사용하였다. 테스트 문장에서 평균 약 3.45명의 개체 모호성이 있었다. 형태소 분석을 위해 Twitter 한국어 형태소 분석기(https://github.com/twitter/twitter-korean-text)를 사용하였다.

비교 모델은 [2]과 [3]의 모델을 재구현하여 사용하였고 실험 결과는 (표 1)과 같다. 실험 결과에서 볼 수 있듯이 기존의 모델은 비정형 문장에 대해서 성능이 크게 하락한 것을 확인할 수 있다. 반면 본 논문에서 제안하는 문서 임베딩과 선형 변환 모델은 단어 기반 모델과 비교하여 크게 개선된 성능을 보여주었다. 이러한 결과는 제안하는 문서 임베딩과 선형 변환이 트위터 문장과 위키백과 문서 사이의 표현 차이 문제를 일정 부분 해소하는 것이라 할 수 있다.

모델	정확도
링크 기반[2]	0.31
링크 + 사용자 모델[3]	0.59
문서 임베딩	0.71
문서 임베딩 + 선형 변환	0.74

표 1. 실험 결과

5. 결론

본 논문에서는 문서 임베딩과 선형 변환을 이용한 개체 연결 모델을 제안하였다. 제안하는 모델은 단어 기반 개체 연결의 단점을 보완하여 비정형 문장에 대해서도 뛰어난 성능을 보여주었다. 향후 소셜 미디어에 부착된 다양한 메타 정보(해쉬태그, 날짜와 시간, 리트윗 관계 등)의 활용 법, 인명 개체 이외의 개체에 대한 실험 등

을 수행할 계획이다.

감사의 글

본 연구는 미래창조과학부 및 정보통신기술진흥센터의 정보통신·방송 연구개발 사업의 일환으로 하였음. [R0126-16-1112, 퍼스널 미디어가 연결공유결합하여 재구성 가능케 하는 복합 모달리티 기반 미디어 응용 프레임워크 개발]

참고문헌

[1] R. Bunescu and M. Pasca, Using Encyclopedic Knowledge for Named Entity Disambiguation, in Proc. of the 11th Conference of the European Chapter of the Association for Computational Linguistics, pp. 9-16, 2006.

[2] E. Charton, M. J. Meurs, L. Jean-Louis and M. Gagnon, Mutual Disambiguation for Entity Linking, in Proc. of the 52nd Annual Meeting of the Association for Computational Linguistics, pp. 476-481, 2014.

[3] W. Shen, J. Wang, P. Luo and M. Wang, Linking Named Entities in Tweets with Knowledge Base via User Interest Modeling, in Proc. of the 19th SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 68-76, 2013.

[4] 정소윤, 박영민, 강상우, 서정연, “유저 모델과 실시간 뉴스 스트림을 사용한 트윗 개체 링킹”, 인지과학, 제26권, 2제4호, pp. 435-452, 2015, 12.

[5] Q. Le and T. Mikolov, Distributed Representations of Sentences and Documents, In in Proc. of the 31st International Conference on Machine Learning, pp. 1188-1196, 2014.

색인어 정규화 및 응답 필터링을 이용한 검색기반 채팅 모델

이현구○, 김민경, 김진태, 김학수, 이연수*, 최맹식*
강원대학교 컴퓨터정보통신공학과, ㈜엔씨소프트*
nlphglee@kangwon.ac.kr, kmink0817@kangwon.ac.kr, wlsxo1119@kangwon.ac.kr, nlpdrkim@kangwon.ac.kr,
yeonsoo@ncsoft.com, mschoi@ncsoft.com

Retrieval-based Chat Model using Index-Term Normalization and Answer Filtering

Hyeon-gu Lee○, Minkyoung Kim, Jintae Kim, Harksoo Kim, Yeonsoo Lee*, Maengsik Choi*
Kangwon National University Computer and Communication Engineering
NCSOFT Corp.*

요 약

채팅 모델은 인간과 컴퓨터가 신변잡기 대화를 나눌 수 있게 해주는 시스템으로 빠른 속도로 발전하는 인공지능 음성언어 비서 시스템에 필수적으로 사용되는 기술이다. 본 논문에서는 검색기반 채팅 모델에서 발생하는 검색 효율 문제와 정확하지 못한 답변을 출력하는 문제를 해결하기 위해 색인어 정규화와 응답 필터링이 적용된 검색기반 채팅 모델을 제안한다. 색인어 정규화를 통해 99.3%의 색인 커버리지를 확보하였으며 필터링 모델을 통해 기존 검색 모델에서보다 향상된 사용자 만족도를 얻었다.

주제어: 채팅 모델, 색인어 정규화, 문장 임베딩, 필터링 모델

1. 서론

최근 애플의 시리(Siri), 아마존의 알렉사(Alexa)는 물론 국내 SKT 누구(Nugu), KT 기가지니(GIGA Genie), 네이버 클로바(Clova)와 같은 인공지능 음성언어 비서 시스템이 활발히 연구되고 있으며 관련 시장이 빠른 속도로 성장하고 있다[1]. 이러한 인공지능 음성언어 비서 시스템에서 인간과 컴퓨터가 신변잡기 대화를 나눌 수 있도록 하는 채팅 모델은 가장 필수적인 기술로 입력된 문장을 통해 적절한 답변을 출력하는 시스템이다. 채팅 모델은 보유한 대화쌍에서 가장 유사한 내용을 선별하는 검색기반 모델과 입력된 문장을 통해 답변문장을 생성하는 생성기반 모델이 있다. 검색기반 모델은 제한된 데이터를 효과적으로 검색하고 응답의 정확도를 높여야하며 생성기반 모델은 문법적 오류 및 의미적 정확성을 향상시켜야 하는 이슈를 가지고 있다. 본 논문에서는 검색기반 채팅 모델의 이슈를 해결하기 위해 색인어 정규화 및 응답 필터링을 적용하여 검색 커버리지와 검색 결과 정확도를 향상시키는 검색기반 채팅 모델을 제안한다.

2. 관련 연구

검색기반 방식의 연구는 채팅 모델뿐만 아니라 질의응답에서도 많이 사용되고 있으며 검색 커버리지를 높이기 위해 질의를 정규화[2]하거나 질의확장[3]을 통해 어휘 일치율을 향상시키는 연구가 진행되었다. 또한 어휘 정보만 사용하는 기존의 검색 모델과 달리 의미정보를 함께 반영하고 구문 패턴을 통해 말잇기를 하는 등 답변 문장의 품질을 향상시키기 위한 채팅 모델 연구도 진행되었다[4]. 검색 모델 외에도 질의를 분석하여 얻어진 자질을 통해 검색 결과를 재순위화 하여 문장의 품질을 향상시키는 후처리 방식의 연구도 진행되고 있다[5]. 본 논문에서는 검색의 커버리지를 향상시키기 위해 개체명과 시제, 보조 용언의 양상 정보를 통한 색인어 정규화를 사용하고 문장 임베딩을 활용한 응답 필터링을 통해 응답의 정확도를 향상시키는 검색기반 채팅 모델을 제안한다.

3. 검색기반 채팅 모델

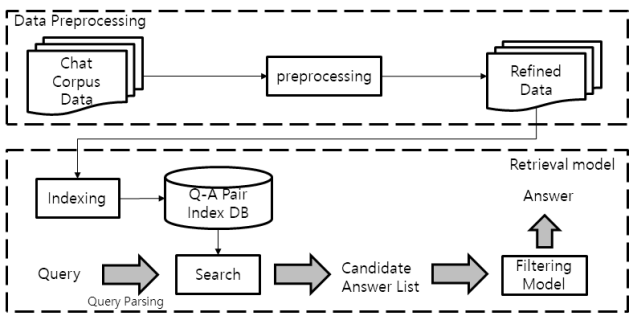


그림 1. 제안 모델의 구조도

[그림 1]은 제안 모델의 구조도를 보여준다. 본 논문에서 제안하는 검색기반 채팅 모델은 주어진 색인 질의/

응답 쌍 데이터를 전처리 및 정규화하여 색인하는 색인 모델, 사용자 질의가 입력됐을 때 사용자 질의와 유사한 색인 질의를 찾고 색인 질의와 쌍으로 존재하는 응답을 검색하는 검색 모델, 마지막으로 검색된 후보 응답이 적절한지를 판단하는 응답 필터링 모델로 구성된다. 색인 모델과 검색 모델에서 사용하는 색인어 정규화는 개체명, 보조 용언, 시제를 통해 생성하고 후보 응답이 적절한지 판단하는 응답 필터링 모델은 auto-encoder방식의 sequence-to-sequence 모델을 통해 사용자 질의, 색인 질의, 색인 응답의 문장 임베딩을 생성하고 생성된 문장 임베딩을 통해 이진 분류하여 사용자 질의와 응답이 적합한지를 판단한다.

3.1. 색인어 정규화

본 논문에서는 검색의 커버리지(coverage)를 높이기 위해 색인 질의와 사용자 질의 색인어를 정규화한다. 정규화 작업은 형태적으로나 구문적으로 상이한 문장이라도 동일한 색인어로 변환하는 기술이다. 본 논문은 다음과 같은 순서로 정규화를 적용한다.

1) 연속된 기호(. ? !)를 하나의 기호로 변환
2) 자주 사용되지 않는 기호를 제거
3) 질문 패턴 사전을 통해 매칭 된 “~까”, “~나”, “?” 등을 질문을 나타내는 심볼 “@Q”로 변경
4) 형태소 분석결과 미등록어(NA)로 구분된 경우 의미를 유추할 수 있는 앞 세글자만 추출
5) 개체명 인식결과를 통해 개체명에 해당하는 형태소를 개체명으로 치환
6) 보조 용언과 시제를 통해 양상(modality)을 파악하고 양상별 심볼로 정규화
7) 체언류, 용언류의 내용어(content word)가 아닌 형태소 제거

정규화 과정 3)의 질문 패턴 사전은 질문을 표현할 때 사용되는 어휘들을 기록해둔 사전으로 질의에서 사전에 매핑되는 어휘가 나타날 시 치환하는 역할을 한다. 4)의 미등록어 처리 부분은 잘못된 어휘나 출현 빈도가 낮은 어휘의 커버리지를 높이기 위한 부분으로 “안녕하세욤”과 같이 잘못된 어휘로 인한 미등록어를 “안녕하”로 치환하여 커버리지를 높인다. 6)의 양상별 심볼은 형태소가 조합되면서 나오는 양상을 그룹화[6]하여 “밥을 먹고 싶다.”와 “밥을 먹기 바란다.” 같이 의미는 같지만 형태가 다른 질의를 정규화 해준다. [그림 2]는 색인어 정규화를 통해 생성된 색인어를 보여준다.

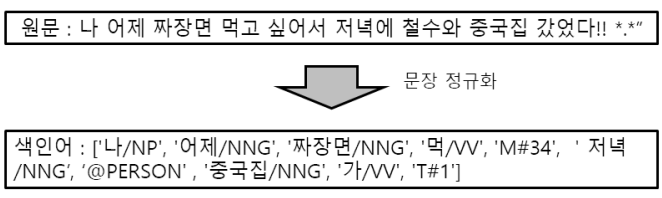


그림 2. 문장 정규화의 예

[그림 2]에서 색인어 정규화를 통해 생성된 색인어 중 “M#34”는 희망을 나타내는 보조 용언이 치환된 것이고 “T#1”은 과거를 나타내는 시제이다. 또한 “철수”의 경우 개체명 인식결과 사람을 나타내는 “@PERSON”으로 분류되어 형태소 “철수/NNP”를 개체명 “@PERSON”으로 변경한다.

3.2. 검색 모델

검색 모델을 사용하기에 앞서 색인을 위한 색인 데이터의 구조는 [그림 3]과 같다.

명칭	의미
ID	색인 질의/응답 쌍의 ID
Question	색인 질의의 원본 문장
Answer	색인 응답의 원본 문장
Index Term	색인 질의를 정규화하여 표현된 색인어

그림 3. 색인 데이터의 구조

[그림 3]의 구조에서 Index Term은 색인 질의를 3.1절에서 언급한 정규화 방법을 통해 생성된 색인어로 검색 모델에서 검색 할 때 사용되는 색인 정보이다. 나머지 정보는 검색이 됐을 때 필요에 따라 정보를 가져오기 위해 같이 색인된 정보로 검색할 때 사용되지 않는 정보이다.

질의/응답 데이터를 색인 한 후 사용자 질의와 유사한 색인 질의를 검색한다. 본 논문에서는 사용자 질의에 가장 유사한 색인 질의/응답 쌍을 검색하기 위해 식 (1)의 okapi BM25를 사용하여 사용자 질의와 색인 질의간의 유사도를 계산한다.

$$score(Q,S)=\sum_{i=1}^{n} IDF(q_i)\cdot\frac{f(q_i,S)\cdot(k_i+1)}{f(q_i,S)+k_1\cdot(1-b+b\cdot\frac{|S|}{avgdl})} \qquad (1)$$
$$IDF(q_i)=\log\frac{N-n(q_i)+0.5}{n(q_i)+0.5}$$

식 (1)에서 Q는 사용자 질의, S는 색인 질의, q_i는 사용자 질의에 포함된 i번째 단어, $avgdl$은 평균 문장 길이, N은 색인된 전체 질의의 수이다. 이때 초모수(hyper parameter)로 사용되는 k_1과 b는 문장 검색에 최적화된 1.0과 0.18로 설정한다[7].

3.3. 응답 필터링 모델

본 논문에서는 검색 모델을 통해 검색된 결과가 사용자 질의에 알맞은 응답이 아닌 경우를 감소시키기 위해 문장 임베딩을 사용하는 응답 필터링 모델을 제안한다. 응답 필터링 모델에 사용되는 문장 임베딩 모델은 입력 문장으로부터 생성되는 문장이 입력과 동일하게 나오도록 하는 auto-encoder방식[8]의 sequence-to-sequence를

모델을 사용한다. [그림 4]는 auto-encoder방식의 문장 임베딩 생성 모델의 구조이다.

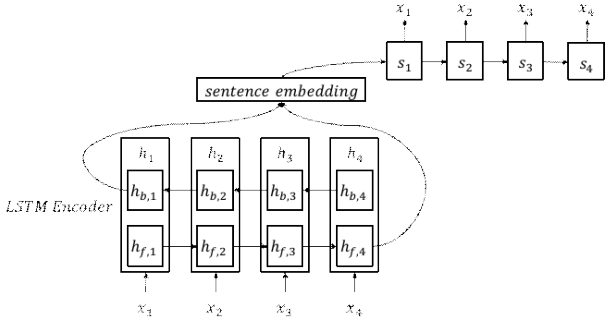


그림 4. Auto-encoder방식의 문장 임베딩 생성 모델

[그림 4]에서 보는 것과 같이 양방향 LSTM 순환 신경망(Bi-directional LSTM Recurrent Neural Network)[9]을 통해 입력 문장을 임베딩하고 생성된 임베딩을 디코딩하여 입력과 똑같은 문장이 생성되도록 학습한다. 다음으로 새로 입력된 문장의 문장 임베딩을 생성하고자 할 땐 양방향 LSTM 순환 신경망을 통해 생성된 인코딩 정보만을 문장 임베딩으로 사용한다. 문장 임베딩 모델은 응답 필터링 모델에 적용하기 전 사전 학습하여 사용하며 600,000개의 구어체 자막 데이터를 사용하여 학습한다.

필터링 모델은 사용자 질의, 색인 질의, 색인 응답의 문장 임베딩을 통해 사용자 질의와 검색된 후보 응답이 적절한지 여부를 판단하는 이진 분류 신경망을 사용한다. [그림 5]는 문장 임베딩을 사용하는 이진 분류 모델을 나타낸다.

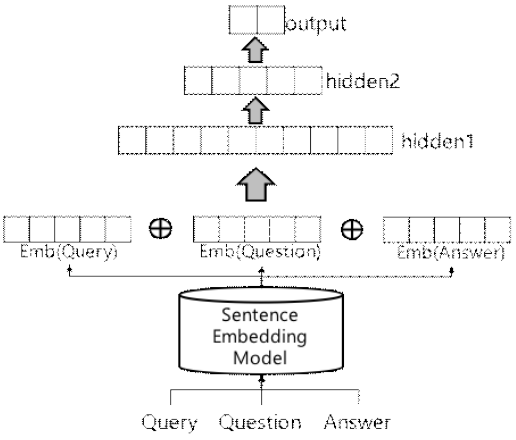


그림 5. 응답 필터링 모델

[그림 5]에서 Query는 사용자 질의, Question은 색인 질의, Answer는 색인 응답을 나타낸다. 사용자 질의, 색인 질의, 색인 응답을 각각 문장 임베딩 모델을 통해 임베딩을 생성하고 생성된 임베딩을 연결(concatenation)한 후 신경망 모델을 통해 이진 분류한다.

4. 실험 및 평가

4.1. 실험 준비

본 논문은 필터링 모델과 검색기반 채팅 모델 두 가지 성능을 평가한다. 필터링 모델을 학습 및 평가하기 위해 색인 데이터의 질의/응답 쌍을 positive 데이터, 색인 질의 Q를 검색하여 나온 검색 순위 4위 이하의 질의/응답 Q' /A' 로 만든 Q/A' 를 negative 데이터로 사용한다. 학습 데이터 82,814개, 평가 데이터 20,704개를 사용하며 총 103,518개의 데이터는 positive 데이터 49,417개, negative 데이터 54,101개로 구성된다. 다음으로 검색기반 채팅 모델의 색인 및 성능을 평가하기 위해 색인 커버리지를 확인하는 close테스트에 54,252개의 질의/응답 쌍 데이터, 사용자 만족도를 확인하기 위한 open테스트에 색인 데이터에 포함되지 않은 200개의 질문을 사용한다.

4.2. 응답 필터링 모델 실험 평가

본 논문에서 응답 필터링 모델의 성능을 평가하기 위해 정확도(accuracy), positive 레이블의 정확률(precision), 재현율(recall), F1-score를 사용한다. 표 1은 필터링 모델의 성능을 나타낸다.

표 1. 필터링 모델의 성능

	Performance
Accuracy	0.983
Positive Precision	0.967
Positive Recall	1.0
Positive F1-score	0.984

표 1에서 F1-score를 보아 사용자 질의와 검색된 색인 응답을 필터링 하는데 높은 성능을 보이고 있어 정확한 답변을 해야 하는 검색기반 채팅 모델에 효율적으로 사용할 수 있을 것으로 판단된다.

4.3. 검색기반 채팅 모델 실험 평가

검색기반 채팅 모델은 사용된 색인 커버리지를 확인하기 위한 close테스트와 채팅 모델의 사용자 만족도를 평가하기 위한 open테스트 두 가지를 평가한다. close테스트는 색인 질의를 검색했을 때 1순위로 검색된 질의가 검색 질의와 같은 여부를 측정한다. close테스트의 성능은 표 2와 같다.

표 2. 검색기반 채팅 모델 close 테스트 성능

	Accuracy
기본 검색 모델	0.989
필터링 모델 추가	0.993

표 2에서 기본 검색 모델은 3.2절까지 모델로 필터링 모델이 적용되지 않은 모델이며 필터링 모델 추가는 3.3절의 필터링 모델을 적용시킨 검색기반 채팅 모델이다. close테스트지만 약간의 오류가 있는데 이는 일부 문자가 다르지만 정규화된 질의어가 같아 발생하는 오류다.

다음으로 open테스트는 색인 질의에 포함되지 않은 200개의 질의를 검색해 나온 결과를 사람이 직접 정성평가한 결과이다. 정성평가는 1점부터 5점까지의 범위로 평가한다. 본 논문에서 정성평가의 기준은 다음과 같다.

1점 - 응답이 전혀 없거나 말이 되지 않는 것
2점 - 질의에 대한 응답으로 조금 적절치 못한 것
3점 - 질의에 대한 응답으로 될 수도 있는 것
4점 - 질의에 대한 응답으로 괜찮은 것
5점 - 질의에 대한 완벽한 응답

정성평가는 4명의 평가자가 진행하며 표 3은 정성평가의 연구자별 평점을 보여준다.

표 3. 정성평가의 성능

	기본 검색 모델	필터링 모델 추가
Human 1	3.069	3.55
Human 2	2.708	3.35
Human 3	3.335	3.6
Human 4	2.968	3.55
평균	3.02	3.51

표 3에서 기본 검색 모델은 평가자들의 평균이 3.02로 대부분의 문장이 응답이 될 수 있는 것이지만 필터링 모델을 적용할 경우 평가자들의 평균이 3.51로 향상되어 모델 응답의 품질이 상승되는 것을 알 수 있었다.

4.4. 응답 필터링 성능 향상 예시

본 논문에서 제안한 색인화 정규화 및 응답 필터링이 적용된 결과는 [그림 6]과 같다.

	질의	응답
기본 검색 모델	정말 어려운 문제야.	누가 뭐랬나요?
필터링 모델 추가		잘 할 수 있을 거예요.

그림 6. 필터링 모델을 통해 정답 품질이 상승한 예

[그림 6]은 기본 검색 모델의 응답 결과와 응답 필터링이 적용될 경우 품질이 상승된 결과 예시를 보여준다. 질의 “정말 어려운 문제야.”라고 입력했을 때 기본 검색 모델은 가장 유사한 문장으로 “정말이라고, 정말!”이 매칭되어 그에 해당하는 “누가 뭐랬나요?”라는 엉뚱한 답변을 출력한다. 하지만 필터링 모델이 적용된 경우 “정말이라고, 정말!”은 필터링되고 “정말 어려워.”와 매칭 되어 “잘 할 수 있을 거예요.”와 같이 질의와 어울리는 문장을 출력하게 해준다.

5. 결론 및 향후 연구

본 논문에서는 개체명, 보조 용언, 시제를 통해 색인어를 정규화하고 검색 결과를 필터링하여 품질을 향상시키는 검색기반 채팅 모델을 제안하였다. 실험 결과 색인어 정규화로 높은 검색 커버리지를 확보했고 문장 임베딩을 이용한 응답 필터링을 통해 응답의 품질을 향상 시킬 수 있었다. 향후 연구로 검색 모델을 통해 결과가 검색되지 않는 경우를 해결하기 위해 생성기반 채팅 모델을 결합한 하이브리드 채팅 모델을 연구할 예정이다.

감사의 글

본 연구는 엔씨소프트 산학연구용역 과제의 지원을 받아 수행되었음.

참고문헌

[1] 김학수, “인공지능 음성언어 비서 시스템의 자연언어처리 기술들”, *정보과학회지*, 35.8 , pp. 9-18, 2017.

[2] A. B. Abacha and P. Zweigenbaum, Medical question answering: translating medical questions into sparql queries, *Proceedings of the 2nd ACM SIGHIT International Health Informatics Symposium.* ACM, pp. 41-50, 2012.

[3] A. R. Aronson and T. C. Rindflesch, Query expansion using the UMLS Metathesaurus, *Proceedings of the AMIA Annual Fall Symposium. American Medical Informatics Association,* pp. 485, 1997.

[4] 전원표, 송영길, 김학수, “채팅 모델 구현을 위한 3단계 문장 검색 방법”, *한국마린엔지니어링학회지 제37권 제2호*, pp. 205-212, 2013.

[5] 이현구, 김민경, 김학수, “의학문서 질의응답을 위한 정답 스닛핏 검색”, *정보과학회논문제 제43권 제8호*, pp. 927-932, 2016.

[6] 안동언, “Corpus를 기반으로 하는 한국어 술어의 양상 생성”, *KAIST 박사학위논문*, 1995.

[7] R. Blanco, H. Zaragoza, Finding support sentences for entities, *Proceedings of the 33rd international ACM SIGIR conference on Research and development in information retrieval*, ACM, pp. 339-346, 2010.

[8] Y. Bengio, Learning Deep Architectures for AI, *Foundations and trends® in Machine Learning*, 2(1), pp. 1-127, 2009.

[9] M. Schuster, K. K. Paliwal, Bidirectional recurrent neural networks, *IEEE Transactions on Signal Processing*, 45(11), pp. 2673-2681, 1997.

버전 상호 호환 가능한 HL7 파서의 설계

이인근, 황도삼
대구도시철도공사, 영남대학교
inkeunlee@gmail.com, dshwang@yu.ac.kr

Design of an Version-Compatible HL7 Parser

In Keun Lee, Dosam Hwang
Daegu Metropolitan Transit Corporation, Yeungnam University

요 약

의료정보시스템의 상호운용을 위해 개발된 의료정보 교환 국제 표준인 HL7은 복잡한 구조와 문법으로 인해 컴퓨터 소프트웨어로 관리되고 있다. 현재 개발되고 있는 HL7 인터페이스 소프트웨어에서는 다양한 버전 간 호환이 되지 않아 의료정보시스템에서 버전 상호 간의 호환을 위해 변환 소프트웨어 모듈을 개발하여 사용한다. 그러나 다양한 버전(V2.1~V2.8)의 HL7 메시지 간 상호 변환을 위해 소프트웨어 모듈을 모두 개발하는 것은 많은 시간과 막대한 비용 및 노력이 필요한 비효율적인 방법이다. 따라서 본 연구에서는 HL7 버전 호환성 정의에 기반을 두어 버전별 상호변환이 가능한 HL7 파서(Parser)를 설계하고, 객체지향적 구조에 기반을 두어 하위 버전과의 호환(Backward Compatibility)뿐만 아니라 상위 버전과 호환(Forward Compatibility) 가능한 파서를 제안한다. 또한, 버전 간 변환 실험을 통해 효용성을 검증하였다.

주제어: HL7 V2, 버전 호환성, 객체지향 프로그래밍, HL7 인터페이스 소프트웨어

1. 서론

의료정보의 전산화를 위한 노력의 결과로서 국제적으로 전자의무기록(EMR), 평생전자건강기록(EHR) 등과 같은 의료정보의 생산과 관리를 위해 다양한 의료정보시스템이 개발됐다. 그리고 의료기관마다 독자적으로 구축되고 있는 의료정보 시스템 사이의 정보 교환을 위해 HL7[1,2], DICOM, CDA 등과 같은 다양한 의료정보 전송 표준이 사용되고 있다[3]. 특히, HL7은 이기종 시스템 사이의 의료정보 교환 목적으로 많이 사용되고 있으나, HL7 메시지는 구조와 문법이 복잡하여 컴퓨터 소프트웨어 파서(Parser)를 이용하여 처리되고 있다. HL7 메시지의 처리를 위해 Symphonia, NeoBrowse TCP, Chameleon, LINKTools, HAPI 등이 개발되었으나[2] 이들 소프트웨어는 HL7의 재사용성과 상호운용성을 보장하기 위한 버전(V2.1~V2.8) 간의 호환성이 충족되지 않아, 서로 다른 버전의 HL7 메시지를 이용하는 시스템 사이에는 버전 간 변환을 위한 새로운 모듈을 개발해야만 한다.

본 연구에서는 HL7에 정의된 버전 호환성 요구사항[4]에 따른 하위 버전과의 호환성(Backward Compatibility)뿐만 아니라, HL7 파서의 활용성을 높이기 위해 상위 버전과의 호환성(Forward Compatibility)을 보장하는 HL7 파서의 구조를 설계한다. 또한, 제안한 구조를 기반으로 ADT_A01 메시지를 분석할 수 있는 HL7 파서를 구현하고, 버전 간의 호환성 실험을 수행하여 효용성을 확인한다.

2. 관련 연구

HL7 버전 호환성 정의[4]는 (1)상위 버전(예:V2.4)의 메시지를 수신하는 하위 버전(예:V2.5)의 시스템은 오류 없이 메시지를 수신할 수 있어야 하고, (2)상위 버전의 시스템은 하위 버전의 메시지를 이해할 수 있어야 한다. 즉, HL7에서 하위 버전의 구조는 상위 버전의 구조에 포함되므로 상위 버전의 메시지를 하위 버전의 메시지로 정보 손실 없이 변환 가능하여야 한다. 그러나 실제로는 버전 간의 상이한 구조로 인해 이미 개발된 파서들에서는 버전 호환성을 충족하지 못하고 있다. 이에 [3,5]에서는 상위 버전에서 하위 버전으로의 변환이 가능한 파서가 설계되었다. 그러나 하위 버전에서 상위 버전으로의 변환을 통해 Forward Compatibility를 보장하는 파서는 전무하다.

3. 버전 상호 호환 가능한 HL7 파서의 설계

버전 간 호환 가능한 파서의 구조를 설계하기 위하여 객체지향 프로그래밍의 상속 개념을 이용하였다. 즉, 그림 1과 같이 하위 버전의 HL7 Message 클래스 객체를 상위 버전 클래스 객체가 상속하도록 함으로써, HL7 메시지의 버전에 무관하게 최하위 클래스 객체만을 사용하여 모든 버전의 HL7 메시지를 처리할 수 있도록 하였다. 예

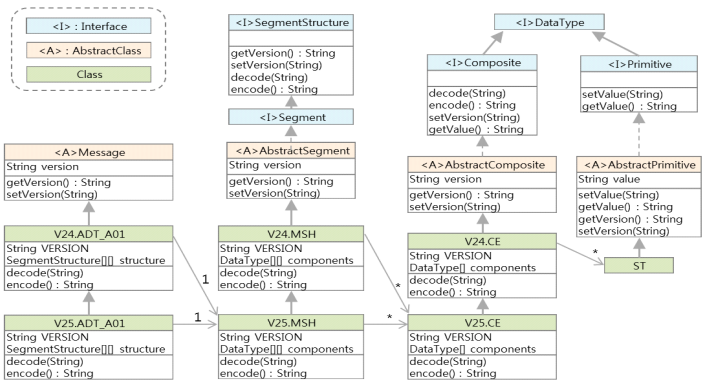


그림 1. 객체지향형 HL7 파서의 구조

를 들어, 그림 2와 같이 HL7 V2.4의 ADT^A01 메시지를 처리한다면, 파서는 메시지의 실제 버전을 먼저 확인한 후, 해당 버전의 클래스인 V24.ADT_A01 클래스 객체에서 메시지가 처리된다. V24.ADT_A01 클래스는 MSH Segment 객체를 포함하는데, 이 또한 Segment 최하위 클래스 객체인 V25.MSH에 처리를 요청하고, 실제 메시지의 버전에 따라 V24.MSH 클래스 객체에서 HL7 메시지를 전달하여 처리하도록 하였다. 이러한 과정을 통해 처리된 데이터는 해당 버전의 클래스 객체 내의 데이터로 저장된다.

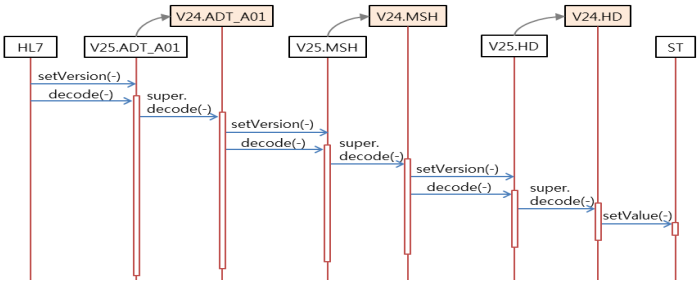


그림 2. HL7 V2.4 메시지 처리 과정

그림 3은 상위 버전(V2.5)의 메시지를 하위 버전(V2.4)으로 변환하는 과정을 보인다. 즉, HL7 V2.5 메시지를 처리한 V25.ADT_A01 객체에서 버전 설정(V2.4) 후 encode() 메소드를 호출하였으며, 버전에 따라 V25.ADT_A01의 부모클래스인 V24.ADT_A01의 encode() 메소드를 호출한다. 또한, 동일한 방법으로 V24.MSH, V24.HD의 encode() 메소드를 호출함으로써 설정된 버전의 메시지를 생성한다. 이는 하위 버전과의 호환성(Backward Compatibility)을 위해 객체지향의 Method overriding을 이용한 방법이다.

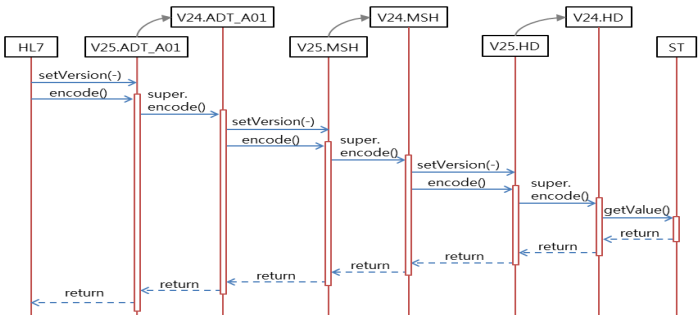


그림 3. HL7 V2.4 메시지 생성 과정

그림 4는 하위 버전(V2.4)의 메시지를 상위 버전(V2.5)으로 변환하는 과정을 보인다. 최하위 클래스인 V25.ADT_A01의 객체에서 encode() 메소드를 호출하면, HL7 V2.4 메시지를 처리한 V24.ADT_A01 객체의 encode() 메소드를 호출하고, 설정된 버전(V2.5)에 따라 V24.ADT_A01 객체 내에 저장된 데이터를 V25.ADT_A01 객체에 적합한 데이터로 변환하여 저장한다. 그리고 다시 V25.ADT_A01객체의 encode() 메소드를 호출하여 설정된 버전의 메시지를 생성한다. 이는 상위 버전과의 호환성(Forward Compatibility)을 위해 객체지향의 Casting을 이용한 방법이다.

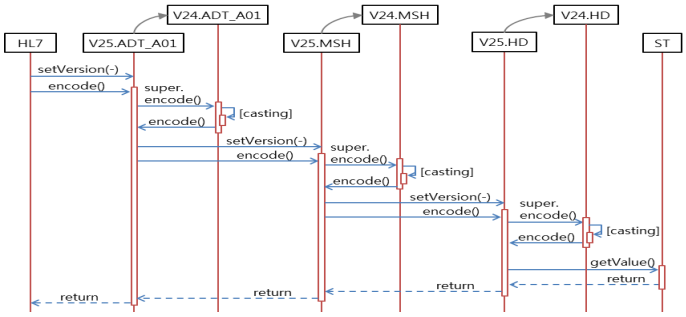


그림 4. HL7 V2.4→V25 메시지 변환 과정

제안한 설계에 따라 ADT_A01 메시지의 처리를 위한 파서를 Java 프로그래밍 언어로 구현하고, 두 가지 버전(V2.4, V2.5)의 HL7 메시지에 대한 상호 변환 실험을 수행하였다. 다음 MSH Segment의 버전 간 상호 변환 결과에서와 같이 버전 별 Segment 규칙에 따라 메시지가 변환됨을 확인하였다.

• **Backward Compatibility Test (V2.5 → V2.4)**
V2.5 : MSH|^~ \ &|MegaReg|UABHospC|ImOrdMgr|UABImgCtr| 20010529090131-0500||ADT^A01^ADT_A01|01052901|P|2.5
→V2.4 : MSH|^~ \ &|MegaReg|UABHospC|ImOrdMgr|UABImgCtr| 20010529090131-0500||ADT^A01^ADT_A01|01052901|P|2.4
• **Forward Compatibility Test (V2.4 → V2.5)**
V2.5 : MSH|^~ \ &|MegaReg|UABHospC|ImOrdMgr|UABImgCtr| 20010529090131-0500||ADT^A01|01052901|P|2.4
→V2.4 : MSH|^~ \ &|MegaReg|UABHospC|ImOrdMgr|UABImgCtr| 20010529090131-0500||ADT^A01^""|01052901|P|2.5

4. 결론

본 연구에서는 버전 상호 호환을 위한 객체 지향적 구조에 기반을 두어 HL7 파서를 설계하였고, 실험을 통해 제안한 구조의 효용성을 확인하였다. 제안한 방법을 의료정보시스템에 적용하기 위해서는 객체 내에 저장된 데이터의 입출력을 위한 메소드의 효과적인 구현에 대한 추가 연구가 필요하다.

참고문헌

[1] HL7, Health Level Seven, Available from: http://www.hl7.org(Sep.2017)
[2] HL7 Korea, Health Level Sevel(HL7)과 개발도구, 한국보건산업진흥원, 2002.
[3] 박현상, 김화선, 조훈, "호환 가능한 HL7 파서의 개발", 한국산학기술학회논문지, 제15권, 제7호, pp.4290-4300, 2014.
[4] Health Level Seven Inc. HL7 Messaging Standard V2.5: An Application Protocol for Electronic Data Exchange in Healthcare Environments, 2003.
[5] 박현상, 이인근, 김화선, 조훈, "이전 버전과 호환 가능한 HL7 파서의 설계", 2013 KSMI 춘계학술대회, 2013.

이 논문은 2017년도 정부(미래창조과학부)의 재원으로 한국연구재단의 지원을 받아 수행된 기초연구사업임(2017R1A2B4009410)

오타에 강건한 자모 조합 임베딩 기반 한국어 품사 태깅

서대룡○, 정유진, 강인호

네이버

daeryong.seo@navercorp.com, youjin.chung@navercorp.com, once.ihkang@navercorp.com

A typing error-robust Korean POS tagging using Hangul Jamo combination-based embedding

Dae-Ryong Seo○, Youjin Chung, Inho Kang

Naver Corporation / NLP

요 약

본 논문은 한글 자모 조합 임베딩을 이용하여 오타에 강건한 한국어 품사 태깅 시스템을 구축하는 방법에 대해 기술한다. 최근 딥 러닝 연구가 활발히 진행되면서 자질을 직접 추출해야 하는 기존의 기계학습 방법이 아닌, 스스로 자질을 찾아서 학습하는 딥 러닝 모델을 이용한 연구가 늘어나고 있다. 본 논문에서는 다양한 딥 러닝 모델 중에서 sequence labeling에 강점을 갖고 있는 bidirectional LSTM CRFs 모델을 사용하였다. 한국어 품사 태깅 문제에서 일반적으로 사용되는 음절 임베딩은 약간의 오타에도 품사 태깅 성능이 크게 하락하는 한계가 있었다. 따라서 이를 개선하기 위해 본 논문에서는 한글 자모 임베딩 값을 조합시킨 음절 임베딩 방식을 제안하였다. 강제로 오타를 발생시킨 테스트 집합에서 실험한 결과, 자모 조합 임베딩 기법이 word2vec 음절 임베딩 방식에 비해 형태소 분할은 0.9%, 품사 태깅은 3.5% 우수한 성능을 기록하였다.

주제어: 한글 자모 조합 임베딩, 품사 태깅, Bidirectional LSTM CRFs

1. 서론

한국어 형태소 분석 및 품사 태깅은 주어진 어절을 형태소 단위로 분리하고 원형을 복원한 후, 각 형태소에 적절한 품사를 부여하는 과정이다. 한국어 형태소 분석은 기계번역, 음성인식, 개체명 인식을 비롯한 많은 자연어처리 응용 분야에서 필수적으로 사용되는 중요한 기술이다. 그러나 형태소 분석기를 구축하기 위해서는 다량의 언어 지식과 자원을 필요로 하기 때문에, 최근에는 기계 학습 방법론에 기반하여 형태소 분석 단계를 생략하고 바로 품사 태깅 시스템을 구축하려는 연구가 많이 시도되고 있다[1,2,4-7].

특히 의미 있는 자질을 직접 추출해야 하는 기존 기계학습 방법론과는 달리, 딥 러닝 모델은 자질 선정 과정 없이 스스로 의미 있는 자질을 찾아서 학습하고, 간단한 모델만으로도 기존 방법론을 뛰어넘는 우수한 성능을 낼 수 있기 때문에 최근 연구 방향에서 큰 흐름을 차지하고 있다.

딥 러닝 모델 기반의 품사 태깅 시스템들은 주로 end-to-end 방식으로 사전 정보나 자질 정보를 이용하지 않는 seq2seq 모델을 사용하거나[4,5], 또는 B(begin), I(inside) 태그를 각 음절에 맞춰서 부착하는 sequence labeling 방식을 사용한다[6,7]. 또한 품사 태깅 과정은 대부분 음절 단위로 처리를 진행하게 되는데, 이 때 음절 임베딩 값은 주로 대규모 말뭉치 상에서 pre-training 시켜 사용하게 된다. 이렇게 구축한 음절 임베딩 값에서는 서로 다른 두 음절은 완전히 다른 값을 갖고 있기 때문에 음절 간 구분이 명확해지는 장점이 있어서 일반적인 경우에는 좋은 성능을 보여줄 수 있다.

그러나 최근에는 폭발적으로 증가하는 신조어 수에 비례하여 동일한 개체명을 다양한 형태로 표기하는 사용자들이 많이 증가하게 되었으며 (예: 어벤져스, 어벤저스), 모바일 기기의 사용 빈도가 늘어남에 따라 오타의 발생 빈도 역시 높은 비율로 증가하게 되었다 (예: 햇어요). 음절 임베딩 값을 사용하는 경우에 '져'와 '저'와 같이 비록 음절의 자모 구성이 거의 유사한 음절들이더라도 두 음절의 임베딩 값이 서로 다르므로 완전히 다른 음절로 취급된다. 따라서 단어의 음절 상에 약간의 변화만 발생해도 (예: 어벤져스 -> 어벤저스) 기존에 학습한 적이 없는 생소한 단어가 되기 때문에 엉뚱한 품사 태깅 결과를 생성할 가능성이 높아지게 된다.

따라서 본 논문에서는 단어의 다양한 변이형 및 오타에 강건한 품사 태깅 시스템을 구축하기 위해 한글의 구성 원리에 착안하여 초성, 중성, 종성으로 구성된 자모 조합 기반의 임베딩 방식을 사용하였다. 자모의 구성이 비슷한 음절은 벡터 공간상에서 코사인 유사도가 높기 때문에, 단어에 일부 오타가 있더라도 매우 유사한 임베딩 값을 갖게 된다. 실험 결과, 강제로 오타를 발생시킨 테스트 집합에서 자모 조합 임베딩 방식은 word2vec을 이용해서 값을 생성 시킨 음절 임베딩 방식 대비 형태소 분할은 0.9%, 품사 태깅은 3.5% 높은 정확률을 기록하여 오타에 강건한 특성을 확인할 수 있었다.

2. 관련 연구

기존 품사 태깅 문제에서는 CRF 또는 SVM 기반 기계학습 방법이 많이 사용되었다. CRF와 SVM은 다양한 자질을 추출하고 이를 조합함으로써 우수한 성능을 얻을 수 있는 장점이 있으나, 자질을 추출하는 과정 자체가 어렵고 많은 비용을 요구하는 작업이기 때문에 어려움이 있었다 [1-2].

반면, 딥 러닝 모델은 별도의 자질을 추출하지 않아도 학습 과정에서 입력에 대한 추상화가 이루어지고 각각의 은닉층에서 특징에 대한 자질을 스스로 학습 한다. [3]은 미리 형태소 분할을 한 뒤 SENNA + CRF 모델을 이용하여 품사를 태깅하는 방법을 제시하였다. 형태소 단위의 단어 임베딩을 이용한 결과 98.23%의 정확률을 얻었으나, 미등록 형태소가 존재하면 성능이 낮아지는 문제가 있었다.

[4,5]는 seq2seq 모델을 이용한 방법을 제안하였다. RNN 기반이며 인코더와 디코더로 구성되어 있는 Seq2seq 모델은 특히 기계번역에서 많이 사용되는 모델이다. 또한 end-to-end 방식이기 때문에 형태소 후보 생성, 기분석 사전, 원형복원 사전 없이도 형태소 분석이 가능하다는 장점이 있다. [4]와 [5]는 음절 단위 임베딩과 attention 기법을 사용하여 seq2seq를 구현한 점에서 서로 유사하나, 각각 입력, 출력의 형식이 다르고, beam search의 이용 여부에서 서로 차이가 있다. Beam search는 각 단계에서 하나의 출력을 내는 게 아닌 top N개의 출력을 내보내는 방식이다. [5]의 실험 결과, beam search를 도입하고, 공백 정보까지 사용하면 태깅 성능 향상에 도움이 됨을 보였다.

[6]은 음절 기반의 bi-LSTM-CRFs 모델을 제안하고 있다. 음절의 임베딩 정보는 64차원으로써 word2vec으로 학습했으며, 추가로 음절의 품사 분포 벡터를 구축하여 태깅 모델에 반영하였다. 음절 임베딩 값을 bi-LSTM모델에 넣어 음절 단위로 품사 태깅을 진행한 후, 기분석 사전과 원형복원 사전을 이용하여 복합 형태소의 품사를 결정한다. 세종코퍼스 40만 어절 상에서 테스트 한 결과, 97.09%의 정확률을 기록하였다.

본 논문 역시 [6]과 마찬가지로 bi-LSTM-CRFs 모델을 사용하고 있으나, 자모 임베딩의 조합을 통해 음절 임베딩 값을 구성한다는 점에서 [6]과 차이가 있다.

3. 자모 조합 임베딩에 기반한 bi-LSTM-CRFs 품사 태깅

본 논문에서는 sequence labeling 문제 해결에 좋은 성능을 보이는 것으로 잘 알려진 bi-LSTM-CRFs 모델을 사용하였다. 학습은 문장 단위로 진행하였으며, 각 음절들은 초성, 중성, 종성의 자모들로 분할한 후, 이들 자모들의 임베딩 값을 조합하여 음절 임베딩 값을 구성하였다. 이후 음절 단위로 각 품사 태그와 조합된 B(beginning), I(inside), E(end) 태깅을 수행하게 된다.

3.1 Bidirectional LSTM CRFs 모델

Bidirectional LSTM CRFs 의 모델은 그림 1과 같이 구성된다. RNN 기반 모델은 이전 입력에 대한 결과가 현재의 입력에도 영향을 준다는 점에서 특징이 있다. 특히 품사 태깅 문제와 같이, 주변의 단어와 음절이 현재 단어와 음절 태깅에 영향을 주는 경우, 다른 딥 러닝 모델들에 비해 RNN 모델이 보다 좋은 효과를 기대할 수 있다. 이전 음절 정보와 다음 음절 정보를 모두 참조하여 현재 음절의 품사 결정에 사용하기 위해 bidirectional RNN 구조를 이용하였다.

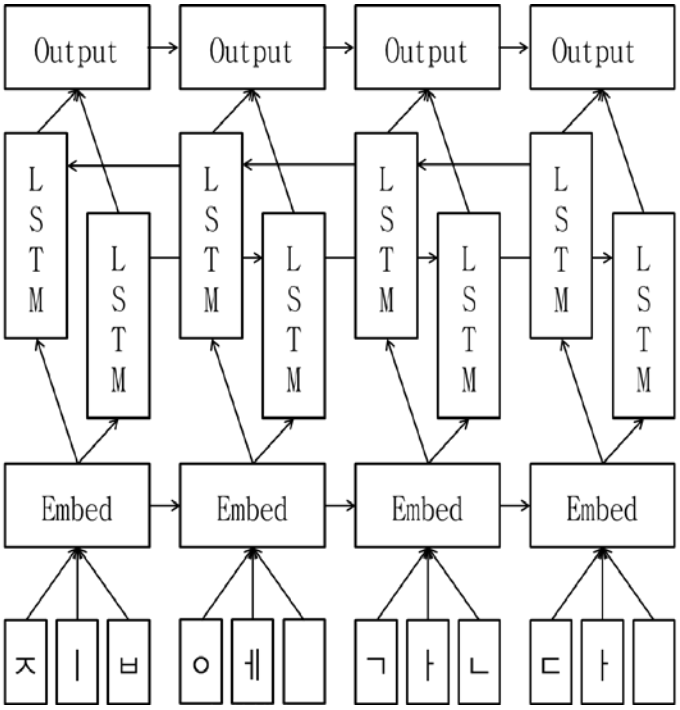


그림 1. 자모 임베딩을 이용한 Bidirectional LSTM CRFs 모델

문장 전체를 대상으로 각 음절마다 임베딩 값을 계산하여 bidirectional RNN 모델의 입력으로 넣으면 forward, backward 두 단계로 출력 값 계산이 이루어진다. 최종적으로 각 음절의 출력 값과 Viterbi 알고리즘을 이용하여 태그 사이의 전이 확률 값을 구하고 태그를 결정하게 된다. 4절의 실험 결과에서는 CRF를 사용한 모델과 사용하지 않은 모델의 성능도 함께 비교하였다.

RNN 모델은 동일한 구조의 cell이 연속적으로 이어져 있는 구조이기 때문에 gradient vanishing 문제가 발생할 수 있으나, LSTM은 cell 내부에 입력, 출력, 제거를 담당하는 게이트를 갖고 있기 때문에 긴 시퀀스에 대해서도 강건한 것으로 알려져 있다[8]. 본 논문에서도 LSTM을 사용하였으며, 시퀀스 길이 100인 경우에서 문제없이 학습이 되는 것을 확인하였다.

3.2 자모 조합 기반의 음절 임베딩

한글 음절은 기본적으로 초성, 중성, 종성으로 분리할 수 있다. 모음은 오직 중성 자리에만 사용될 수 있기 때문에 모두 고유한 값을 가질 수 있지만, 자음의 경우에는

“ㄶ”과 같이 받침으로만 사용될 수 있는 것을 일부 제외하면 초성과 종성 양쪽 모두에서 사용될 수 있는 것들이 대부분이다. 만약 초성과 종성을 구분하지 않고 동일한 자소에 대해서는 동일한 임베딩 값을 부여할 경우, “안 = ㅇ+ㅏ+ㄴ”, “낭 = ㄴ+ㅏ+ㅇ”으로써 “안”과 “낭”의 음절 임베딩 값이 동일해지는 문제가 발생하게 된다. 따라서 동일한 자소 “ㄱ”이더라도 초성인지, 종성인지 여부에 따라 구분하여 서로 다른 값을 갖도록 자모 임베딩 값을 생성하였다.

또한 받침이 없는 음절들을 위해 ‘종성 없음’의 의미로 <J+NONE> 태그를 만들어 임베딩에 이용하였다. 한 음절의 임베딩 값은 식 (1)에 따라 해당 음절을 구성하고 있는 자모 임베딩 값들을 weighted sum 하여 생성한다. 본 논문에서는 $\alpha = \beta = \gamma = 1$ 값을 사용하였다.

$$\mathrm{E}(\text{음절}) = \alpha * \mathrm{E}(\text{초성}) + \beta * \mathrm{E}(\text{중성}) + \gamma * \mathrm{E}(\text{종성}) \quad (1)$$

3.3 Layer Normalization

딥 러닝 모델 학습은 워낙 많은 계산량을 필요로 하기 때문에 학습시간이 길어지게 되는데, 이 시간을 단축시키기 위한 다양한 방법론들이 연구되고 있다. [9]는 학습 데이터를 많은 부분으로 분할시켜 여러 개의 서버에서 동시에 처리하는 방법을 사용하였으나, 데이터의 이동과 학습 모듈 자체가 복잡해지는 문제가 있었다. [10]에서는 deep neural network 내부에서 학습 데이터의 평균과 표준편차에 기반한 batch normalization 을 사용하여 학습 속도를 높였다. 하지만 batch normalization은 고정된 길이의 network에서는 잘 동작하나, RNN과 같이 입력마다 시퀀스의 길이가 달라지는 모델에서는 좋은 성능을 내기 힘들며, 학습 과정과 평가 과정에서 계산 방법이 달라지는 문제가 있었다. [11]에서 사용하는 layer normalization 은 batch normalization과 유사하게 평균과 표준편차를 이용하지만, normalize 하는 위치가 각 layer에서 이루어진다는 점에서 차이가 있다.

현재 입력 값을 x^t, 이전 hidden states h^{t-1}라고 할 때, 각 layer의 입력 a^t는 식 (2)와 같이 표현된다.

$$a^t = w_{hh}h^{t-1} + W_{xh}x^t \quad (2)$$

Layer에서 입력 값의 평균과 분산을 구하는 식은 (3), (4)으로 표현 가능하고, t의 hidden output은 식 (5)에 의해 구할 수 있다. g, b의 차원은 h와 동일하고 모든 time-step에서 공유되는 값이다.

$$\mu^t = \frac{1}{H}\sum_{i=1}^{H} a_i^t \quad (3)$$

$$\sigma^t = \sqrt{\frac{1}{H}\sum_{i=1}^{H}(a_i^t - \mu^t)^2} \quad (4)$$

$$h^t = f\left[\frac{g}{\sigma^t} \odot (a^t - \mu^t) + b\right] \quad (5)$$

이렇게 각 layer의 값을 정규화시킴으로써 covariate shift 문제를 해결할 수 있다. 따라서, loss 값이 수렴하는 시간이 짧아지고 학습 속도를 개선 할 수 있다. 또한 learning rate를 설정할 때 조금 큰 값을 사용하여도 loss 값이 튀지 않고 일정하게 학습이 되는 것을 확인할 수 있었다. RNN-LSTM을 사용하고 있는 본 논문에서의 실험 결과 역시, layer normalization 적용에 의해 동일 epoch 대비 빠르게 성능이 올라가는 것을 확인 할 수 있었다.

4. 실험 및 성능 평가

4.1 자모 및 음절 임베딩

자모 조합 임베딩 방식에 사용할 자모 임베딩 값은 랜덤으로 생성하여 실험을 진행하였다. 전체 자모의 개수는 기호류를 정규화한 경우에는 77개, 기호류를 정규화하지 않은 경우에는 234개 였다.

본 논문에서 제안하는 자모 조합 임베딩 방식과의 성능 비교를 위한 음절 임베딩 데이터는 word2vec을 이용한 pre-training 방식, 랜덤 초기화 방식 두 가지 방법을 이용하여 구축하였다. Word2vec을 이용한 학습에는 뉴스 50만 문서를 사용하였다. 전체 음절의 개수는 기호류를 정규화한 경우에는 5,602개, 기호류를 정규화하지 않은 경우에는 6,034개였다.

또한, 최종 품사 태그는 NNG, NNP, VV를 비롯한 43개의 세종 코퍼스 품사에 B, I, E 태그를 조합하여 43 x 3 = 129개를 생성하였다 (예: NNG-B, NNG-I, NNG-E). 여기서 B는 형태소에서 시작을(beginning), I는 중간을(inside), E는 끝을(end) 의미한다. 그리고 추가로 문장의 처음, 끝, PADDING을 나타내는 3개의 태그를 정의하여 129 + 3 = 총 132개의 태그를 사용하였다.

4.2 학습 및 평가 데이터 구축

학습 및 평가 데이터로는 세종코퍼스 16만 문장을 이용하였으며, RNN 학습에 15만 문장, 평가에 1만 문장을 이용하였다. 본 연구에서는 색인어 추출을 위한 품사 태거 개발을 염두에 두고 진행한 관계로, 세세하게 분리된 어미 정보는 필요하지 않기 때문에 각 어미들은 앞의 어간과 결합하여 하나의 용언으로 구성하였다. 예를 들어 “가/VV + 았/EP + 다/EF”는 어간과 어미를 결합시킨 표층형 단어에 품사 정보를 부여하여 “갔다/VV”로 변환시켜 사용했다.

RNN sequence length 는 100, LSTM의 hidden layer 개수는 256, learning rate는 0.03으로 설정하였고, 최대 140 epoch까지 학습을 진행하였다.

4.3 자모 조합 임베딩 기반 태깅 성능

성능 비교는 형태소 분할만 하는 경우와(segmentation) 형태소 분할에 이어 품사 태깅까지 하는 경우(POS tagging) 두 가지 문제로 구분하여 실험하였다.

먼저 형태소 단위 분할(segmentation) 문제는 입력문 “집에 간다”에 대해 품사 정보 없이 “집/에/간다”로

형태소 분할 경계를 구분하는 것까지만을 목표로 한다. 용언은 어간과 어미를 분리하지 않고 하나의 형태소로 취급하였으며, 태그셋은 B, I, E 3개를 사용하였다.

두번째로 품사 태깅(POS tagging) 문제는 형태소 분할 결과에 품사 정보까지 추가하여 "집/NNG", "에/JKB", "간다/VV"의 결과를 얻는 것까지를 목표로 한다. 모델 학습에는 앞서 4.1절에서 설명한 132개의 태그셋을 사용하였다. 자모 임베딩 및 음절 임베딩은 모두 동일하게 256차원으로 설정하였다.

표 1은 자모 조합 임베딩을 사용하여 실험한 결과를 보여주고 있다. 기존의 논문들에서도 확인된 바 있듯이, 일반적으로 CRF를 사용하는 경우 품사 태깅 성능이 더 향상되는 것을 확인할 수 있었으며, 특히 기호류를 정규화 하지 않고 그대로 사용하는 편이 정규화 시킨 경우에 비해 더 좋은 성능을 기록하였다. 이는 형태소 분할 뿐만 아니라 품사 태깅 과정에서 특별한 기호들의 정보가 주변 음절의 태깅 성능 향상에 도움을 주는 것으로 보여진다.

표 1. 자모 조합 임베딩 사용 시 정확률 (256차원)

학습 방식	형태소 분할	품사 태깅
기호류 정규화	98.7%	96.9%
CRF + 기호류 정규화	98.7%	97.2%
기호류 비정규화	99.2%	97.1%
CRF + 기호류 비정규화	**99.2%**	**97.4%**

4.4 음절 임베딩 기반 태깅 성능

표 2는 4.1절에서 기술한 바와 같이 뉴스 50만 문서상에서 word2vec으로 pre-training 시킨 음절 임베딩 값을 이용해서 실험한 결과이다. 4.3절의 자모 조합 임베딩 결과와 비슷하게 CRF를 이용하는 경우 형태소 분할에서는 큰 차이가 없었으나 품사 태깅 시에는 음절 임베딩 방식이 약간 더 높은 정확률을 기록하였다. 그림 2의 결과에서 볼 수 있듯이, 일반적인 문서 분석의 경우에는 word2vec 음절 임베딩 성능이 자모 조합 임베딩에 비해 전반적으로 비슷하거나 또는 0.1~0.2% 정도 높은 성능을 보여주는 것을 확인할 수 있다.

Word2vec 음절 임베딩 방식과의 성능 비교를 위해 별도의 학습 과정 없이 랜덤으로 초기화하여 생성한 음절 임베딩 데이터를 이용한 실험도 진행하였다. 표 3는 랜덤으로 값을 생성한 음절 임베딩 데이터를 사용했을 때의 성능이다. 표 2,3에서 보이듯이, 음절 임베딩 방식은 대규모 말뭉치 상에서 학습시켜 사용하거나 또는 랜덤으로 초기화시켜 사용하거나 어느 방식이건 관계없이 둘 간에는 성능 상의 차이가 거의 발생하지 않음을 확인할 수 있었다.

표 2. Word2vec 음절 임베딩 사용 시 정확률 (256차원)

학습 방식	형태소 분할	품사 태깅
기호류 정규화	98.7%	97.1%
CRF + 기호류 정규화	98.8%	97.3%
기호류 비정규화	99.3%	97.3%
CRF + 기호류 비정규화	**99.3%**	**97.5%**

표 3. 랜덤 음절 임베딩 사용 시 정확률 (256차원)

학습 방식	형태소 분할	품사 태깅
기호류 정규화	98.7%	97.1%
CRF + 기호류 정규화	98.6%	97.2%
기호류 비정규화	99.3%	97.3%
CRF + 기호류 비정규화	**99.3%**	**97.5%**

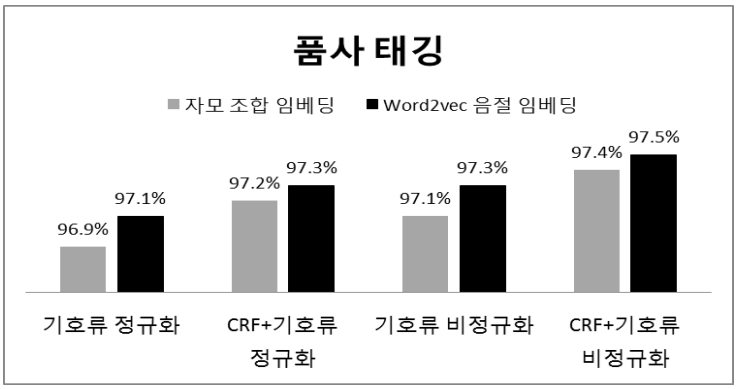


그림 2. 학습 방식별 품사 태깅 결과 비교

4.5 자모, 음절 임베딩 차원에 따른 성능 비교

입력으로 사용되는 임베딩 벡터의 차원을 줄이더라도 동일한 결과를 얻을 수 있다면 메모리와 속도 측면에서 이득을 얻을 수 있다. 따라서 최적의 임베딩 크기를 찾기 위해 자모 및 음절 임베딩 벡터를 32, 64, 128, 256 차원으로 변화시키면서 성능을 비교해 보았다.

표 4는 각 임베딩 차원에 따른 품사 태깅 정확률 추이를 보여주고 있다. 실험 결과를 보면 어떤 임베딩 방식을 사용하더라도 전체적으로 임베딩 크기가 증가할수록 태깅 성능이 비례하여 향상되는 경향을 확인할 수 있다.

또한 McNemar's test를 이용하여 word2vec 128차원 결과와 256차원 결과를 비교한 결과 p-value 값으로 0.015가 나왔고, 통계적으로 유의미하다는 결과를 얻을 수 있었다. 본 논문에서는 임베딩 벡터의 크기를 256차원으로 결정하여 사용하였다.

표 4. 임베딩 차원에 따른 품사 태깅 정확률 변화 (CRF+기호류 비정규화)

임베딩 차원	자모 조합 임베딩	Word2vec 음절 임베딩	랜덤 음절 임베딩
32	97.0%	96.0%	96.6%
64	97.3%	97.1%	97.0%
128	97.3%	97.4%	97.3%
256	**97.4%**	**97.5%**	**97.5%**

4.6 오타가 빈번한 환경에서의 성능 비교

다수의 오타가 발생하는 경우에서의 성능 평가를 위해 평가 데이터에서 각 어절 당 1개씩의 자모를 랜덤으로 변경시켰다. 그 후 평가 데이터 상에서 4.3, 4.4절과 동일한 조건으로 성능 평가를 진행하였다. 예를 들어 "중요한 일을 앞두고"이라는 문장이 있다면, 랜덤으로 한 어절 당 1 개씩 자모 변경이 적용되어 "중**쵸**한 일**일** 앞두**교**"과 같은 문장이 생성된다. 단, 자모가 변경되었더라도 품사는 변경 없이 원래 값을 그대로 유지하였다.

표 5,6은 강제로 오타를 생성한 문장에서의 형태소 분할 및 품사 태깅 성능을 보여주고 있다. 일반 문장 분석 시에는 자모 조합 임베딩, word2vec 음절 임베딩, 랜덤 음절 임베딩의 성능이 거의 비슷했던 것에 비해 (표 1,2,3), 강제 오타를 발생시킨 경우에는 자모 조합 임베딩의 성능이 word2vec 음절 임베딩 방식보다 모든 조건에서 우월하게 나타나는 것을 확인할 수 있다 (표 5,6). 자모 조합 임베딩이 형태소 분할만 하는 경우에는 0.9%, 품사 태깅 시에는 약 3.5% 이상 높은 정확률을 기록하였다.

음절 임베딩을 사용하는 경우에는 자모 한 개만 달라져도 완전히 다른 단어로 간주되기 때문에, 오타가 존재하는 경우 오타를 낸 형태소뿐만 아니라 주변 형태소까지 제대로 분석하지 못하는 경우가 발생한다. 하지만 자모 조합 임베딩 방식에서는 자모 한 개의 변형 정도로는 기존 단어와 거의 유사한 임베딩 값을 유지하게 된다. 따라서 이와 같은 특성 덕분에 표 7에서 확인 할 수 있듯이 자모 조합 임베딩은 word2vec 음절 임베딩보다 오타에 강건한 분석 결과를 생성해 낼 수 있다.

표 5. 강제 오타 생성한 경우 형태소 분할 정확률 비교

학습 방식	자모 조합 임베딩	Word2vec 음절 임베딩
기호류 정규화	96.3%	95.9%
CRF + 기호류 정규화	96.3%	**96.0%**
기호류 비정규화	96.9%	95.8%
CRF + 기호류 비정규화	**96.9%**	95.9%

표 6. 강제 오타 생성한 경우 품사 태깅 정확률 비교

학습 방식	자모 조합 임베딩	Word2vec 음절 임베딩
기호류 정규화	88.2%	**84.9%**
CRF + 기호류 정규화	88.2%	84.7%
기호류 비정규화	88.4%	84.7%
CRF + 기호류 비정규화	**88.4%**	84.8%

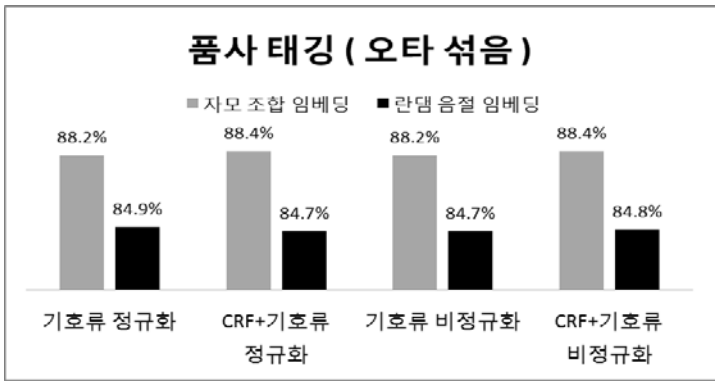


그림 3. 학습 방식별 오타 데이터 품사 태깅 결과 비교

[정답]
중요/NNG+한/XSV 일/NNG+을/JKO 앞두고/VV

-> 강제 오타 생성 (어절당 한 개)
중**쵸**/NNG+한/XSV 일/NNG+**일**/JKO 앞두**교**/VV

[자모 조합 임베딩 태깅 결과]
중쵸/NNG+한/XSV (O) 일/NNG+**일/VCP** (X) **앞두교/VV** (O)

[랜덤 음절 임베딩 태깅 결과]
중쵸한/VA (X) 일/NNG+**일/VCP** (X) **앞두교/NNG** (X)

그림 4. 임베딩 방식 별 품사 태깅 결과 비교

4.7 Layer Normalization 적용 결과

그림 5, 6는 학습 시 LSTM에 layer normalization을 적용한 경우와 그렇지 않은 경우의 loss 및 정확도 추이를 보여주고 있다. Layer normalization을 사용하면 훨씬 더 빠르게 loss값이 수렴하는 것을 확인할 수 있으며, 약 절반 정도의 epoch만으로도 동일한 정확률에 도달함으로써 학습 시간을 단축 할 수 있었다.

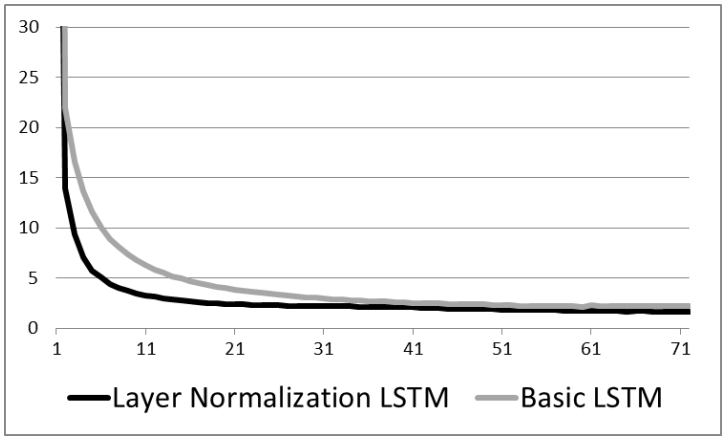


그림 5. LN_LSTM과 LSTM의 Loss 값 변화

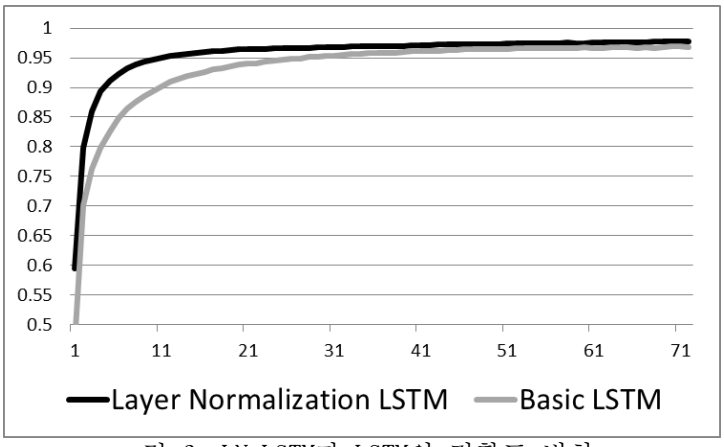


그림 6. LN_LSTM과 LSTM의 정확도 변화

5. 결론

본 논문은 기존의 딥 러닝 기반 품사 태깅 방법론들에서 사용하던 pre-training 시킨 음절 임베딩 값이나 랜덤 초기화된 음절 임베딩 값을 이용하지 않고 한글 자모 임베딩 값을 조합시킨 음절 임베딩 값의 사용을 제안하였다. 음절 임베딩 방식은 오타가 있거나 또는 기존에 학습되지 못한 미등록어가 들어오는 경우 이로 인한 영향을 많이 받기 때문에 품사 태깅 성능이 크게 하락하는 문제가 있었다. 그러나 자모 조합 임베딩을 사용하면 오타 및 변이형에 강건한 품사 태거 구축이 가능함을 실험으로 보였다. 자모 조합 임베딩의 이러한 특성은 신조어가 끊임없이 증가하고 오타가 빈번하게 발생하는 모바일 환경에서의 품사 태깅 시 안정적인 태깅 성능을 유지하는데 큰 도움이 되리라 생각된다. 또한 layer normalization을 적용하면 훨씬 빠른 학습 모델이 수렴이 가능하여 학습 시간 단축에 매우 효과적임을 확인할 수 있었다.

향후에는 보다 실제 조건에 근접한 오타 발생 실험을 위해, 랜덤 자모 변경이 아닌 실제 물리적인 키보드 또는 모바일 자판 상의 인접 키 위주로 오타 생성 실험을 진행해보면 더 의미 있는 결과를 얻을 수 있을 것으로 예상된다.

참고문헌

[1] 심광섭, “형태소 분석기 사용을 배제한 음절 단위의 한국어 품사 태깅”, 인지과학 2011.

[2] 나승훈, 양성일, 김창현, 권오욱 “CRF에 기반한 한국어 형태소 분할 및 품사 태깅”, 한글 및 한국어 정보처리 2012.

[3] 나승훈, 정상근. “딥 러닝에 기반한 한국어 품사 태깅”, 동계학술발표회 2014.

[4] 정의석, 박전규. “seq2seq 주의집중 모델을 이용한 형태소 분석 및 품사 태깅”, 한글 및 한국어 정보처리 2016.

[5] 이건일, 이의현, 이종혁, “Sequence-to-sequence 기반 한국어 형태소 분석 및 품사 태깅”, 한국정보과학회 2016.

[6] 김혜민, 윤정민, 안재현, 배경만, 고영중. “품사 분포와 Bidirectional LSTM CRFs를 이용한 음절 단위 형태소 분석기”, 한글 및 한국어 정보처리 2016

[7] 이충희, 임준호, 임수종, 김현기, “기분석사전과 기계학습 방법을 결합한 음절 단위 한국어 품사 태깅”, 한국정보과학회 2016

[8] 이창기, “Long Short-Term Memory 기반의 Recurrent Neural Network를 이용한 개체명 인식”, 한국컴퓨터종합학술대회 2015

[9] Jeffrey Dean, Greg Corrado, Rajat Monga, Kai Chen, Matthieu Devin, Mark Mao, Andrew Senior, Paul Tucker, Ke Yang, Quoc V Le, et al. “Large scale distributed deep networks.” NIPS 2012.

[10] Sergey Ioffe, Christian Szegedy. “Batch normalization: Accelerating deep network training by reducing internal covariate shift.” ICML 2015.

[11] Jimmy Lei Ba, Jamie Ryan Kiros, Geoffrey E. Hinton. “Layer Normalization.” NIPS 2016.

Word2Vec 모델을 활용한 한국어 문장 생성

남현규○, 이영석

충남대학교 컴퓨터공학과
hkanm@cnu.ac.kr, lee@cnu.ac.kr

Generating Korean Sentences Using Word2Vec

Hyun-Gyu Nam○, Young-Seok Lee
Chungnam National University, Dept. of Computer Engineering

요 약

고도화된 머신러닝과 딥러닝 기술은 영상처리, 자연어처리 등의 분야에서 많은 문제를 해결하고 있다. 특히 사용자가 입력한 문장을 분석하고 그에 따른 문장을 생성하는 자연어처리 기술은 기계 번역, 자동 요약, 자동 오류 수정 등에 널리 이용되고 있다. 딥러닝 기반의 자연어처리 기술은 학습을 위해 여러 계층의 신경망을 구성하여 단어 간 의존 관계와 문장 구조를 학습한다. 그러나 학습 과정에서의 계산양이 방대하여 모델을 구성하는데 시간과 비용이 많이 필요하다. 그러나 Word2Vec 모델은 신경망과 유사하게 학습하면서도 선형 구조를 가지고 있어 딥러닝 기반 자연어처리 기술에 비해 적은 시간 복잡도로 고차원의 단어 벡터를 계산할 수 있다. 따라서 본 논문에서는 Word2Vec 모델을 활용하여 한국어 문장을 생성하는 방법을 제시하였다. 본 논문에서는 지정된 문장 템플릿에 유사도가 높은 각 단어들을 적용하여 문장을 구성하는 Word2Vec 모델을 설계하였고, 서로 다른 학습 데이터로부터 생성된 문장을 평가하고 제안한 모델의 활용 방안을 제시하였다.

주제어: 문장 생성, 형태소 분석, word2vec

1. 서론

자연어처리는 인간의 언어 현상을 기계적으로 분석하여 컴퓨터가 이해할 수 있는 형태로 변환하고, 다시 인간이 이해할 수 있는 형태로 표현하는 기술을 의미한다. 최근 많은 기업들이 고도화된 머신러닝, 딥러닝 기술을 자연어처리 분야에 적용하여 서비스를 제공하고 있다. 구글 어시스턴스, 애플의 시리, 아마존 알렉사 등은 음성 인식 기술과 자연어처리 기술을 결합하여 사람의 발화를 문장화 하고 자연어처리를 통해 의도를 파악한다. 또한 채팅으로 사람과 대화할 수 있는 챗봇(Chatbot) 형태의 서비스 역시 자연어처리 기술을 적용하여 사람이 입력한 문장의 의미를 파악하고 사람이 이해할 수 있는 문장으로 결과물을 표현한다.

딥러닝 기반의 언어 모델은 여러 개의 문장과 단어, 말뭉치(Corpus) 데이터를 학습하여 단어와 문장 간의 의존관계를 분석한다. 문장의 일부가 주어졌을 때 나머지 부분을 추론하여 가장 높은 확률을 가진 단어들로 문장을 완성하는 확률 기반의 모델이며, 대표적으로 RNN과 LSTM 등이 있다. 딥러닝 기반 언어 모델은 학습 데이터로부터 문장의 특징을 자동적으로 학습하고 텍스트 뿐만 아니라 사진, 음성 등을 같이 활용할 수 있다. 그러나 실제 출력 계층의 차원이 크기 때문에 은닉 계층의 병렬 배치와 같이 연산 속도를 줄이는 작업이 추가로 필요하며, 작업을 뒷받침할만한 하드웨어 성능이 필요하다.

* 이 논문은 2016년도 정부(교육부)의 재원으로 한국연구재단의 지원을 받아 수행된 기초연구사업임 (NRF-2016R1D1A1A09916326)

그러나 텍스트를 처리하기 위해 개발된 Word2Vec 모델은 신경망 구조를 유지하면서 선형 구조를 가지므로 이전 모델에 비해 적은 시간 복잡도로 단어 간 유사도를 계산할 수 있다. 단어를 벡터로 표현하여 학습 과정에서 한 단어를 기준으로 단어 주변의 문맥을 얼마나 정확하게 예측하는지 계산하고, 계산된 결과를 바탕으로 단어 간 관계를 파악할 수 있다.

따라서 본 논문에서는 Word2Vec 모델을 기반으로 하여 단어 벡터를 학습하고 한국어 문장을 생성하는 방법을 제안한다. 학습 대상이 되는 한국어 문장을 형태소 분석을 통해 품사별로 단어 벡터를 생성한다. 생성한 단어 벡터는 Word2Vec 모델에 적용하여 유사도가 높은 단어 벡터를 사전에 지정한 문장 템플릿의 주어, 목적어 등 문장의 각 구성 요소로 하여 한국어 문장을 생성하였다. 학습 데이터에 따라 생성된 문장이 어떻게 달라지는지 분석하기 위해 뉴스 기사와 온라인 커뮤니티 게시물을 수집하여 각각 학습 모델을 만들고, 동일한 주어로 시작하는 문장을 생성하여 서로 비교하였다.

2. 관련 연구

Word2Vec 모델은 단어 학습의 계산 복잡도를 최소화하기 위해 고안되었다[1]. Feed-Forward Neural Net Language Model(NNLM)[2]의 한계를 극복하기 위해 고안되었다. NNLM 모델은 단어의 특징 벡터를 학습하여 단어의 분포를 구하는 과정에서 투영 계층과 출력 계층 간 연산이 오래 걸리는 문제점이 있다. NNLM의 한계를 해결

하기 위해 고안된 Recurrent Neural Network Language Model(RNNLM)은 투영 계층 없이 입력-은닉-출력 계층으로 신경망을 구성하고, 은닉 계층에서는 모든 단어에 대한 확률 분포를 계산하여 시간 흐름에 따라 재귀적으로 반복 학습한다. RNNLM은 NNLM에 비해 상대적으로 시간 복잡도가 여전히 높다는 문제점이 있다. 그러나 Word2Vec 모델은 여러 개의 주변 단어를 통해 대상 단어를 유추하는 Continuous Bag-of-Word(CBOW), 주어진 단어 하나를 가지고 주위에 등장하는 나머지 단어들을 유추하는 Skip-gram 으로 구성되어 가공되지 않은 텍스트로부터 효율적으로 예측할 수 있다. 또한 해당 단어가 나올 확률을 예측하는 트리를 구성할 때 Binary Huffman Tree를 사용한다. 즉, 자주 등장하는 단어들이 보다 짧은 path로 도달하게 되어 전체적인 계산 복잡도가 낮아지는 효과가 있다.

[3]에서는 여러 문장을 만들어 둔 상태에서 상황과 맥락에 따라 문장을 끌어다 사용하는 방식으로 프로야구 경기를 요약한 기사를 작성하였다. 문장의 주어와 서술어 등을 비워 둔 상태에서 프로야구 경기 데이터를 결합하여 뉴스 기사의 각 문장을 생성하고, 기록 시점 이전에 경기 데이터와 현재 이벤트 이후 경기 데이터를 비교하여 '안타깝게도' 등과 같이 무드를 더하여 문장을 생성하였다. 그러나 이 연구는 문장 템플릿에 사용하는 목적어와 서술어가 정형화 되어 있고, 새로운 문장을 생성하기 위해서는 템플릿을 추가하여 경기 데이터를 적용해야 한다는 불편함이 있다. 본 연구에서 사용하는 Word2Vec 모델은 템플릿을 적용한 기본 문장을 구성한 후 워드 벡터에서 가장 유사한 품사들을 추가하여 문장을 확장할 수 있으므로 이벤트마다 별도의 문장 템플릿이 필요하지 않다.

3. 한국어 문장 생성 모델

3.1 형태소 분석

Word2Vec 모델에서 단어 벡터는 문장을 구성하는 각각의 품사가 된다. 따라서 Word2Vec 모델을 구성하기 위해서는 학습 데이터를 형태소 분석을 통해 문장을 어근, 접두사/접미사, 품사 등으로 가장 세분화하여 단어 벡터로 생성하는 전처리 과정이 필요하다. 본 연구에서는 파이썬 한글 형태소 분석 라이브러리인 KoNLPy[4]의 트위터 형태소 분석기를 이용하여 단어 벡터를 구성하였다. 그림 1은 학습할 한국어 문장을 형태소 분석한 예시이다. 분석 과정에서 품사(POS) 태그를 함께 태깅 하여 동음이의어를 구분하였다. 또한 학습 데이터 중 복합 명사의 경우 형태소 분석 과정에서 명사와 명사로 분리되지 않도록 미리 사전을 구성하여 하나의 명사로 분류하였다.

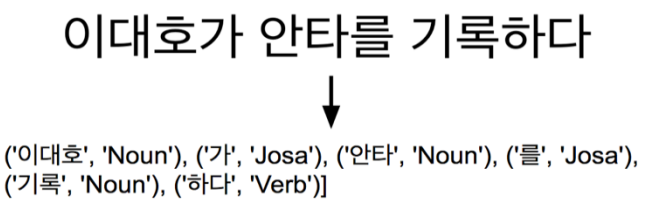


그림 1 형태소 분석 예시

3.2 Word2Vec 모델 구성

형태소 분석을 통해 품사 태그가 포함된 단어들로 벡터를 구성하여 Word2Vec 모델을 생성하였다. Word2Vec 모델은 학습 과정에서 기준이 되는 단어로부터 주변의 단어 문맥을 파악하여 현재의 단어 벡터 위치가 얼마나 정확한지를 계산한다. 예를 들어, 차원의 크기가 100인 벡터 공간에서 각 워드 벡터는 100차원 공간의 점 하나에 해당되며, 학습 과정을 통해 의미가 유사한 단어들이 근처에 위치하게 된다. 단어 간 유사도가 높다는 것은 벡터 공간에서 단어 간 거리가 가깝다는 것을 의미한다.

학습이 완료되면 미리 지정한 템플릿에 Word2Vec의 단어 벡터를 이용하여 문장을 생성한다. 주어와 유사도가 가장 높은 단어 벡터들이 목적어, 서술어 등 나머지 문장의 구성 요소가 되어 전체 문장을 완성한다. 그림 2는 단어 벡터들을 t-SNE 기법으로 시각화한 예시이다.

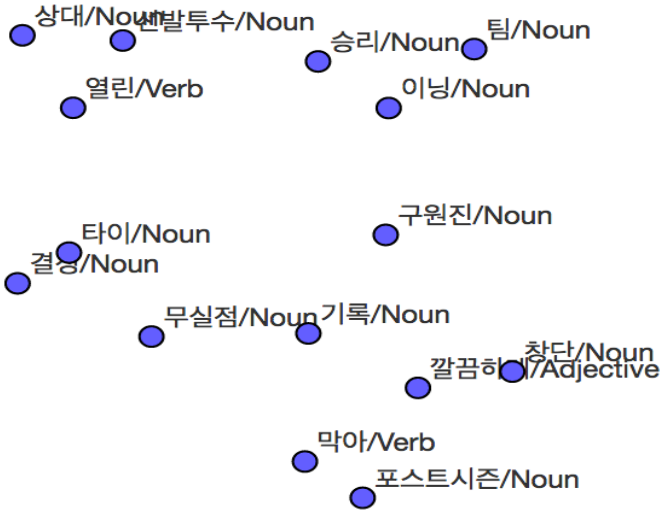


그림 2 Word2Vec 단어 임베딩 예시

벡터 공간에 단어가 임베딩이 완료되면 단어 벡터 간 유사도를 계산하여 결과값이 가장 높은 단어 벡터를 이용하여 문장을 생성한다. 먼저 생성할 문장의 주체가 되는 명사를 기준으로 가장 유사도가 높은 단어 벡터 중 조사와 명사를 찾는다. 조사는 명사와 함께 결합하여 '~은', '~가'와 같이 문장의 주어를 구성하고, 주어와 가장 유사도가 높은 명사는 목적어가 되어 주어가 하는 행위의 대상이 된다. 목적어 역시 주어와 마찬가지로 유사

도가 가장 높은 조사와 동사를 찾아 문장의 나머지 구성 요소인 목적어와 서술어로 사용한다. 그림 3은 Word2Vec 모델을 이용하여 문장을 생성하는 전체 과정을 플로우 차트로 나타내었다.

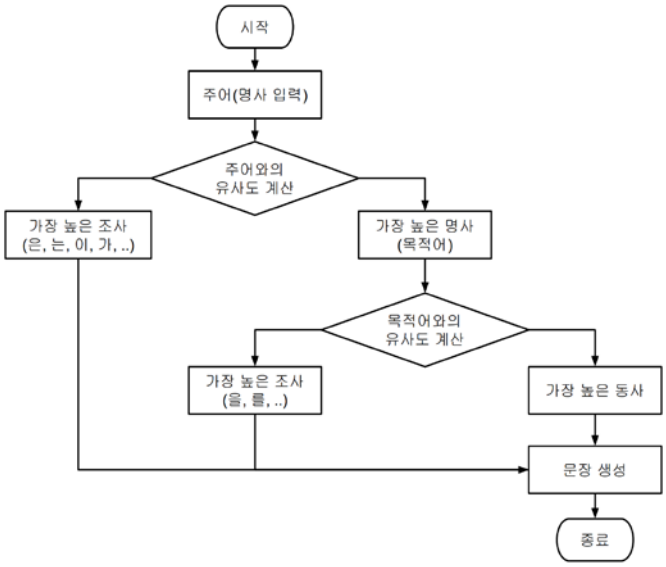


그림 3 유사도 기반 문장 생성 플로우차트

4. 한국어 문장 생성 예시

4.1 학습 데이터

본 논문에서 제안한 Word2Vec 모델의 문장 생성 성능을 평가하기 위해 동일한 주제의 두 가지 학습 데이터를 이용하였다. 학습 데이터로는 네이버 스포츠의 야구 뉴스 기사 1,575개의 본문과 대표 온라인 커뮤니티 디시인사이드의 국내야구 갤러리 게시물 38,724개를 수집하였다. 뉴스 기사와 커뮤니티 게시물 모두 같은 기간 동안 생성된 일주일 분 텍스트 데이터를 수집하였으며, 커뮤니티 게시물의 경우 뉴스와는 달리 본문보다는 제목에 의견을 표현하므로 학습할 문장의 개수를 조절하여 뉴스에 비해 상대적으로 많은 양의 게시물을 수집하였다.

4.2 실험 결과

생성한 문장의 템플릿은 지정한 주어와 유사도가 가장 높은 명사를 목적어로 하고, 목적어와 가장 유사도가 높은 동사를 서술어로 하여 (주어+목적어+서술어) 형태로 구성하였다. 주어와 목적어와 함께 사용되는 조사 역시 각각의 단어 벡터와 가장 유사한 조사를 사용하였다. 두 학습 모델을 서로 비교하기 위해 동일한 주어로 유사도를 계산하여 문장을 생성하였다. 표 1은 두 가지 데이터로 학습한 Word2Vec 모델이 생성된 문장의 예시이다.

표 1 문장 생성 결과

NO	뉴스 기사	커뮤니티 게시물
1	**롯데**가 승리를 하다	**롯데**가 극장을 하다
2	**최진행**이 삼진을 당하다	**최진행**이 진짜 이다
3	**한화**가 감독을 바꾸다	**한화**가 감독은 나가다
4	**이대호**는 도루를 하다	**이대호**는 돼지 이다
5	**한화**가 실책을 하다	**한화**는 수비가 망했다

학습 데이터에 따라 생성된 문장의 주어-서술어 간 관계가 어색할 수 있다. 문장 3의 뉴스 학습 데이터는 주어-서술어 구조를 유지한 상태에서 '바꾸다'라는 서술어로 일관성 있게 작성된 반면, 커뮤니티 게시물 학습 데이터의 대다수는 주어-서술어 구조가 아닌 '나가라/나가'와 같이 명령형으로만 작성되었다. 따라서 커뮤니티 게시물 기반 문장은 뉴스 기사에 비해 명사와 동사 간 유사도가 낮아 상대적으로 문맥상 어색한 부분이 있다. 또한 같은 의미를 가진 문장이지만 학습 데이터에 따라 서로 다른 표현으로 생성될 수 있다. 문장 1과 2는 커뮤니티에서 주로 사용하는 표현인 극적인 승리를 '극장'으로, '실책을 하다'라는 문장은 '진짜다'로 생성되었다. 문장 4와 5에서는 뉴스에서는 찾을 수 없는 개인의 주관적인 표현을 확인할 수 있다.

5. 결론

본 논문에서는 Word2Vec 모델을 이용하여 한국어 문장을 생성하는 방법에 대해 제안하였다. Word2Vec 모델은 충분한 양의 데이터로 학습할 경우 높은 정확도로 단어 간 유사도를 계산할 수 있었으며, 다른 텍스트 모델에 비해 상대적으로 계산 효율이 높음을 확인할 수 있었다. 그러나 단어 벡터를 구성하는 과정에서 형태소 분석이 제대로 이루어지지 않을 경우 전달하고자 하는 의미가 달라지거나 문맥상 어색한 문장이 생성될 수 있으며, 신경망 모델처럼 단어 간 의존관계를 파악하여 문장을 자동으로 생성하기에 어려움이 있다. 또한 학습 데이터에 따라 생성하는 문장의 표현이 서로 달라진다. 예를 들어, '실책' 이라는 이벤트에서 뉴스 기사 기반의 모델에서는 '실책을 하다'로 생성하였지만, 커뮤니티 게시물 기반 모델에서는 '수비가 망했다'로 표현하였다. 이것은 같이 사용되는 단어들이 유사도가 높게 측정되기 때문에 같은 주어를 사용하더라도 서로 다른 내용의 문장을 함께 학습할 경우 유사도가 높더라도 함께 사용된 품사가 무엇인가에 따라 달라질 수 있다.

그러나 유사도를 기반으로 하여 문장을 생성하여 주어와 목적어, 목적어와 서술어 간의 관계가 잘 표현되어 뉴스 기사와 같이 객관적인 사실을 전달하는 문장을 생성하거나 문서의 내용을 요약하는 문장을 생성할 경우에는 활용 가능성이 있음을 확인할 수 있었다.

참고문헌

[1] T. Mikolov, K. Chen, G. Corrado, and J. Dean.

Efficient estimation of word representations in vector space. arXiv preprint arXiv:1301.3781, 2013.
[2] C. Chelba, M. Norouzi, and S. Bengio. N-gram language modeling using recurrent neural network estimation. arXiv preprint arXiv:1703.10724, 2017.
[3] 김동환, 이준환. (2015). 로봇 저널리즘 : 알고리즘을 통한 스포츠 기사 자동 생성에 관한 연구. 한국언론학보, 59(5), 64-95.
[4] 박은정, 조성준, “KoNLPy: 쉽고 간결한 한국어 정보처리 파이썬 패키지”, 제 26회 한글 및 한국어 정보처리 학술대회 논문집, 2014.

한국어 생략어복원 가이드라인

류지희†, 임준호, 임수종, 김현기

한국전자통신연구원 언어지능연구그룹
{chrisjihee, joonho.lim, isj, hkk}@etri.re.kr

Korean Zero Anaphora Resolution Guidelines

Jihee Ryu†, Joon-Ho Lim, Soojong Lim, Hyunki Kim
Electronics and Telecommunications Research Institute

요 약

말과 글에서 유추가 가능한 정보에 대해서는 사람들이 일반적으로 생략해서 표현하는 경우를 볼 수 있다. 사람들은 생략된 정보를 문맥적으로 유추하여 이해하는 것이 어렵지 않지만, 컴퓨터의 경우 생략된 정보를 고려하지 못해 주어진 정보를 완전하게 이해하지 못하는 문제를 낳게 된다. 우리는 이러한 문제를 생략어복원을 통해 해결할 수 있다고 여기면서 본 논문을 통해 한국어 생략어복원에 대해 정의하고 기술 개발에 필요한 말뭉치 구축 시의 생략어복원 대상 및 태깅 사례를 포함하는 가이드라인을 제안한다. 또한 본 가이드라인에 의한 말뭉치 구축 및 기술 개발을 통해서 엑소브레인과 같은 한국어 질의응답 시스템의 품질 향상에 기여하는 것이 본 연구의 궁극적인 목적이다.

주제어: 자연어처리, 한국어 생략어복원, 생략어복원 태깅 가이드라인, 엑소브레인

1. 서론

우리의 일상 언어 사용에서 경제성의 원리가 작용되어 청자가 알고 있는 것이나 충분히 유추가 가능한 정보는 축약하거나 생략하여 표현하는 경우가 있다. 축약되었거나 생략된 표현은 대용어(anaphora: 조응어 또는 조응대용어)로 나타날 수 있고, 컴퓨터가 이것을 명확하기 인식하기 위하여 대용어 해결(anaphora resolution)이라는 자연어처리 문제로 정의하여 다루고 있다[1]. 생략어복원(zero anaphora resolution)은 어떠한 동사 표현 어구나 명사 표현 어구에서 일부 문장 성분이 미리 나타나 유추가 가능하거나 암묵적으로 알고 있기에 문장 내에서 생략된 해당 성분을 찾아 복원해주는 문제이다. 본 논문에서는 생략된 문장 성분을 생략어(zero anaphora: 생략된 대용어 또는 무형대용어)라 하고, 생략된 문장 성분이 종속되는 대상을 지배소(head)라 하고, 생략어가 복원되어야할 원래 표현을 선행어(antecedent)라고 한다.

생략어복원은 상호참조해결와 달리 선행어를 대신하여 사용된 대용어가 대명사나 약어 등의 형태로 나타나는 것이 아니라 아예 생략되었다는 것이 차이점이라고 할 수 있다. 대용어가 생략되어 있기 때문에 주어진 문장을 읽다가 특정 동사 표현 어구나 명사 표현 어구 내에서 생략어가 존재함을 먼저 알아내야 한다. 그 뒤, 해당 생략어에 대한 선행어를 결정하는 과정에서 문서 내에 나타난 표현 이외에도 암묵적이기에 문서 내에 존재하지 않는 표현까지 고려해야하는 특수성이 있다.

이러한 생략어를 복원시킨 결과는 텍스트 상에서 이전에 언급되었거나 암묵적으로 표현된 정보를 찾아주어 텍스트의 의미를 보다 명확하게 이해하게 해주고, 담화나 문서 내에서 언급하는 대상에 대한 정보를 일관성 있게 유지하게 해준다. 따라서 생략어복원 문제의 해결은 문서에서 등장하는 개체와 그에 대한 정보를 이해하는데 상당히 중요한 역할을 하며 정보 검색, 정보 추출, 질의응답, 문서 요약, 기계 번역 등에서 유용하게 사용될 수 있다.

생략어복원을 이해하기 위해서 [2]에서 들었던 예시를 통해 명시적인 대용어와 암묵적인 대용어가 나타나는 경우를 살펴볼 수 있다.

(ㄱ) **철수는**† 학교에 갔다. 가는 도중 **그는**¶ 영희를 만났다.
(ㄴ) **철수는**† 학교에 갔다. 가는 도중 영희를 만났다.

(ㄱ)에서 대용어 “그는”의 선행어는 이전 문장의 “철수는”이다. 대명사로 표현된 대용어 “그는”의 선행어를 찾아내는 대용어 해결 문제는 상호참조해결로 해결이 가능하다. 반면, (ㄴ)에서는 대용어가 생략되어 있고, 이것은 생략어복원으로 해결이 가능하게 된다. 두번째 문장에서 “만나다”는 동사에 대한 주어가 생략되어 있어 생략어가 있음을 먼저 인지한 뒤, 해당 생략어에 대한 선행어는 앞에서 등장했던 “철수는”임을 판단할 수 있어야 한다. 즉, “가는 도중 철수는 영희를 만났다.”로 생략어를 복원시키는 것이 정보를 보다 명확하고 구체적으로 드러낼 수 있다는 사실을 인지해야 한다. 이러한 복원 결과로부터 지식을 생성할 때, (ㄷ)과 같이 표현할 수 있는 관계 정보를 알 수 있게 된다.

(ㄷ) [**철수**† - 만나다 - 영희]

생략어복원 문제 중에서도 위키피디아와 같은 백과사전 본문에서는 표제어를 암묵적으로 알고 있다고 판단하여 문장 내에서 대부분 표현하지 않는 측면이 있다.

(ㄹ) **지미 카터는**† 조지아 주 섬터 카운티 플레인스 마을에서 태어났다. 조지아 공과대학교를 졸업하였다. 그 후 해군에 들어가 전함·원자력·잠수함의 승무원으로 일하였다.

(ㅁ) 지미 카터는† 조지아 주 섬터 카운티 플레인스 마을에서 태어났다. 지미 카터는¶ 조지아 공과대학교를 졸업하였다. 그 후 해군에 들어가 전함·원자력·잠수함의 승무원으로 지미 카터는¶ 일하였다.

(ㄹ)은 "지미 카터"에 대한 위키피디아 본문 설명의 일부분이다. 이 텍스트의 첫번째 문장에서 나타난 선행어 "지미 카터는"으로 복원시킬 수 있는 대용어들이 첫 번째 문장 이후에 모두 생략되어 있음을 알 수 있다. 생략어가 복원된 결과의 예로 (ㅁ)과 같은 결과를 들 수 있고, 복원되는 생략어의 위치는 한국어의 어순 특성상 자유로울 수 있음을 알 수 있다. 생략어복원 문제는 필요에 따라 백과사전 본문 내에서 백과사전의 표제어로 생략된 대용어를 복원시키는 표제어 복원 문제로 축소시킬 수 있다.[2-4]

한국어 뿐만 아니라, 중국어[5-9]와 일본어[10-14]에 대해서도 이러한 생략어복원 문제를 해결하기 위한 방법들이 각각 제안되어 왔다. 방법론 면에서도 규칙과 구문적 패턴을 활용하는 방법부터 전통적인 기계학습 방법에서 최근에는 딥러닝을 활용하는 방법까지 다양하게 시도되고 있다.

이러한 생략어복원을 해결하는 기술을 한국어 질의응답 시스템인 엑소브레인에 활용하기 위해서 본 논문은 생략어복원 대상과 태깅 사례에 대한 가이드라인을 제안하고자 한다. 본 논문의 구성은 다음과 같다. 2장에서는 생략어복원의 대상을 소개하고, 3장에서는 태깅 결과물 포맷과 함께 태깅 사례들을 소개한다. 그리고 4장에서는 말뭉치 구축 도구를 소개한다. 마지막 5장에서는 결론을 맺도록 한다.

2. 생략어복원 대상

생략된 모든 정보를 복원하는 데에는 한계가 있으므로, 본 논문에서는 각 생략어복원 주요 개념에 대한 후보를 다음과 같이 정한다.

- 지배소 후보
  - 동사 표현 어구
  - 주어 및 목적어 역할을 하는 명사 표현 어구
- 생략어 후보
  - 동사 표현 어구에서 생략된 주어
  - 동사 표현 어구에서 생략된 목적어 및 필수 부사어
  - 주어 및 목적어 표현 어구에서 생략된 관형어
- 선행어 후보
  - 해당 문서의 표제어
  - 필자가 염두에 두고 있는 포커스
  - 상호참조해결에 의해 탐지된 멘션 및 개체
  - 암묵적으로 또는 상식적으로 알고 있거나 아직 알려지지 않은 개념

지배소 후보에 대해서는 2.1절에서, 생략어 후보에 대해서는 2.2절에서, 선행어 후보에 대해서는 2.3절에서 각각 설명한다.

2.1. 지배소 후보

2.1.1. 동사 표현 어구

본 논문의 생략어복원 대상이 되는 생략어에 대한 지배소 후보로서 먼저 동사 표현 어구를 생각해볼 수 있다. 동사 표현 어구는 일반적으로 담화나 문서 내에서 개체와 개체 또는 개체와 값 간의 사건, 행동 및 상태와 같은 주요한 정보를 서술하는 형태이다. 이러한 동사 표현 어구에서 생략 표현된 정보가 있어 이를 일차적인 지배소 후보로 생각할 수 있다. 동사 표현 어구를 인식하기 위해 의존 구문분석 결과 및 의미역 부착 결과를 참고할 수 있으며, 구문 태그로 VP[용언]나 VNP[긍정지시사구]를 가지는 것이 이에 해당한다고 할 수 있다. 이에 대한 예시는 다음과 같다.

케냐 공화국 또는 케냐는 동아프리카의 공화국이다*[VNP]. 인도양에 면해*[VP] 있으며 북동쪽으로 소말리아, 북쪽으로 에티오피아와 남수단, 서쪽으로 우간다, 남쪽으로 탄자니아와 국경을 맞닿고*[VP] 있다.

2.1.2. 주어 및 목적어 역할을 하는 명사 표현 어구

다음으로 주어 및 목적어 역할을 하는 명사 표현 어구를 생각해볼 수 있다. 명사 표현 어구는 일반적으로 담화나 문서 내에서 어떠한 구체적 개체나 추상적 개념의 명칭을 나타내는 형태이다. 이러한 명사 표현 어구에서도 생략된 정보가 있어 이를 이차적인 지배소 후보로 생각할 수 있다. 그 중에서도 주어 및 목적어 역할을 하는 명사 표현 어구는 문장에서 주요한 성분이므로 생략 표현된 정보까지 이해하는 것이 중요할 수 있다. 주어 및 목적어 역할을 하는 명사 표현 어구를 인식하기 위해 구문 태그로 NP[체언]를 가지면서 기능 태그로 SBJ[주어]와 OBJ[목적어]를 가지는 것이 이에 해당한다고 할 수 있다. 이에 대한 예시는 다음과 같다.

케냐 공화국 또는 케냐는*[NP_SBJ] 동아프리카의 공화국이다. 인도양에 면해 있으며 북동쪽으로 소말리아, 북쪽으로 에티오피아와 남수단, 서쪽으로 우간다, 남쪽으로 탄자니아와 국경을*[NP_OBJ] 맞닿고 있다. 수도는*[NP_SBJ] 나이로비이며 공용어는*[NP_SBJ] 영어와 스와힐리어이다.

2.2. 생략어 후보

한가지 전제할 것은 생략어 태깅은 생략된 성분을 파악하는 것이 목표이지 자연스러운 완성된 문장을 만드는 것이 목표가 아니라는 것이다. 따라서 생략어가 어느 지배소에 속하는지, 선행어는 무엇인지까지만 태깅해주면 된다.

2.2.1. 동사 표현 어구에서 생략된 주어

생략어 후보 종류 중에 가장 많은 비율을 차지하는 것은 바로 생략된 주어이다. 일반적으로 동사는 최소 1개의 주어가 필수적으로 필요하지만 앞에서 등장한 개체를 이미 알고 있을 것이라는 가정 하에 생략하여 표현하는 경우가 많다. 이에 대한 예시는 다음과 같다.

케냐 공화국 또는 케냐는 동아프리카의 공화국이다. [?는]¶ 인도양에 면해*[VP] 있으며 북동쪽으로 소말리아, 북쪽으로 에티오피아와 남수단, 서쪽으로 우간다, 남쪽으로 탄자니아와 [?는]¶ 국경을 맞닿고*[VP] 있다.

2.2.2. 동사 표현 어구에서 생략된 목적어 및 필수 부사어

생략어 후보 종류 중에 다음으로 많은 비율을 차지하는 것은 바로 생략된 목적어 및 필수 부사어이다. 일반적으로 타동사는 필요에 따라 1개 또는 그 이상의 목적어 및 필수 부사어가 필요하지만 생략하여 표현하는 경우가 있다. 이에 대한 예시는 다음과 같다.

> 월스트리트 저널은 다우존스가 발행하는 조건으로서 세계 10대 신문 중 하나이며, 세계적으로 가장 영향력이 큰 경제지이다. 1889년 다우존스사의 다우가 기업과 금융 관계를 전문적으로 보도하고자 [?를]¶ 창간했다*[VP].

2.2.4. 주어 및 목적어 표현 어구에서 생략된 관형어

적은 비율이기는 하지만 정보 추출 관점에서 추가로 다룰 필요가 있는 것은 바로 생략된 관형어이다. 주어 및 목적어를 표현하는 데 있어서 앞에서 등장한 개체가 관형어가 됨에도 불구하고 이미 알고 있는 정보가 될 때는 이를 생략한 채 표현하는 경우가 있다. 이에 대한 예시는 다음과 같다.

> 케냐 공화국 또는 케냐는 동아프리카의 공화국이다. 인도양에 면해 있으며 북동쪽으로 소말리아, 북쪽으로 에티오피아와 남수단, 서쪽으로 우간다, 남쪽으로 탄자니아와 국경을 맞닿고 있다. [?의]¶ 수도는*[NP_SBJ] 나이로비이며 [?의]¶ 공용어는*[NP_SBJ] 영어와 스와힐리어이다.

2.3. 선행어 후보

본 절에서 설명하는 선행어 후보는 표제어이면서 멘션이 되는 복합적인 형태도 가능한데, 여기에서의 구분은 선행어를 어느 한가지 종류로 규정하려는 것이 아니고 선행어의 후보로 살펴야할 대상을 단계적으로 제공함으로써 태깅 작업자가 놓치지 않고 선행어를 찾도록 가이드하기 위함이다. 선행어 후보에 대한 전제사항은 상호참조해결에 의해 탐지된 멘션 및 개체에 대해서는 멘션의 중심어 중 생략어와 가까운 위치의 것을 선행어로 결정하는 것을 원칙으로 한다는 것과, 포커스에 대한 고려는 새롭게 고려되는 의미 자질이므로 포커스로 여길 수 있는 것을 선행어로 결정할 수는 있지만 포커스에 대한 태깅은 추후에 다시 논의하여 태깅 가이드를 구체화한 뒤 진행하도록 한다는 것이다. 본 가이드라인에서는 후보의 발생 개수가 적고 명시적인 것에서부터 범위가 넓거나 암묵적인 것으로 순서를 정하여 소개한다.

2.3.1. 해당 문서의 표제어

생략어에 대한 선행어 후보로서 먼저 해당 문서의 표제어를 생각해볼 수 있다. 해당 문서의 표제어는 텍스트를 이해하는 데 있어서 가장 먼저 접하는 정보이면서 해당 문서 전체를 한 마디로 설명할 수 있는 개념이기 때문에 필자는 독자가 이를 이미 알고 있다고 생각할 수 있다. 특히 백과사전 종류의 텍스트에서는 대부분의 문장이 표제어를 구체적으로 설명하기 위한 문장들이 많으며, 컴퓨터가 문서 내의 문장들을 분석할 때는 생략된 표제어를 복원시켜야 백과사전에서 설명하는 내용을 일관적으로 이해할 수 있게 된다. 이러한 선행어가 사용된 예는 다음과 같다.

> (표제어 : 케냐†)

> 케냐 공화국 또는 케냐는 동아프리카의 공화국이다. [케냐†는]¶ 인도양에 면해*[VP] 있으며 북동쪽으로 소말리아, 북쪽으로 에티오피아와 남수단, 서쪽으로 우간다, 남쪽으로 탄자니아와 [케냐†는]¶ 국경을 맞닿고*[VP] 있다.

2.3.2. 필자가 염두에 두고 있는 포커스

생략어에 대한 선행어 후보로서 다음으로 필자가 염두에 두고 있는 포커스를 생각해볼 수 있다. 일반적인 말과 글에는 각 단락 또는 문장 단위에서 필자가 생각하고 있는 주된 포커스(focus)가 명시적으로 또는 묵시적으로 존재하는 경우가 있다. 포커스는 단위 텍스트 내에서 존재하지 않거나, 새롭게 등장하거나, 그대로 유지되거나, 기존의 포커스 중 하나로 이동할 수 있다. 상호참조해결이나 생략어복원과 같은 대용어 해결 문제에서 현재 등장한 멘션에 대해 참조되는 선행 멘션이나 생략어에 대한 선행어는 이러한 포커스인 경우가 있다. 따라서 생략어복원 문제에서 선행어를 더 정확하게 선택하는데 있어서, 현재의 포커스가 무엇인지 고려하여 생략어에 대한 선행어를 결정할 필요가 있다. 특히 뉴스 분야의 텍스트에서는 사건의 전말을 설명하기 위하여 전체 사건 내에서 부분적으로 포커스를 차츰 옮겨가면서 전체 사건의 내용을 전반적으로 다루는 경우가 종종 있다. 이러한 선행어가 사용된 예는 다음과 같다.

> 문재인 대통령은 21일 '경제 사령탑'인 경제부총리 겸 기획재정부 장관 후보자에 김동연(60) 아주대 총장, 외교부 장관 후보자에 여성인 강경화(62) 유엔 사무총장 정책특보를 각각 내정했다. … 김동연 부총리 후보자†는 충북 음성 출신으로 '고졸신화의 인간승리 드라마'로 불린다. [김동연 부총리 후보자†는]¶ 덕수상고 졸업 뒤 은행에 취직해*[VP] 직장생활을 하며*[VP] 행정고시와 입법고시에 동시 합격한*[VP_MOD] 입지전적의 인물로 평가 받는다*[VP]. …

2.3.3. 상호참조해결에 의해 탐지된 멘션 및 개체

생략어에 대한 선행어 후보로서 일반적으로는 상호참조해결에 의해 탐지된 멘션 및 개체를 생각해볼 수 있다. 상호참조해결에 대한 보다 상세한 내용은 한국어 상호참조해결 가이드라인 및 관련 연구 결과[15]를 참고할 수 있다. 상호참조해결(coreference resolution)은 임의의 개체(entity)에 대하여 다른 표현으로 사용되는 단어들을 찾아 서로 같은 개체로 연결해주는 자연어처리 문제이다.[16] 멘션(mention)은 상호참조해결의 대상이 되는 모든 명사구를 의미한다. 멘션에서 해당 구의 실질적인 의미를 나타내는 단어를 중심어라하며, 멘션은 중심어를 중심으로 이를 수식하는 수식어까지도 포함한다. 개체(entity)는 동일한 멘션의 집합으로써 상호참조해결의 결과이다. 선행 멘션(antecedent)과 현재 등장한 멘션 간의 참조를 해결하면 하나의 개체로 포함된다. 이러한 선행어가 사용된 예는 다음과 같다.

> 비텐베르크 대학교의 요한 스타우피츠 교수†는 루터가 성서에 대해 진지하게 공부하면 평안을 찾을 것이라고 생각하였다. 그래서 [요한 스타우피츠 교수†는]¶ 그를 성서학 교수사제로 임명하였는데*[VP], 스타우피츠 교수의 결정은 루터가 신앙적인 고민을 해결하는데 도움이 되었다.

2.3.4. 암묵적으로 또는 상식적으로 알고 있거나 아직 알려지지 않은 개념

생략어에 대한 선행어 후보로서 마지막으로 암묵적으로 또는 상식적으로 알고 있거나 아직 알려지지 않은 개념을 생각해볼 수 있다. 필자는 이러한 개념들은 굳이 명시적으로 표현하지 않아도 될 것이라고 생각할 수 있다. 게다가 만약 아직 알려지지 않은 개념이라면 표현하기 어려울 수 있다. 이런 개념에 대해 사람은 내용을 이해하면서 자연스럽게 생각해낼 수 있어도 컴퓨터는 외부 지식의 도움을 받지 않고서는 알기 어려운 정보일 수 있다. 특히 이런 종류의 선행어는 해당 문서 어디에도 발견되지 않을 수 있어서 대상을 정확히 찾는데 더 어려움이 있다. 이러한 선행어가 사용된 예는 다음과 같다.

케냐 공화국 또는 케냐는 동아프리카의 공화국이다. … 동아프리카에서 [어느 발굴가†에 의해]¶ 발견된✱[VP_MOD] 화석에 따르면 조상이 2백만 년 전 이 지역에서 살았다고 한다.

3. 생략어복원 태깅 사례

생략된 정보 중에는 사람조차 구체적으로 알기 어려운 정보가 존재하기도 하며, 알더라도 정보로서의 가치가 적은 것도 존재한다. 본 생략어복원 태깅의 궁극적 목적은 주어진 텍스트에서 정보로서의 가치가 있는 생략된 정보를 복원시킴으로써, 사람이 보기에도 의미가 명확해지고 그것을 해석하는 컴퓨터가 중요한 정보들을 놓치지 않고 잘 파악하여 엑소브레인을 포함하여 앞서 제시한 응용 등에서 도움을 받기 위함이다.

실제 태깅 결과에 대한 이해를 위하여 본 장에서 다루는 예시들은 생략어복원 태깅 결과물 포맷으로 어떻게 표현되는지를 함께 살펴본다. 3.1절에서 태깅 결과물 포맷을 소개하고 이전 장에서 정의한 생략어복원 대상인 생략어와 선행어를 중심으로 태깅 사례들을 3.2절과 3.3절에서 소개한다. 포커스에 대한 태깅은 추후에 정한다.

3.1. 태깅 결과물 포맷

생략어복원 태깅의 결과물은 엑소브레인 언어분석 말뭉치[17-18]와 같은 Json 포맷을 따르고 있다. 이 중에서 생략어복원에 해당하는 부분은 다음과 같다.

```
"ZA" : [
        {
          "id" : integer, // 0부터 시작함
          "type" : { "s" | "o" | "a" },
          "head_wid" : integer, // 0부터 시작함
          "ant_text": "string",
          "ant_sid" : integer, // 0부터 시작함
          "ant_wid" : integer, // 0부터 시작함
          "ant_is_title" : { 0 | 1 } // 0=No, 1=Yes
        }, …
       ]
```

3.2. 생략어 태깅

생략어 태깅의 전형적인 순서는 먼저 문장 내에서 지배소 후보들을 먼저 찾은 뒤, 각 지배소 후보에서 생략어 후보가 포함되어 있는지를 찾아보는 것이다. 구체적인 생략어 태깅 사례들을 살펴보면 다음과 같다.

가) 동사 표현 어구에서 필수 성분 중 생략된 것이 있다면 해당 성분을 생략어로 태깅한다.

[?는]¶ 인도양에 면해✱[VP] 있으며 북동쪽으로 소말리아, 북쪽으로 에티오피아와 남수단, 서쪽으로 우간다, 남쪽으로 탄자니아와 [?는]¶ 국경을 맞닿고[VP] 있다.

```
"ZA" : [
        {
          "id" : 0,
          "type" : "s", // 주어
          "head_wid" : 1, // 2번째 어절(면해)
          ...
        }, …
       ]
```

나) 동사 표현 어구에서 필수 성분 중 생략된 것이 여러 개가 있다면 각 성분을 모두 생략어로 태깅한다.

[?가]¶ [?를]¶ 출시한지✱[VP] 6개월이 지나 가격이 많이 떨어진 상태다.

```
"ZA" : [
        {
          "id" : 0,
          "type" : "s", // 주어
          "head_wid" : 0, // 1번째 어절(출시한지)
          ...
        },
        {
          "id" : 1,
          "type" : "o", // 목적어
          "head_wid" : 0, // 1번째 어절(출시한지)
          ...
        }
       ]
```

다) 여러 절로 구성된 문장에서 개별적인 절 단위에서 생략되어 있는 문장 성분이 있다면 생략어로 태깅한다.

대사의 오용과 남용을 강하게 성토한 그는 1517년 95개 논제를 게시함으로써✱[VP_AJT] 도미니코회 수사이자 대사령 설교 담당자인 요한 테첼에 [?는]¶ 맞섰다✱[VP].

```
"ZA" : [
        {
          "id" : 0,
          "type" : "s", // 주어
          "head_wid" : 17, // 18번째 어절(맞섰다)
          ...
        }
       ]
```

라) 주어 및 목적어 표현 어구에서 생략한 정보성이 있는 관형어를 생략어로 태깅한다.

[?의]¶ 수도는✱[NP_SBJ] 나이로비이며 [?의]¶ 공용어는✱[NP_SBJ] 영어와 스와힐리어이다.

```
"ZA" : [
        {
          "id" : 0,
          "type" : "a", // 관형어
          "head_wid" : 0, // 1번째 어절(수도는)
          ...
        }, …
       ]
```

3.3. 선행어 태깅

선행어 태깅은 앞서 제시한 선행어 후보를 태깅 작업

자가 정한 순서에 따라 후보들을 검토한 뒤, 해당 선행어를 복원하였을 때 필자가 의도했던 의미에 따라 명확해진 정보가 주어지는지 확인하는 과정을 거친다. 구체적인 선행어 태깅 사례들을 살펴보면 다음과 같다.

가) 해당 문서 표제어 또는 제목 내의 표현 일부가 선행어가 될 수 있다면 선행어로 태깅한다.

(표제어 : 케냐[T])
케냐 공화국 또는 케냐는 동아프리카의 공화국이다. [케냐[†]는]¶ 인도양에 면해[✱][VP] 있으며 북동쪽으로 소말리아, 북쪽으로 에티오피아와 남수단, 서쪽으로 우간다, 남쪽으로 탄자니아와 [케냐[†]는]¶ 국경을 맞닿고[✱][VP] 있다.

```
"ZA" : [
        {
          "id" : 0,
          "type" : "s", // 주어
          "head_wid" : 1, // 2번째 어절(면해)
          "ant_text": "케냐",
          "ant_sid" : -1, // 미 존재 문장
          "ant_wid" : -1, // 미 존재 어절
          "ant_is_title" : 1 // 표제어
        }, …
       ]
```

나) 해당 단락 및 문장 내에 포커스가 선행어가 될 수 있다면 선행어로 태깅한다.

문재인 대통령은 21일 '경제 사령탑'인 경제부총리 겸 기획재정부 장관 후보자에 김동연(60) 아주대 총장, 외교부 장관 후보자에 여성인 강경화(62) 유엔 사무총장 정책특보를 각각 내정했다. … 김동연 부총리 후보자[†]는 충북 음성 출신으로 '고졸신화의 인간승리 드라마'로 불린다. [김동연 부총리 후보자[†]는]¶ 덕수상고 졸업 뒤 은행에 취직해[✱][VP] 직장생활을 하며[✱][VP] 행정고시와 입법고시에 동시 합격한[✱][VP_MOD] 입지전적의 인물로 평가받는다[✱][VP].

```
"ZA" : [
        {
          "id" : 0,
          "type" : "s", // 주어
          "head_wid" : 4, // 5번째 어절(취직해)
          "ant_text": "후보자",
          "ant_sid" : 6, // 7번째 문장
          "ant_wid" : 2, // 3번째 어절(후보자는)
          "ant_is_title" : 0 // 표제어 아님
        }, …
       ]
```

다) 일반적으로 생략어 앞에서 나타난 멘션들 중 선행어를 찾아 태깅한다.

… 비텐베르크 대학교의 요한 스타우피츠 교수[T]는 루터가 성서에 대해 진지하게 공부하면 평안을 찾을 것이라고 생각하였다. 그래서 [요한 스타우피츠 교수[†]는]¶ 그를 성서학 교수사제로 임명하였는데[✱][VP], 스타우피츠 교수의 결정은 루터가 신앙적인 고민을 해결하는데 도움이 되었다.

```
"ZA" : [
        {
          "id" : 0,
          "type" : "s", // 주어
          "head_wid" : 4, // 5번째 어절(임명하였는데)
          "ant_text": "교수",
          "ant_sid" : 26, // 26번째 문장
          "ant_wid" : 4, // 5번째 어절(교수는)
          "ant_is_title" : 0 // 표제어 아님
        }
       ]
```

라) 중요한 개체를 나중에 말하는 화법 등에서는 생략어 뒤에서 나타난 멘션들 중에서도 선행어를 찾아 태깅한다.

제2차 세계 대전 당시 [이 영국 수학자[T]는]¶ 독일군 암호를 풀어[✱][VP] 전쟁을 승리로 이끌었으며[✱][VP] 컴퓨터의 원조인 자동 기계 이론을 개척했다[✱][VP]. 이 영국 수학자[†]는 과연 누구일까?

```
"ZA" : [
        {
          "id" : 0,
          "type" : "s", // 주어
          "head_wid" : 6, // 7번째 단어(풀어)
          "ant_text": "수학자",
          "ant_sid" : 1, // 2번째 문장
          "ant_wid" : 2, // 3번째 단어(수학자는)
          "ant_is_title" : 0 // 표제어 아님
        }, …
       ]
```

마) 암묵적으로 또는 상식적으로 알고 있거나 아직 알려지지 않은 개념이 선행어로 될 수 있다면 태깅한다.

케냐 공화국 또는 케냐는 동아프리카의 공화국이다. … 동아프리카에서 [어느 발굴가[†]에 의해]¶ 발견된[✱][VP_MOD] 화석에 따르면 조상이 2백만 년 전 이 지역에서 살았다고 한다.

```
"ZA" : [
        {
          "id" : 0,
          "type" : "s", // 주어
          "head_wid" : 1, // 2번째 어절(발견된)
          "ant_text": "[Unknown]", // 알 수 없음
          "ant_sid" : -1, // 미 존재 문장
          "ant_wid" : -1, // 미 존재 어절
          "ant_is_title" : 0 // 표제어 아님
        }
       ]
```

4. 생략어복원 말뭉치 구축 도구

엑소브레인 언어분석 말뭉치는 형태소분석, 어휘의미분석, 개체명인식, 구문분석, 의미역인식, 상호참조해결, 생략어복원에 대한 언어분석 정답을 제공한다. 현재까지 공개된 엑소브레인 언어분석 말뭉치는 언어분석 기술 개발을 위한 학습용으로는 그 양이 많지는 않으나, 동일 문장에 대해서 형태소분석부터 어휘의미분석, 개체명인식, 구문분석, 의미역인식, 상호참조해결, 생략어복원까지의 언어분석 정답을 포함하고 있기 때문에, 세부 언어분석 기술 뿐만 아니라 전체 언어분석 파이프라인을 평가하기 위한 용도로도 활용이 가능하다.[17-18]

엑소브레인 언어분석 말뭉치에 생략어복원을 태깅하기 위하여 반자동 태깅 도구를 가이드라인 수립과 함께 구축하였다. 엑소브레인 언어분석 결과를 고려하면서 생략어복원에 대한 태깅을 진행한다. 다만, 엑소브레인 언어분석이나 원문 상의 오류에 대해서는 작업자가 감안하고 작업해야 하며, 알려지지 않은 선행어에 대한 직접적인 유추는 하지 않고 [Unknown]으로 선행어 텍스트를 고정하여 태깅한다. 제공되는 주요한 기능은 다음과 같다.

- 작업 목록 및 작업 진행 상황 확인
- 작업 환경 설정 및 태깅 결과 시각화

- 각 문장 내 단어의 의존 관계 정보 확인
- 문서 내 존재하는 선행어 태깅
- 문장 내 존재하는 생략어 태깅

기존에는 작업자가 컴퓨터에서 툴을 이용하여 태깅한 것을 모아서 검토자에게 전달하여 일괄적으로 검토하는 방식이었다. 보다 효율적인 태깅 작업을 위하여 새롭게 구축한 방식은 서버에서 제공하는 태깅 대상 문서에 대해서 권한이 있는 작업자들과 검토자가 실시간으로 작업하고 함께 검토할 수 있는 환경이며, 태깅 작업자가 웹 브라우저에서 작업한 내용이 Json 포맷으로 서버에 실시간으로 저장되도록 설계하였다.

그림 1. 생략어복원 태깅 도구의 편집 화면

5. 결론

본 논문에서는 언어 사용에서 발생하는 생략된 정보를 명확하게 밝히기 위한 생략어복원 문제에 대해 한국어 생략어복원에 대해 정의하고 가이드라인을 제안하였다. 본 가이드라인을 통해서 한국어 생략어복원 말뭉치를 구축하는데 있어서 고려되는 지배소, 생략어, 선행어에 대한 대상을 정하였고, 태깅 포맷과 함께 실제 태깅 사례들을 살펴보았다. 우리는 본 가이드라인을 이용하여 종전에 개발된 한국어 생략어복원 시스템을 개선해 나가고 있으며, 새롭게 고려되는 개념과 방법론을 통하여 궁극적으로 엑소브레인 시스템의 품질이 보다 향상되기를 기대한다. 다만, 정보 추출 및 질의 응답 관점에서 도움이 되는 생략된 정보들을 찾는 데에 초점을 두었기 때문에 모든 생략된 정보들을 찾는 데에는 한계점이 존재한다. 향후, 본 연구에서 제안한 가이드라인을 통한 말뭉치 구축과정에서 발생할 수 있는 불분명한 기준 등에 대해서는 추가적인 개선이 필요할 것이며, 기술 개선과 함께 포커스에 대한 구체적인 사항도 단계적으로 가이드라인에 포함시킬 것이다.

감사의 글

본 연구는 과학기술정보통신부 및 정보통신기술진흥센터의 정보통신·방송 연구개발 사업의 일환으로 수행하였음. [2013-0-00131, (엑소브레인-1세부) 휴먼 지식증강 서비스를 위한 지능진화형 WiseQA 플랫폼 기술 개발]

참고문헌

[1] G. Hirst, *Anaphora in Natural Language Understanding*. Springer Verlag, Germany, 1981.
[2] 황민국, 김영태, 나동열, 임수종, "무형대용어 해결 기술을 이용한 백과사전 표제어 복원," 제26회 한글 및 한국어 정보처리 학술대회 논문집, pp. 65–69, 2014.
[3] 황민국, 김영태, 나동열, 임수종, 김현기, "Structural SVM을 이용한 백과사전 문서 내 생략 문장성분 복원," 지능정보연구, vol. 21, no. 2, pp. 131–150, Jun. 2015.
[4] 임수종, 이창기, 장명길, "백과사전 질의응답을 위한 생략된 표제어 복원에 관한 연구," 한국정보과학회 학술발표논문집, vol. 32, no. 2, pp. 541–543, Nov. 2005.
[5] C. Yeh and Y. Chen, "Zero Anaphora Resolution in Chinese with Shallow Parsing," *Journal of Chinese Language and Computing*, vol. 17, no. 1, pp. 41–56, 2007.
[6] D. S. Wu and T. Liang, "Zero anaphora resolution by case-based reasoning and pattern conceptualization," *Expert Systems with Applications*, vol. 36, no. 4, pp. 7544–7551, 2009.
[7] F. Kong and G. Zhou, "A Tree Kernel-based Unified Framework for Chinese Zero Anaphora Resolution," in *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, pp. 882–891, 2010.
[8] C. Chen and V. Ng, "Chinese Zero Pronoun Resolution with Deep Neural Networks," in *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*, pp. 778–788, 2016.
[9] Y. Qingyu, Z. Weinan, Z. Yu, and L. Ting, "A Deep Neural Network for Chinese Zero Pronoun Resolution," in *Proceedings of the 26th International Joint Conference on Artificial Intelligence*, pp. 3322–3328, 2017.
[10] R. Iida, K. Inui, and Y. Matsumoto, "Exploiting Syntactic Patterns as Clues in Zero-Anaphora Resolution," in *Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics*, pp. 625–632, 2006.
[11] R. Iida, K. Inui, and Y. Matsumoto, "Zero-Anaphora Resolution by Learning Rich Syntactic Pattern Features," *ACM Transactions on Asian Language Information Processing*, vol. 6, no. 4, 2007.
[12] R. Sasano, D. Kawahara, and S. Kurohashi, "A Fully-Lexicalized Probabilistic Model for Japanese Zero Anaphora Resolution," in *Proceedings of the 22nd International Conference on Computational Linguistics*, pp. 769–776, 2008.
[13] K. Imamura, K. Saito, and T. Izumi, "Discriminative Approach to Predicate-Argument Structure Analysis with Zero-Anaphora Resolution," in *Proceedings of the ACL-IJCNLP 2009 Conference Short Papers*, pp. 85–88, 2009.
[14] R. Iida and M. Poesio, "A Cross-Lingual ILP Solution to Zero Anaphora Resolution," in *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics*, pp. 804–813, 2011.
[15] C. Park, K.-H. Choi, C. Lee, and S. Lim, "Korean Coreference Resolution with Guided Mention Pair Model Using the Deep Learning," *ETRI Journal*, vol. 38, no. 6, pp. 1207–1217, Dec. 2016.
[16] H. Lee, A. Chang, Y. Peirsman, N. Chambers, M. Surdeanu, and D. Jurafsky, "Deterministic Coreference Resolution Based on Entity-Centric, Precision-Ranked Rules," *Computational Linguistics*, vol. 39, no. 4, pp. 885–916, 2013.
[17] 임준호, 배용진, 김현기, 김윤정, 이규철, "의존 구문분석을 위한 한국어 의존관계 가이드라인 및 엑소브레인 언어분석 말뭉치," 제27회 한글 및 한국어 정보처리 학술대회 논문집, pp. 234–239, 2015.
[18] 임수종, 권민정, 김준수, 김현기, "ExoBrain을 위한 한국어 의미역

가이드라인 및 말뭉치 구축,” 제27회 한글 및 한국어 정보처리
학술대회 논문집, pp. 250–254, 2015.

RNN 문장 임베딩과 ELM 알고리즘을 이용한 금융 도메인 고객상담 대화 도메인 및 화행분류 방법

오교중○, 박찬용, 이동건, 임채균, 최호진

한국과학기술원, 전산학부
{aomaru, ptparty, hagg30, rayote, hojinc}@kaist.ac.kr

RNN Sentence Embedding and ELM Algorithm Based Domain and Dialogue Acts Classification for Customer Counseling in Finance Domain

Kyo-Joong Oh○, Chanyong Park, DongKun Lee, Chae-Gyun Lim, Ho-Jin Choi
KAIST, School of Computing

요 약

최근 은행, 보험회사 등 핀테크 관련 업체에서는 챗봇과 같은 인공지능 대화 시스템을 고객상담 업무에 도입하고 있다. 본 논문에서는 금융 도메인을 위한 고객상담 챗봇을 구현하기 위하여, 자연어 이해 기술 중 하나인 고객상담 대화의 도메인 및 화행분류 방법을 제시한다. 이 기술을 통해 자연어로 이루어지는 상담내용을 이해하고 적합한 응답을 해줄 수 있는 기술을 개발할 수 있다. TF-IDF, LDA, 문장 임베딩 등 대화 문장에 대한 자질을 추출하고, 추출된 자질을 Extreme learning machine(ELM)을 통해 도메인 및 화행 분류 모델을 학습한다.

주제어: 대화 시스템, 자연어 이해, 문장 임베딩, 화행 분류

1. 서론

올해 들어 국내 은행, 증권, 신용평가 등 금융 관련 도메인에서는 비대면 상담 서비스를 위한 챗봇 시스템의 도입이 활발히 일어나고 있다. 챗봇은 시간과 장소에 구애 받지 않고 고객상담 서비스를 제공할 수 있다. 이를 통해 콜센터 내 야간 및 주말 고객상담 인력을 줄일 수 있으며, 단순 반복 질문에는 자동으로 대처하고, 상담사는 보다 복잡한 상담 사례에 집중할 수 있다. 현재 하나은행의 핀크, 우리은행의 위비톡 등 주요 은행을 시작으로, 카카오뱅크의 챗봇, K뱅크의 핑거뱅킹 등 인터넷 전문은행에서도 챗봇의 도입이 가속화 되고 있다.

현재까지 개발된 고객상담 챗봇 시스템은, 메뉴 기반으로 답변 가능한 질문을 객관식으로 제공하거나, 사전에 정의된 시나리오에 기반하여 준비된 질문을 하고 사용자의 답변과 유사한 시나리오의 응답을 제공하고 있다. 이 같은 서비스 형태는 모바일뱅킹의 새로운 인터페이스 방식일 뿐이며, 상품 소개와 정보 조회 등의 단순한 서비스만 제공 가능하다. 지속적으로 고도의 자연어처리기술과 기계학습 기반의 인공지능 기술을 통하여 사람처럼 대화를 하는 대화 시스템의 연구와 개발이 필요하다.

이 같은 인공지능 기술에 기반한 대화 시스템을 개발하기 위해서는, 전화 등 음성 대화를 문자형 데이터로 바꿔주는 STT 기능, 기존의 자연어처리(NLP), 기계학습 기술에 기반한 상담 대화의 자연어 이해(NLU) 기술, 응답 내용을 결정하고, 자연어 대화를 생성(NLG)하는 응답 기술 등 높은 수준의 대화 기술이 필요하다.

본 논문에서는 대화형 시스템을 위한 자연어 이해 기술에 초점을 맞춘다. 그 중에서 비정형 정보에 해당하는 고객상담 대화의 도메인 및 화행 분류 기술에 대해 다룬다. 도메인 및 화행 분류는 질의에 정확하게 응답하기 위해서 수행되어야 하는 기술이다.

본 논문에서 제안하는 도메인 및 화행 분류 정보는 금융 도메인 고객상담 서비스에 특화된 자연어 대화 문장에 대한 비정형 정보이며 기계학습 기반의 접근 방법을 적용하여 분석한다. 전처리 과정으로써 고객상담 대화 말뭉치를 워드임베딩(Word2Vec) 모델을 통해 형태소 수준의 언어 모델을 학습하고, TF-IDF로 추출된 주요 키워드, LDA모델을 통해 얻은 분류 카테고리 자질을 입력 정보로 ELM 알고리즘을 사용한 분류 모델을 학습하여 입력된 상담 대화의 도메인과 화행을 분류한다.

2. 관련 연구

챗봇이란, 텍스트나 음성 등을 이용하여 인간과 대화를 수행하는 컴퓨터 프로그램을 말하며, 기존의 토크봇(Talbots), 채터봇(chatterbot), IM봇, 대화형 에이전트(interactive agent), 인공 대화 개체(artificial conversational entity) 등을 지칭한다. 챗봇 기술은 기존의 대화 시스템(dialog system) 응용에서 고객 상담, 정보 획득과 같은 실증적인 문제를 풀기 위해서 시작되었으며, 의미, 화행, 화용 분석 등 복잡한 자연어 처리와 인공지능 기술을 필요로 한다.

최근 온라인 쇼핑몰과 유통 업계에서 대화형 시스템을 도입하여 사용자에게 맞춤형 상품을 추천하거나[1], 상담원을 대신하여 빈번한 상담에 대한 답변을 해주는 대화형 시스템[2]을 개발한 사례가 있다. 그 외에 법률 상담[3]과 심리치료와 같은 헬스케어 분야[4] 등 다양한 산업 분야에서도 챗봇 기술을 도입한 응용 시스템의 연구 개발 사례가 시도되고 있다.

금융 분야에서도 챗봇과 관련된 연구 및 개발이 이루어지고 있다. [5]에서는 뱅킹 시스템을 위한 인공지능 챗봇을 최초로 제시하였으며, [6]에서는 뱅킹서비스를 돕기 위하여, 자연어 이해 및 생성 등 챗봇 시스템의 여러 접근 방법을 정리하였다. [7]에서는 고도의 자연어 이해 기술을 적용하여 담화 구조를 가진 대화를 관리하고, 이를 챗봇 서비스에 적용한 연구 사례가 발표되었다. 이처럼 자연어 이해 기술은 챗봇을 구현하는데 있어 반드시 필요한 기술이다.

본 논문은 챗봇의 응답 모델을 구현하기 위한 초기 단계로, 자연어 이해 기술 중 하나인 대화의 도메인과 화행을 분석하는 방법에 관해 기술한다. 본 연구팀은 기존 연구 [8]에서 tripadvisor.com의 여행지에 관한 사용자 리뷰를 바탕으로 여행 의도를 분석하는 방법을 연구하였으며, 이를 기반으로 여행지를 추천하는 기술을 개발하였다. 본 연구에서는 기존에 연구한 의도 분류 방법을 금융 도메인의 자연어 대화 데이터에 적용하여, 대화 도메인 뿐만 아니라 화행까지 분류함으로써 챗봇의 응답 성능을 향상시키는데 기여한다.

본 연구에서는 전처리 기술로 Word2Vec[9] 알고리즘을 적용한 워드임베딩 기술을 사용한다. 은행 고객상담 대화 말뭉치로부터 신경망을 통해 언어 모델을 학습하며, 단어나 구를 원하는 크기의 벡터로 표현할 수 있다. 고객상담 말뭉치에 사용된 어휘들의 언어 모델을 도메인에 특화하여 학습하고, TF-IDF로부터 추출된 중요 키워드 자질과 입력 문장을 벡터 형태의 학습 자질로 바꾸는데 사용되었다.

이렇게 얻어진 벡터 형태의 학습 자질을 이용하여 지도 학습 방법의 분류 모델을 구현한다. 본 논문에서는 학습기반의 분류 알고리즘으로 ELM 알고리즘 [10]을 사용한다. 얕은 신경망 혹은 적은 수의 은닉 층을 가지는 신경망을 이용하여 출력 층과 은닉 층 사이의 가중치는 한번만 연산함으로써, 신경망 모델의 학습 속도가 느린 단점을 보완한 분류 모델로, 최근 여러 연구에서 SVM과 비슷하거나 더 좋은 분류 결과를 보이는 것으로 보고되고 있다.

3. 고객상담 대화 화행분류

본 논문에서는 금융 도메인 고객상담 대화의 도메인과 화행 분류 방법을 제안한다. 크게 전처리, 자질 추출, 도메인 및 화행분류 3개의 단계로 이루어진다. 그림 1은 고객상담 대화에 대한 도메인 및 화행 분류 방법에 대한 개념도이다. 도메인 분류는 대분류와 세부 분류 2단계로 나눠지며, 두 분류 결과에 따라 화행이 분류된다.

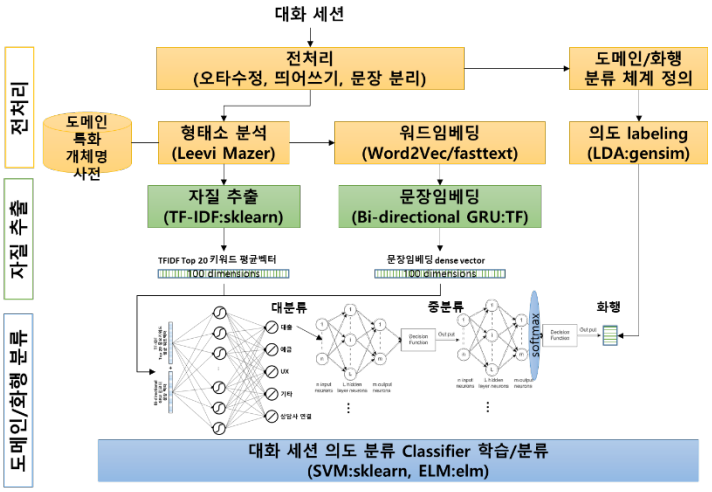


그림 1 고객상담 대화 화행 분류 방법

3.1 전처리 과정

이 단계에서는 고객상담 대화 데이터의 내용을 비식별화 및 정제하고, 언어모델 학습을 위한 말뭉치 구축과 형태소 분석 및 개체명 사전 구축 등, 자질 추출을 위한 전처리 과정이다. 고객상담 대화 말뭉치는 메신저나 대화 플랫폼에서 이루어지는 자연어 문장으로 이루어져 있으며 오타, 띄어쓰기 오류, 개인 정보가 포함된 대화가 많이 포함된다. 또한 문장을 문장 부호 기준으로 간단하게 분리하기 어려우며, 한 문장이 여러 대화에 걸쳐 쪼개지기도 한다. 따라서 연속된 대화를 처리하고 문맥의 흐름에 따라 대화를 분리해야 한다.

그림 2는 고객상담 대화 데이터의 예시이다. 그림2 와 같이 연속된 대화에 대해서는 문장을 결합하고 문맥 상 질문이 변경되었을 때는 대화 세션을 분리하여 질의-답변 말뭉치를 구축하여야 한다.

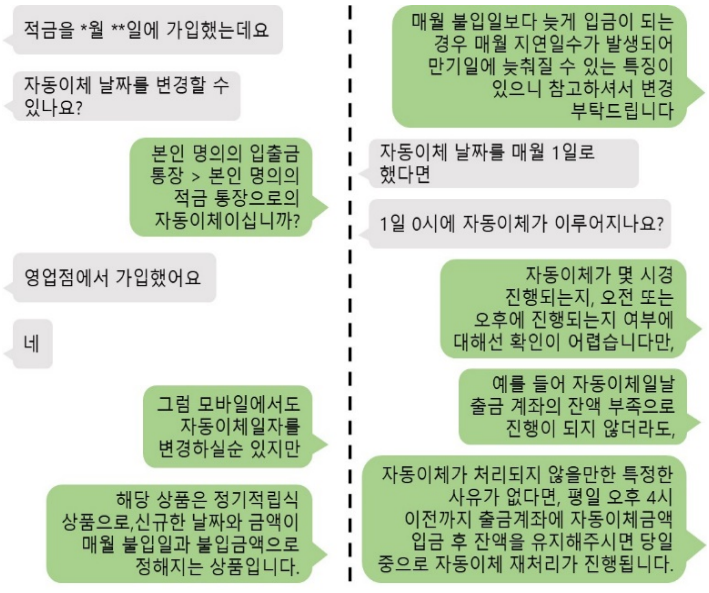


그림 2 금융 도메인 고객상담 대화 예시

다음 단계에서 대화 문장의 벡터 화와 도메인 및 화행 분류를 위해서는 학습 자질로 정해진 크기의 실수 벡터를 사용해야 한다. 이를 위해 Word2Vec 모델을 통한 언어모델 학습이 전처리 과정으로 필요하며, 본 연구에서는 형태소 분석을 통한 형태소 수준의 워드임베딩 모델[11]을 생성하였다. 워드임베딩 학습에 이용된 말뭉치는 그림2와 같은 고객상담 대화와 한글 위키피디아의 분류: 금융 관련 문서에 등장한 한글 30만 문장이 사용되었다.

3.2 도메인 및 화행 분류 체계 정의

이 단계는 지도 학습을 위한 학습 데이터를 생성하는 단계이다. 도메인 및 화행 분류 카테고리를 정의하고 학습 기반의 분류 모델을 위한 도메인 및 화행 레이블링 데이터를 구축한다. 이를 위해 우선 도메인 및 화행의 분류 체계를 정의 해야한다.

본 연구팀은 Latent Dirichlet allocation(LDA) 모델을 사용하여 데이터 기반으로 클러스터를 정의하였다. 기존의 고객센터에서 사용하던 상담 분류 체계에 기초하여, 4가지 대 분류(대출, 예금, UX, 기타)에 대하여 각 대 분류 별 7~15개 의 중분류로 이루어진 초기 분류체계를 가지고, 대 분류로 나누어진 대화 데이터를 LDA 클러스터링 (클래스 개수 40개)을 통해 초기 분류 체계와 매핑 또는 세분화하는 방법을 거쳐 최종적으로 도메인 세부 분류와 화행 수준의 구분까지 수행하였다. 이 방법은 도메인 전문가의 개입없이 고객상담 서비스 제공자의 입장에서 분류 체계를 구축하는 방법으로, 데이터만 있으면 비전문가도 도메인 분류와 화행 분류 체계를 정의할 수 있다.

그림 3은 실제로 수행한 대출에 해당하는 대분류의 세부 도메인과 화행 분류 체계를 보여준다. 붉은색으로 표시한 화행은 특정 세부 분류에 특화된 화행을 나타낸다.

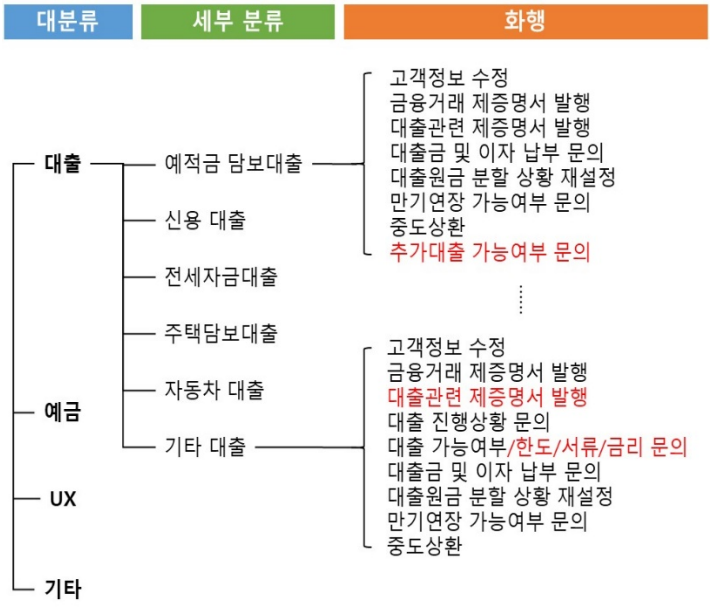


그림 3 도메인 및 화행 분류 체계 및 대출 분야 예시

3.3 학습 자질 추출 및 문장 임베딩

본 논문에서는 화행분류를 위한 입력 자질로 TF-IDF, GRU 문장인코딩 결과를 사용한다.

TF-IDF(Term frequency and Inverse Document Frequency)는 분석하려는 대화 문장이나 문서에서 다른 대화 세션에 비해 중요도가 큰 키워드를 추출한다. 이를 통해 중요 키워드 자질을 화행분류 학습에 반영할 수 있다. 추출된 Top-20 중요 키워드의 워드임베딩 평균 벡터를 입력 벡터의 일부로 사용한다. 추가로 문장에 사용된 주요 개체명의 워드 벡터도 TF-IDF 자질과 함께 반영한다. 주요 개체명은 응용 도메인과 서비스에 맞추어 별도로 제작된 뱅킹 서비스를 위한 개체명 사전으로, 예금 또는 대출 상품명 등 직접적으로 대화의 도메인을 반영하는 정보이다.

마지막 RNN 문장 인코딩 자질은 문장의 의미적 자질을 화행 분석에 사용하기 위해 사용된다. 그림 4는 문장 임베딩에 사용한 RNN 인코더 모델을 나타낸다. Bi-directional GRU dynamic RNN모델을 사용하였으며, 입력층에는 각 단어의 워드임베딩 결과가 순차적으로 입력된다. 최종적으로 Forward layer와 backward layer의 마지막 GRU 셀의 RNN state을 이어 붙인 dense 벡터를(200차원) 문장 임베딩 결과로 사용한다.

RNN 문장 인코더의 학습 방법은 그림 5와 같다. Bidirectional GRU로 이루어진 sequence to sequence 모델을 그림 5와 같이 같은 문장으로 학습한다. 고객 상담 대화 말뭉치의 모든 문장에 대하여 loss가 수렴할 때까지 반복 학습한 후, 인코더 부분만 분리하여 새로운 입력 문장에 대해 문장 벡터를 추출한다.

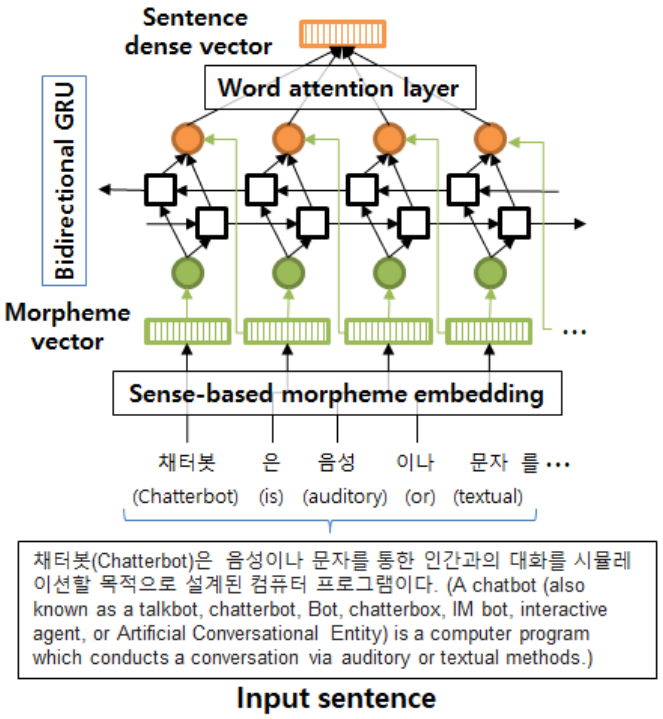


그림 4 문장 임베딩을 위한 RNN 인코더 개념도

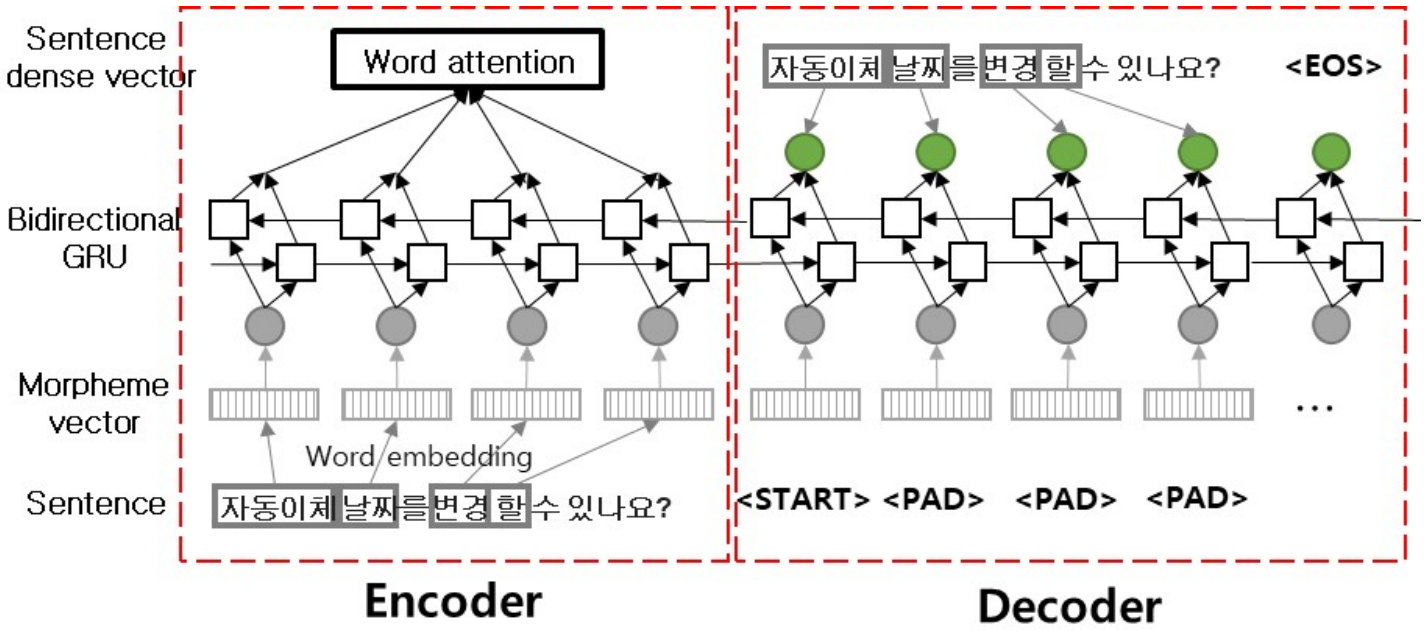


그림 5 Bidirectional GRU RNN 인코더 학습 과정

3.4 지도 학습 기반 대화 문장 도메인 및 화행 분류

Extreme Learning Machine(ELM)은 단일 또는 소수의 은닉 층으로 구성된 전방 전달 신경망(feed forward neural network)의 한 종류로 주로 분류, 회귀 분석, 클러스터링 등에 많이 이용된다. 임의의 가중치를 갖는 입력 층과 은닉 층과, 한 차례의 역전사 과정을 통해 가중치를 갖는 은닉 층과 출력 층 구조의 신경망으로 구성된다. 낮은 오류로 선형 분류 모델을 학습하는 SVM의 장점과, 다차원의 공간을 접거나 구부리는 신경망 분류 모델의 장점을 차용하여 multi-class classification에서 속도와 정확도 면에서 좋은 성능을 보이는 분류 모델이다.

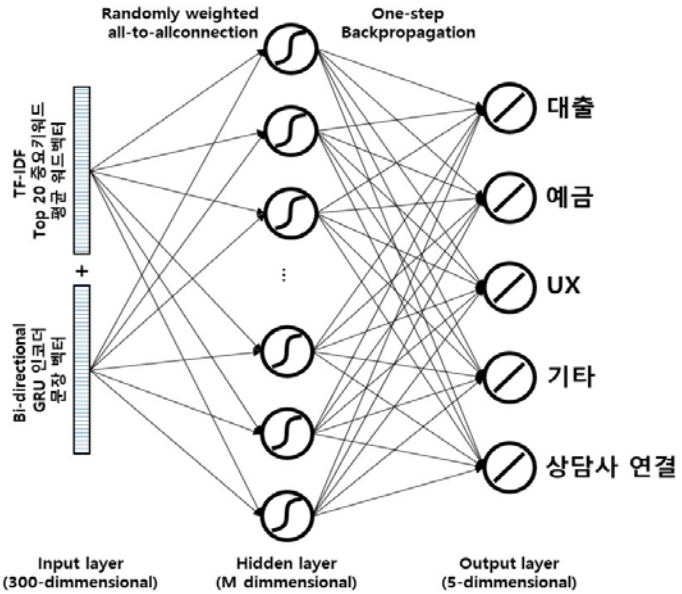


그림 6 ELM 알고리즘 기반 도메인 분류 모델

본 논문에서는 대화 세션의 도메인 대분류와 대분류 내 세부 화행을 분류하기 위한 총 3개의 ELM classifier가 사용되었으며 대분류, 세부 분류, 화행 각각의 레이블링 된 데이터를 통해 학습되었다.

첫번째 ELM classifier는 그림 6와 같이 5개의 대분류(대출, 예금, UX, 기타, 상담원 연결)에 대한 분류를 수행한다. 초반 분류 과정에서 상담사 연결 확률을 계산함으로써, 챗봇이 응답할 수 없는 대화에 대해서 빠르게 상담사에게 넘긴다.

두번째 ELM classifier는 입력 문장의 벡터와 대분류 자질을 함께 이용하여 세부 분류를 분류한다. 대출에는 6개, 예금에는 15개의, UX은 5개, 기타에는 4개의 세부 분류 체계가 있다. 출력 층에서 softmax 활성화 함수를 사용하여 [0, 1] 사이의 정규화된 수치를 얻음으로써 multi-label 분류를 할 수 있도록 모델을 구성하였다.

세번째 ELM classifier는 첫번째 대분류와 세부 분류까지 입력 자질로 활용하여 화행 분류를 수행하는 모델이다. 화행의 경우 대분류, 세부 분류와 독립적으로 대화의 행위적 측면에 대한 정보이므로 세부 분류의 아래 단계에서 수행된다.

4. 실험 설계

4.1 실험 데이터 설명

기존의 은행 및 금융권에서는 고객과의 상담 대화를 공개하지 않기 때문에 기계 학습과 관련된 연구 사례가 많지 않았다. 본 연구는 모바일 뱅킹 앱에서 발생한 비식별화 처리된 고객상담 대화 데이터의 일부 샘플을 이용하여 수행하였으며, 추후 더 많은 양의 고객상담 데이터를 정제하여 화행분류 모델의 성능 평가 및 고도화 연구를 추가적으로 수행될 것이다.

그림 5에서도 알 수 있듯이, 상담 고객이 원하는 은행

업무로는 기능을 찾기 위한 상담, 특정 상품에 대한 구체적인 조건이나 설명 제공, 고객의 금융 정보(자동이체 금액, 납일 일, 이체 한도 상품 등) 조회/변경/해지 등, 상품 추천, 오류 신고 및 확인 등이 있다.

본 논문에서 문장 임베딩 모델에 사용된 한글 말뭉치는 일부 대화 데이터와 금융 도메인 관련 한글 위키피디아 문서, 뉴스 기사에 등장한 문장 30만 문장으로 학습하였다, 도메인 및 화행 분류 모델의 성능 평가를 위한 평가셋으로는 대분류 별로 50개씩 총 200개의 고객상담 질의 문장이 사용되었다.

4.2 부분적 실험 결과

본 논문에서 제시한 고객상담 대화 화행분류 연구는 현재 수행 중인 연구로, 이 논문에서는 부분적 실험 결과인 대분류 분류 결과만을 공개한다.

표 1은 ELM 알고리즘을 사용한 분류 모델의 대분류 정확도이다. 추후 연구에서 보다 많은 평가 셋 대화 세션 데이터를 정제하여 화행분류를 수행한 결과를 공개하고자 한다.

표 1 은행업무 대분류 분류 결과

평가항목	대출	예금	UX	기타	평균
평가 문장 수 (개)	50	50	50	50	50
분류 일치 문장 수 (개)	44	46	34	41	41.3
정확도(%)	88	92	68	82	82.5

5. 결론

본 논문은 은행 업무과 관련된 고객 상담을 위한 챗봇을 개발하기 위한 자연어 이해 과정의 일부로, 응답 결정하기 위하여 대화의 도메인과 화행을 분류하는 방법에 대해 기술하였다.

LDA 클러스터링을 통해 대화 데이터의 도메인과 화행 분류 체계를 정의하고, 전처리 과정으로 금융 도메인에 특화된 언어 모델을 워드임베딩 모델을 통해 학습하고, 대화 세션 내 TF-IDF로 추출된 주요 키워드와 개체명을 통해 얻은 입력 자질과, RNN 인코더를 통해 문장 벡터를 사용하여 ELM 알고리즘을 사용하여 도메인과 화행을 분류한다.

이 논문은 현재 진행 중인 연구 개발로 학습데이터를 일부 샘플링 한 데이터를 기반으로 평가셋을 구축하여 작성되었다. 추후 연구를 통해 고객상담 대화 말뭉치를 질의 별로 정제하고, 대화 화행 분류 결과에 따른 응답 방법을 결정하는 대화 모형에 대한 실험과, 고객상담 말뭉치 전수 데이터를 사용하여 응답 성능의 평가를 진행할 예정이다.

감사의 글

본 연구는 미래창조과학부 산업융합원천기술개발사업의 “휴먼 지식증강 서비스를 위한 지능진화형 WiseQA 플랫폼 기술 개발”(과제번호 2013-0-00131)과 ICT유망기술개발지원사업 “지능형 대화 서비스를 위한 화용 및 문맥 분석 기반 대화솔루션 개발”(과제번호 2017-0-00868) 과제의 지원으로 수행되었음.

참고문헌

[1] A. Iftene and J. Vanderdonckt, MOOCBuddy: a chatbot for personalized learning with MOOCs, In proc. of RoCHI, vol. 91, 2016.

[2] A. Xu, Z. Liu, Y. Guo, V. Sinha, and R. Akkiraju, R, A New Chatbot for Customer Service on Social Media, In proc. of ACM CHI 2017, pp. 3506-3510, May 2017.

[3] J. Aguilar, K. Berboso, K. Cayube, R. Sagum, and B. Comendador, PHILEX: Philippine Land Law Expert Chatbot, In proc. of ICCRD 2013, ASME Press, 2013.

[4] K. Oh, D. Lee, B. Ko, and H. Choi, A Chatbot for Psychiatric Counseling in Mental Healthcare Service Based on Emotional Dialogue Analysis and Sentence Generation, In proc. of IEEE MDM 2017 pp. 371-375, May, 2017.

[5] A. Dole, H. Sansare, R. Harekar, and S. Athalye, Intelligent Chat Bot for Banking System, International Journal of Emerging Trends & Technology in Computer Science (IJETTCS), vol. 4 no. 5, 2015.

[6] K. B. Shah, M. S. Shetty, D. P. Shah,and Pamnani, R, Approaches towards Building a Banking Assistant. International Journal of Computer Applications, vol.166, no.11, 2017.

[7] B. Galitsky and D. Ilvovsky, Chatbot with a Discourse Structure-Driven Dialogue Management, In proc. of EACL 2017, 2017.

[8] K. Oh, Z. Kim, H. Oh, C. Lim, and G. Gweon, Travel intention-based attraction network for recommending travel destinations, In proc. of IEEE BigComp 2016, pp. 277-280, , Jan. 2016.

[9] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado,and J. Dean, Distributed representations of words and phrases and their compositionality, In Advances in neural information processing systems, pp. 3111-3119, 2013.

[10] G. B. Huang, Q. Y. Zhu, and C. K. Siew, Extreme learning machine: theory and applications, Neurocomputing, vo.70, no.1, pp.489-501, 2006.

[11] 이동건, 오교중, 최호진, 허정, 질의응답 시스템에서 형태소임베딩 모델과 GRU 인코더를 이용한 문장유사도 측정, 제28회 한글 및 한국어 정보처리 학술대회 논문집, 2016.

한국어 음소열 기반 워드 임베딩 기술

정의석O, 송화전, 이성주, 박전규

한국전자통신연구원, 음성지능연구그룹
eschung@etri.re.kr, songhj@etri.re.kr, lee1862@etri.re.kr, jgp@etri.re.kr

Korean Phoneme Sequence based Word Embedding

Euisok ChungO, Hwa Jeon Jeon, Sung Joo Lee, Jeon-Gue Park
ETRI, Speech Intelligence Research Group

요 약

본 논문은 한국어 서브워드 기반 워드 임베딩 기술을 다룬다. 미등록어 문제를 가진 기존 워드 임베딩 기술을 대체할 수 있는 새로운 워드 임베딩 기술을 한국어에 적용하기 위해, 음소열 기반 서브워드 자질 검증을 진행한다. 기존 서브워드 자질은 문자 n-gram을 사용한다. 한국어의 경우 특정 단음절 발음은 단어에 따라 달라진다. 여기서 음소열 n-gram은 특정 서브워드 자질의 변별력을 확보할 수 있다는 장점이 있다. 본 논문은 서브워드 임베딩 기술을 재구현하여, 영어 환경에서 기존 워드 임베딩 사례와 비교하여 성능 우위를 확보한다. 또한, 한국어 음소열 자질을 활용한 실험 결과에서 의미적으로 보다 유사한 어휘를 벡터 공간상에 근접시키는 결과를 보여 준다.

주제어: 워드임베딩, 서브워드, 음소열

1. 서론

워드 임베딩 기술은 어휘 목록을 벡터 공간상에 배치하는 기술이다. 이는 유사한 어휘들을 근접한 벡터 공간에 위치하게 한다. Word2vec 공개툴 [1]로 보편화된 워드 임베딩 기술은 다양한 응용 분야에 적용할 수 있다. [2]는 워드 임베딩 기술을 이용하여 텍스트로부터 연결된 문장 체인을 추출하여 도메인 텍스트 확장에 대한 가능성을 보여줬다. 또한 워드 임베딩 기술을 이용하여 클래스 언어모델에 적용하여 성능개선을 시도한 연구도 있었다 [3]. [5]는 한국어 대상의 워드 임베딩 기술로 다양한 학습 패러미터 실험결과를 제시하고 있다. WS353평가셋을 한국어로 번역하여 실험하였는데, 해당 평가셋은 영어 어휘의 다양한 뉘앙스를 기반하고 있어, 해당 워드 유사도 평가셋을 독립적으로 구축하는 것이 옳다고 판단된다.

기존의 워드 임베딩 기술은 미등록어 문제를 갖고 있었다. 즉, 학습 시점에 벡터공간상에 할당할 어휘 목록을 미리 결정해야 했다. 그러나 최근 서브워드 정보 기반 워드 임베딩 연구 [4]는 미등록어 문제 해결 방법을 제시하였다. 이는 각 어휘를 구성 '문자 n-gram'으로 표현하는 방법이다. 학습되는 벡터 값은 '문자 n-gram'을 대상으로 학습되며, 각 어휘는 '문자 n-gram' 벡터 값의 합으로 벡터 공간에 할당된다. 이는 FastText라는 오픈툴로 공개되어 있고, 미리 학습된 결과들을 제공하고 있어, 유용하게 활용할 수 있다. [6]의 경우 한국어 미등록어 워드 임베딩 처리를 위해 음절열 대상 CNN기반 워드 임베딩 기술을 제안하였다. 해당 연구는 응용 DNN 구조에 내재 되었을 경우와 분리되어 단어 단위 입력 레이어로 적용되었을 경우에 대한 비교 후속 연구가 필요하다고 판단된다.

본 논문은 기존 FastText를 재현하여 다양한 자질들을 이용한 워드 임베딩 기술 개발을 위해, 우선 한국어 음소열 기반 워드 임베딩 기술을 검토한다.

2. 서브워드 기반 워드 임베딩

이 장에서는 기존의 워드 임베딩 기술과 서브워드 임베딩 기술의 차이점을 [4]를 참조하여 기술한다. 워드 임베딩 기술의 경우 T 개의 어휘로 구성된 입력 텍스트에 대하여 하나의 단어 w_t의 컨텍스트 어휘 w_c에 대하여 다음의 로그 우도값 (1)을 최대화는 방식으로 기술 할 수 있다. 이는 [1]에 따르면 Skip-gram모델로 명명 된다.

$$\sum_{t=1}^{T}\sum_{c\in C_t}\log p(w_c|w_t) \qquad (1)$$

여기서 $p(w_c|w_t)$는 컨텍스트 어휘의 확률값을 나타내는 소프트맥스 (2)로 기술될 수 있다. 즉, 일종의 W개의 어휘 셋에 대한 언어모델 값이 된다.

$$p(w_c|w_t)=\frac{e^{s(w_t,w_c)}}{\sum_{i=1}^{W}e^{s(w_t,i)}} \qquad (2)$$

(2)에서 $s(w_t,w_c)$는 각 어휘에 해당하는 특정 벡터값들의 스칼라 곱, $u_{w_t}^{\mathrm{T}}v_{w_c}$로 표현할 수 있다. 서브워드 워드 임베딩의 경우, 하나의 어휘를 '문자 n-gram'으로 표현한다고 하였다. "she"의 경우, 최소 3-gram에서 4-gram으로 제한하였을 때, { <sh, <she, she, she>, he> }의 문자 n-gram 집합으로 표현할 수 있고, 해당 문자 n-gram은 벡터값을 갖게 되고, 모든 벡터 값의 합으로 "she"의 벡터값이 결정 된다. 여기서 "<,>"는 단어의 시작과 끝을 표현한다. [4]의 표현을 빌리자면, G개의 문자 n-gram 사전이 주어졌을 때, 어떤 어휘 w가 갖는 문자 n-gram의 집합을 C_w 라고 한다면, C_w 는 문자 n-

gram 사전의 부분 집합이 되고, s(w,c)는 w의 구성 문자 n-gram의 벡터 표현을 z_g라고 했을 때 다음 (3)과 같이 표현된다.

$$s(w,c) = \sum_{g \in C_w} z_g^{\mathrm{I}} v_c \qquad (3)$$

3. 음소열 기반 한국어 워드 임베딩

한국어를 위한 서브워드 접근 방법으로는 문자 n-gram, 자소 n-gram, 음소 n-gram 접근 방법이 가능하다고 판단된다. 본 논문은 문자 n-gram과 음소 n-gram만을 다룬다. 문자 n-gram은 한국어 단어를 구성하는 각 '음절'을 이용하는 반면, 음소 n-gram은 단어의 '발음열'을 이용한다. 직관적으로 음소 n-gram의 경우 자소 n-gram과 차이는 없으리라 본다. 특징을 논의해 보자면, 발음에 대한 표현으로 '독립문'의 경우 '동닙문'이라는 음소열로 표현된다. 즉, 발음기호 'd o N n i xm m u xn'으로 기술 할 수 있다. 음소열은 음성인식에서 사용되는 한국어 g2p (grapheme to phoneme)의 표준 음소셋을 이용한다.[1] 이 접근방법의 장점은 '독선'의 경우 발음은 '독썬'이 되므로, 서브워드 '독'에 대한 분리 표현이 가능하다는 점을 들 수 있겠다. 표 1은 '독립문'의 서브워드 집합 (min=3, max=4)을 보여 준다.

표 1. '독립문'의 서브워드 집합 예

	서브워드 집합
문자 n-gram	{ <_독_립, <_독_립_문, 독_립_문, 독_립_문 _>, 립_문_> }
음소 n-gram	{ <_d_o, <_d_o_N, d_o_N, d_o_N_n, o_N_n, o_N_n_i, …, xm_m_u, xm_m_u_xn, m_u_xn, m_u_xn_> }

4. 실험

한국어 서브워드 워드 임베딩 실험을 위해 c++로 구현된 FastText를 텐서플로우 환경으로 재현하는 작업을 진행하였다. 텐서플로우 튜토리얼에서 제공되는 word2vec 코드를 이용하여, 문자 n-gram 자질을 반영하는 부분과 analogy 평가셋을 사용한 평가를 진행하는 부분을 추가하였다.

실험은 우선 영어환경에서 진행하였다. 100메가 분량의 text8코퍼스를 이용하였고, 어휘셋 빈도수 5이상, 71,290단어, 서브워드 범위 3gram~6gram, 서브워드 총 비율 90%를 이용하였다. 7만여 어휘에 대하여 총 3만 수준의 서브워드 목록이 추출되었다. 배치 크기 64, 임베딩 크기 128, 윈도우 크기 4, 스킵빈도 2, 네거티브 샘플링 개수 10, 서브 샘플링값 1e-3을 적용하였다. 다음 표 2는 word analogy 평가셋을 사용한 평가 결과를 보여준다. 성능 상으로는 기존 word2vec보다 우수한 결과를 보여 준다. 그러나 좀더 대용량 텍스트에 대하여 검증해 볼 필요가 있다고 본다.

표 2. word analogy task 실험 결과

	Accuracy
word2vec	35.8%
sub-word word2vec	39.0%

한국어의 경우 평가셋을 찾을 수 없어, 어휘 유사도, 문장 유사도, 어휘 analogy셋을 직접 구축하기로 시도하는 중이고, 본 논문의 경우 정성 평가와 [6]의 접근방법에 따라 기존 WS353평가셋[2]을 한국어로 변환하여 테스트해 보았다. 사용된 코퍼스는 [2]에서 실험용으로 사용된 82cook코퍼스 중 일부인 35M바이트 분량을 사용하였다. 코퍼스는 word-segmentation 도구를 적용한 상태로 형태소 분석된 결과와 유사하나 어휘 변형이 없는 상태로 보면 된다. 하이퍼 패러미터는 영어 실험과 동일하고, 어휘 셋만 1만 5천 단어로 제한하였다. 정성 평가는 특정 어휘와 벡터 공간상에 근접해 위치한 어휘 출력으로 하였다. 본 논문에서 제안된 음소열 기반 sub-word word2vec (p-sub-word word2vec)와 기존의 word2vec의 유사 어휘 목록 중 차이가 있는 항목에 대하여 표 3에서 기술한다. 또한 기존 문자 n-gram (s-sub-word word2vec) 결과도 포함하였다.

표 3. 유사 어휘 실험 결과 (top5)

어휘	p-sub-word word2vec	s-sub-word word2vec	word2vec
요구_하	요구 요구_했 밝히 지급_하 밝히_고	요구 요청_하 밝히 지키 거부_하	거부_하 적용_하 강조_하 밝히 강화_하
어르신	어르신들 노인 부모_님 부모님 할아버지	노인 어르신들 노인들 할머니 연세	여성_분 노인 분 회원_님 선배_님
시어머니	시어머님 어머니 시아버지 시모 며느리	시아버지 어머니 시모 시어머님 어머님	어머니 시아버지 시모 시엄니 아버지
깨끗_한	깨끗_하 깨끗_해 깨끗 깔끔_한 깔끔_하	깨끗_하 깔끔_한 깔끔 안전_한 깔끔_하	깔끔_한 조용_한 오래된 깨끗_하 어두운
선택	결정 판단 선택_할 선택_한 선택_하	결정 선택_할 선택_했 선택_한 판단	판단 결정 노력 성공 행동

[1] 에트리 음성인식 엔진 내부 dd-g2p, https://itec.etri.re.kr/itec/sub02/sub02_01_1.do

[2] WordSim353 - Sim. and Rel. http://alfonseca.org/eng/research/wordsim353.html

표 3에서 알 수 있는 점은 p-sub-word 실험의 경우 유사 의미의 어휘가 더 근접해 있다는 데 있다. '요구_하'의 경우, word2vec은 유사한 컨텍스트를 보이는 어휘들이 벡터 공간상 근접해 있으므로, 실제 반대 되는 의미를 가진 어휘 들이 유사 어휘로 선정되었으나, p-sub-word의 경우 발음의 유사성이 유사 컨텍스트 클러스터링 현상을 회피할 수 있게 한다고 볼 수 있다. s-sub-word 실험의 경우 표 3 두 실험의 중간 정도 수준의 출력 결과를 보였다.

서론에서 전술했듯이 WS353을 한국어로 변환하는 접근 방법은 부적절하다고 판단된다. 해당 평가셋은 영어 어휘의 다양한 의미를 고려하여 설계 되었다. 따라서 번역 어휘의 선정은 단어 쌍의 할당 점수를 무의미 하게 한다. 그러나, 한국어에 대한 적절한 평가셋을 구할 수 없어, 본 논문은 [6]의 접근방법에 따라, 번역을 통하여 WS353 평가를 진행해 보았다.

표 4. 한국어 WS353 평가

	(min, max) / 총 subword수 / word당 고정 subword 수	WS353 -S	WS353 -R
[6]	-	0.67	0.49
문자 n-gram	(2, 4) / 15k / 21	0.40	0.32
음소 n-gram	(2, 5) / 10k / 56	0.40	0.30
	(2, 4) / 5k / 29	0.34	0.27
	(3, 6) / 24k / 30	0.35	0.33

표 4는 한국어 WS353 평가 결과를 보여 준다. 기존 연구 [6]에 비해서 성능 결과는 좋지 않았다. [6]의 경우 학습 데이터도 어휘 3k 단어, 텍스트 427k 단어로 구성된 소량 뉴스 텍스트로 도출한 결과이다. 본 연구의 경우는 15k단어, 텍스트 8000k 단어를 이용한 결과이다. 두 실험의 WS353 평가셋이 다른 점과 본 연구의 텍스트 도메인이 인터넷 게시판 도메인이라는 차이점이 있다. 향후 다양한 경로로 검토와 보완을 진행할 예정이다.

표 4에서 (min, max)는 서브워드 자질의 (최소 길이, 최장 길이)이다. '총 subword 수'는 1만 5천 단어로부터 추출된 subword 개수를 말한다. '고정 subword 수'는 워드를 구성하는 서브워드 개수의 최장 길이가 된다. 실험 결과는 문자 n-gram과 음소 n-gram 결과의 큰 차이를 보여 주지 못하고 있다. 음소 n-gram 실험 결과들의 차이점은 다양한 패러미터 최적화 작업의 필요성을 보여준다. 실험의 epoch당 성능 개선 수준은 문자 n-gram이 더 안정적인 결과를 보여 줬다. 음소 n-gram의 경우, g2p결과를 이용하므로, 다중 발음이라는 문제점을 갖고 있다. 즉 한 단어를 다양한 발음으로 표현할 수 있다는 점인데, 본 논문에서는 최소 길이의 발음열을 적용하였다. 향후 이 부분에 대하여도 검토해 볼 예정이다.

5. 결론 및 향후 연구

본 논문은 한국어 서브워드 기반 워드 임베딩 기술 개발을 위해 음소열 기반 서브워드 접근 방법을 제시하였다. 이를 위해 FastText를 텐서플로우 환경으로 재현하였고, 영어 환경에서 성능 검증을 진행하였다. 한국어의 경우 음소열 기반 워드 임베딩 도구를 개발하였고, 이를 이용하여 기존의 word2vec와 비교하였다. 정성 실험 결과 좀 더 의미적으로 유사한 어휘가 근접 어휘로 추출되는 결과를 보였다. 반면 한국어 WS353평가에서는 기존 연구보다 좋지 않은 결과를 보였다. 향후 한국어 어휘 유사도, 문장 유사도, 어휘 analogy 평가셋 구축을 진행하여 정량 평가를 시도하여, 다양한 서브워드 자질에 대하여 연구를 진행할 계획이다. 또한, 영어 환경에 음소열 서브워드 자질 실험도 진행하여, 본 연구의 일반성 검증을 진행할 예정이다.

감사의 글

이 논문은 2017년도 정부(과학기술정보통신부)의 재원으로 정보통신기술진흥센터의 지원을 받아 수행된 연구임. (R0126-15-1117, 언어학습을 위한 자유발화형 음성대화처리 원천기술 개발)

참고문헌

[1] T. Mikolov et al., "Distributed Representations of Words and Phrases and their Compositionality", Int. Conf. NIPS, pp. 3111-3119, 2013

[2] E. Chung and G. Park, "Sentence-Chain Based Seq2seq Model for Corpus Expansion", ETRI Journal, Vol. 39, Num. 4, pp. 455-466, Aug. 2017.

[3] 정의석, 박전규, "워드 임베딩과 품사 태깅을 이용한 클래스 언어모델 연구", 정보과학회 컴퓨팅의 실제 논문지, 제22권, 제7호, pp. 315-319, 2016.7.

[4] P. Bojanowski et al., "Enriching Word Vectors with Subword Information", arXiv:1607.04606v2, Jun. 2017.

[5] 최상혁, 설진선, 이상구, "한국어에 적합한 단어 임베딩 모델 및 파라미터 튜닝에 관한 연구", 제28회 한글 및 한국어 정보처리 학술대회 논문집, 2016.

[6] 최상혁, "음절 기반 한국어 단어 임베딩 모델 및 학습 기법", 서울대학교 공학석사학위논문, 2017.

공개와 협업을 통한 세종 형태 분석 말뭉치 오류 개선 방법

한경은○, 백슬예, 임재수

(주)카카오

grace.han@kakaocorp.com, cecil.rosa@kakaocorp.com, jamie.lim@kakaocorp.com

Open Sourced and Collaborative Method to Fix Errors of Sejong Morphologically Annotated Corpora

Gyeong-Eun Han○, Seul-Ye Baek, Jae-Soo Lim

KAKAO Corp

요 약

본 논문에서는 21세기 세종계획 “현대문어 형태 분석 말뭉치"에서 나타나는 오류를 개선하는 방법으로 패치 시스템을 제안한다. 이 패치 시스템은 패치 파일과 패치 적용-생성 스크립트로 구성되며, 사용자들은 패치 파일을 사용하여 원래의 말뭉치에서 어떤 파일과 어절을 수정하였는지 확인할 수 있어 개발 목적에 맞는 학습 말뭉치를 생성할 수 있다. 또한 이 시스템을 이용해 서로의 수정 사항을 공유하고, 지속적으로 세종 말뭉치의 오류를 개선할 수 있다. 본 논문에서는 총 1,015만 어절을 대상으로 31만여 개의 오류를 수정하였다. 오류의 유형으로는 문장, 어절 분리 오류, 철자 오류, 불일치 오류, 분석 오류, 형식 오류가 있으며, 오류 수정 사항을 패치 파일에 반영하였다.

주제어: 세종 형태 분석 말뭉치, 오류 수정, 공개 및 협업

1. 서론

자연언어처리 분야에서 세종 형태 분석 말뭉치는 형태소 분석기나 품사 태거를 개발하는 데 활용된다. 그러나 세종 형태 분석 말뭉치 자체에는 철자 오류, 분석 오류, 형식 오류 등이 포함되어 있어 원래의 말뭉치 그대로를 학습 말뭉치로 사용하는 데 어려움이 있다. 따라서 대부분의 연구에서는 세종 형태 분석 말뭉치를 학습 말뭉치로 활용하기 위해 1차적으로 말뭉치의 오류를 수정하는 작업을 수행한다.

그러나 위 연구들의 결과물이 공개되어 있지 않아 수정된 말뭉치를 활용하는 것이 쉽지 않다. 또한 공개된 말뭉치라도 원래의 말뭉치에서 어떠한 부분이 수정되었는지 파악하기 어려워 개발 목적에 따라 반영하고 싶지 않은 수정 사항이 있을 경우, 2차적으로 그러한 부분을 일일이 확인하고 수정하는 데 오랜 시간이 걸린다.

본 논문에서는 위와 같은 어려움을 개선하기 위해 패치 시스템에 기반한 말뭉치의 오류 수정 방법을 제안한다. 사용자들은 이 시스템으로 누구나 오류가 수정된 세종 형태 분석 말뭉치를 생성할 수 있고, 말뭉치에서 어떤 파일과 어절을 수정하였는지 확인할 수 있어 개발 목적에 맞는 학습 말뭉치를 생성할 수 있다. 또한 서로의 수정 사항을 공유해 지속적으로 세종 말뭉치의 오류를 개선할 수 있다.

본 논문의 구성은 다음과 같다. 2장에서는 세종 말뭉치의 오류 수정과 관련된 연구와 오류를 수정한 대용량 말뭉치에 관해 살펴본다. 3장에서는 패치 시스템에 기반한 오류 수정 방법을 제안하고, 4장에서는 3장에서 제안한 방법을 적용한 결과를 제시한다. 5장에서는 본 논문의 결과와 앞으로의 연구 방향에 대해 설명하고 논문을 마친다.

2. 관련 연구

세종 말뭉치에서 나타나는 오류들을 검출하고 수정하기 위해서 다양한 연구들이 선행되었다. [1]에서는 원어절과 형태 분석 결과를 자모 단위로 분리하고 서로의 대응관계를 비교하여 오류를 수정하였다. [2]에서는 형태소 분석 결과에 포함된 형태소들을 결합하여 이를 원어절과 비교하는 방식으로 오류를 검출하였고, 이렇게 검출된 오류들을 수정할 수 있는 도구를 개발하였다. [3]에서는 세종 말뭉치로 학습한 품사 태거의 결과와 정답 결과를 비교해 오류 어절을 추출하고, 고빈도로 나타난 어절을 우선적으로 수정하여, 약 15만여 개의 오류를 수정하였다. [4]에서는 1500만 어절 규모의 세종 형태 의미 말뭉치를 어휘 분석 말뭉치로 재가공하는 과정에서 세종 지침을 수정하고, 오류 유형별로 검출 도구를 이용하여 오류를 수정하였다. [5]에서는 학습 데이터로 사용하는 말뭉치의 신뢰도를 검증하기 위해 오류 유형을 분류하고, 각 오류 유형별 검증 방법을 제안하였다.

위와 같이 세종 말뭉치의 오류 수정과 관련된 연구들은 세종 말뭉치의 오류 유형을 정의하고, 이러한 오류를 효율적으로 검출할 수 있는 오류 검출 도구나 오류 수정 도구 개발에 관한 것이다.

그러나 수정된 말뭉치의 활용 측면을 고려해 볼 때 이러한 연구들의 연구 결과물들은 학습 말뭉치로 사용하기에 어려움이 있다. 수정된 말뭉치가 공개되어 있지 않거나, 공개가 되어 있더라도 어떤 어절이 수정되었는지 수정된 내용을 파악하기 힘들기 때문이다.

또한, 대용량의 세종 형태 분석 말뭉치를 대상으로 오

류를 수정한 창원대 말뭉치, 고려대 말뭉치(SJ-RIKS), 울산대 말뭉치(UCorpus-HG)에서는 기존의 세종 말뭉치 지침을 수정하거나 원문 자체를 수정하였기 때문에 수정된 말뭉치 그대로 사용하기 어렵다.

구체적으로 창원대 말뭉치에서는 접두사는 인정하지 않고, 접두사와 후행 명사가 결합한 형태를 일반명사로 분석하였고, 명사와 명사로 이루어진 어절은 하나의 명사로 분석하였다[3]. 예를 들어서, '응시자격을' 이라는 어절이 있다면, 세종 말뭉치에서는 '응시/NNG + 자격/NNG + 을/JKO'로 명사와 명사의 결합을 분리하여 분석하지만, 창원대 말뭉치에서는 '응시자격/NNG + 을/JKO' 와 같이 명사와 명사의 결합을 단일 명사로 분석한다.

울산대 말뭉치에서는 원어절에서 철자 오류가 발생한 경우, 원본 데이터를 수정하거나 형태 중심이 아닌 표준국어대사전에 등재된 어휘 중심으로 분석하였다. 예를 들어서 '생산적' 이라는 어절을 형태 중심으로 분석한다면, '생산/NNG + 적/XSN'으로 분석해야 하지만, 울산대 말뭉치에서는 표준국어대사전 표제어를 기준으로 해당 어절을 단일어인 '생산적/NNG'로 분석하였다.

고려대 말뭉치에서도 어휘 중심의 정보를 추출할 수 있는 말뭉치로 재가공하였기 때문에 접두사를 인정하지 않았고, 접미사도 세종 말뭉치에서 지정한 54개의 목록과 달리 16개의 형태를 제외하고 선행하는 명사와 통합하여 분석하였다[4].

창원대, 고려대, 울산대 말뭉치와 같이 어휘 중심으로 분석한 말뭉치로 학습한 형태소 분석기나 품사 태거는 정보 검색에 사용할 경우, 재현율이 떨어질 가능성이 있다.

따라서 본 논문에서는 원래의 세종 지침에 따라 형태 중심의 분석을 따르면서 지침에 벗어나는 오류들을 수정하고, 원래의 코퍼스에서 어떤 부분이 수정되었는지 수정된 내용을 파악할 수 있는 패치 시스템을 제안한다.

3.제안 방법

본 논문에서 제안하는 패치 시스템은 패치 파일과 패치 적용 및 생성 스크립트로 구성된다.

그림1과 같이 패치 시스템은 동일한 세종 형태 분석 말뭉치 원본을 가지고 있는 사용자와 협업할 수 있으며, 협업은 커미터(committer)가 배포한 패치 적용 스크립트, 패치 생성 스크립트, 패치 파일을 이용해 이루어진다.

사용자1처럼 세종 말뭉치의 원본을 가지고 있는 경우, 커미터가 배포한 패치 적용 스크립트와 패치 파일을 이용해, 오류가 수정된 세종 말뭉치 수정본을 생성할 수 있다. 또한 수정된 말뭉치에서 더 수정하고 싶은 부분이 있다면, 패치 파일을 직접 수정하여 새로운 세종 말뭉치 수정본을 생성할 수 있다.

사용자2와 같이 세종 말뭉치의 원본과 세종 말뭉치 수정본을 가지고 있는 경우, 커미터가 배포한 패치 생성 스크립트를 이용해 패치 파일을 생성할 수 있다. 사용자2는 이러한 패치 파일과 커미터가 배포한 패치 파일을 비교하여, 자신이 반영하지 못한 오류 사항이 있다면 자신의 패치 파일을 보완할 수 있다. 반대로 커미터가 배포한 패치 파일에 반영되지 못한 수정 사항이 있다면 커미터에게 피드백을 줘 배포용 패치 파일을 보완할 수 있다.

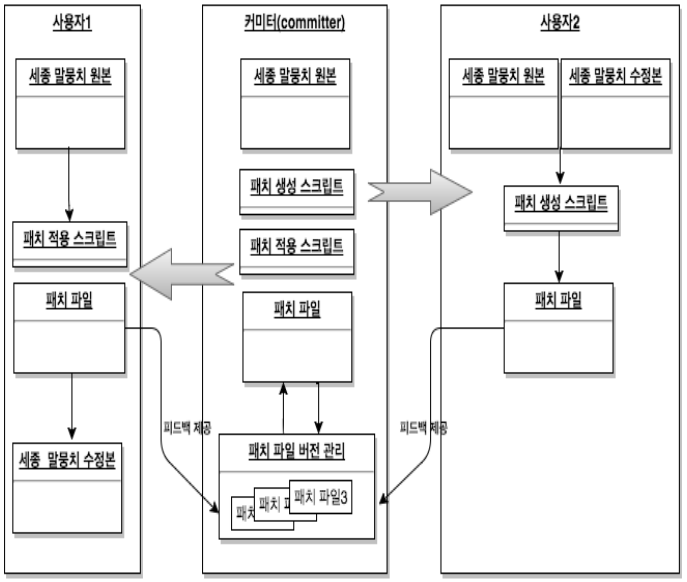


그림1. 패치 시스템을 이용한 협업 과정

이와 같이 패치 파일을 매개로, 패치 파일을 배포하는 커미터와 패치 파일을 사용하는 사용자가 서로 협업하여 지속적으로 세종 코퍼스의 오류를 개선할 수 있다.

패치 파일			
=	BTHO0397-00058767	소년들에서	소년/NNG + 들/XSN + 에서/JKB
+	BTAA0011-00012666-1	보리는	보리/NNG + 는/JX
-	BTBZ0074-00028385		
M	BTGO0348-00013286	BTGO0348-00013287	
S	BTAE0200-00042670	BTAE0200-00042671	

그림2. 패치 파일 예

패치 파일에는 세종 말뭉치에서 수정한 어절의 정보를 담고 있다. 패치 파일은 그림2와 같이 패치 종류, 어절 식별 번호, 원어절, 형태소 분석 결과로 구성되며 패치 종류는 아래와 같이 총 5개이다.

(1) 어절 변경 및 삭제/추가
- = : 해당 어절을 다음과 같이 변경
- + : 해당 어절 추가
- - : 해당 어절 삭제

(2) 문장 분리 오류
- M : 두 어절 사이 문장 분리 표지를 삭제하고 하나의 문장으로 합침(merge)
- S : 두 어절 사이에서 문장 분리 표지를 추가하여 두 문장으로 분리(split)

어절 관련 패치는 오류로 인해 어절을 변경할 경우, '=' 패치가 적용되며, 어절을 추가하거나 삭제할 때는 '+' 패치나 '-' 패치가 적용된다.

문장 분리 관련 패치는 문장 종결(예:)과 시작 마커(예:)로 잘못 분리된 어절들을 병합하는 'M, 문장 종결과 시작 마커로 분리되지 못한 두 어절을 분리하는 'S' 로 구성된다.

본 논문에서 구축한 패치 파일에는 이와 같은 패치의 종류뿐만 아니라, 세종 형태 분석 말뭉치와 동일한 어절 식별 번호를 함께 기술하기 때문에 사용자들은 어떠한 파일의 어절이 변경되었거나 삭제, 추가되었는지 확인할 수 있다.

4. 적용결과

4.1 오류 유형 정의

본 논문은 21세기 세종계획의 "현대문어 형태분석 말뭉치"를 대상으로 오류를 검출하고 수정을 진행하였다. 말뭉치의 오류 수정은 세종 지침에 따라 형태 중심의 분석을 따르면서 지침에 벗어나는 오류들을 수정하고 가급적 원문을 수정하지 않았다.

원문 자체를 수정한 경우는 의미를 파악할 수 없어 분석이 불가능하거나 오탈자 때문에 의미 없는 문장이 되었을 때이다.

오류의 검출과 수정은 1, 2차로 나누어 진행되었다. 형태 분석 말뭉치의 오류를 발견하고 수정하기 위해 모든 어절을 검증하는 것은 현실적으로 불가능하므로 오류 검출을 위해 1차적으로 세종 코퍼스와 창원대, 울산대 코퍼스를 각각 비교하여 고빈도로 나타나는 수정 사항을 참고하여 오류를 검출하였다. 그리고 2차적으로 원어절과 형태소 분석 어절의 음절 대응관계를 비교하여 오류를 검출하고 수정하였다.

현재까지 이루어진 형태 분석 말뭉치에서 수정한 오류를 유형별로 분류하면 다음과 같다.

(1) 문장 및 어절 분리 오류

하나의 문장임에도 여러 개의 문장으로 분리된 오류, 각각 다른 문장임에도 문장 분리가 이루어지지 않은 오류 유형이 해당한다.

예) 집회(?)를 집회/NNG + (/SS + ?/SF +)/SS +
를/JKO

이루고 이루/VV + 고/EC

(2) 철자 오류

원어절 및 형태 분석 결과에 나타나는 철자 오류이다. 원어절에 나타난 철자 오류의 경우, 사람들이 빈번하게 사용하는 오타(예: 요드, 횡경막)는 수정하지 않고, 코드 변환 과정에서 발생한 오류나, 분석이 불가능하거나 의미 없는 어절이 된 경우에 한해 수정하였다.

예) 밀?수입 밀/NNG + 수입/NNG

형태 분석 결과에 철자 오류가 나타난 경우, 오타는 원어절에 따라 분석 결과를 수정하였다.

예) 웬놈의 왠/MM + 놈/NNB + 의/JKG

(3) 불일치 오류

원어절과 형태 분석 결과의 음절 대응 관계를 비교하였을 때, 원어절에는 나타난 형태지만 형태 분석 결과에는 누락되었거나, 원어절에 나타나지 않은 형태가 형태 분석 결과에 추가되어 나타나는 오류이다.

예) 것"이라며 것/NNB + "/SS + 이/VCP + 며/EC
도쿄의 도쿄/NNP + (/SS + 동경/NNP +)/SS + 의/JKG

(4) 분석 오류

원어절을 엉뚱하게 잘못 분석한 경우로, 분석 표지를 잘못 부착하거나, 하나의 형태소로 분석해야 하는데 잘못 분할한 오류 유형이다.

예) 홍걸씨 홍/NNP + 것/NNB + 이/VCP + ㄹ/ETM + 씨/NNB
소년들에서 소년/NNG + 이/VCP + ㄴ들/EC + 에서/JKB

(5) 형식 오류

세종 형태 분석 말뭉치에서 제시하는 분석 형식을 준수하지 않은 오류로, 형태 분석 결과를 기술하는 과정에서 '+' 또는 '/' 를 누락하였거나, 태그를 이중으로 기술하는 오류 유형이 해당한다.

예) 국제자유도시로 국제/NNG + 자유/NNG + 도시/NNG 로/JKB
청계고가도로 청계/NNP+고가/NNG+도로/NNG/NNG

4.2 패치 적용 결과

본 논문에서는 총 1,015만여 개의 어절을 대상으로, 약 31만여 개의 오류를 수정하였고, 패치 적용 결과는 표1과 같다.

'M' 패치와 'S' 패치는 문장 및 어절 분리 오류에 적용된다. 'M' 패치는 하나의 문장으로 묶여야 할 어절이 다른 문장으로 분리된 오류에 적용된다. 표1의 예처럼 한 문장 내에서 목적어-술어 관계를 이루는 어절(예: 곤욕을 치렀다)들이 문장 시작 (예:) 및 종결 마커(예:)로 잘못 분리된 오류가 있다. 이러한 오류는 'M'

패치 적용 후, 하나의 문장으로 수정된다.

표1. 패치 적용 결과

패치 종류	세종 말뭉치	수정된 말뭉치
M	BTAB0170-00002897 곤욕(?)을 곤욕 /NNG + (/SS + ?/SF +)/SS + 을/JKO BTAB0170-00002898 치렀다. 치르/VV + 었 /EP + 다/EF + ./SF	BTAB0170-00002897 곤욕(?)을 곤욕 /NNG + (/SS + ?/SF +)/SS + 을/JKO BTAB0170-00002898 치렀다. 치르/VV + 었 /EP + 다/EF + ./SF
S	BTBD0236-00089764 오세요 오/VV + 시/EP + 어요/EC BTBD0236-00089765 !양경숙 !/SF + 양경숙 /NNP	BTBD0236-00089764 오세요! 오/VV + 시/EP + 어요/EC + !/SF BTBD0236-00089765 양경숙 양경숙/NNP
+		BTAA0155-00051766-1 지난주 지난주/NNG
-	BTBZ0074-00028385 는 늘/VV + ㄴ /ETM	
=	BTGO0345-00002470 가사로부터 가사 /NNG + 부터/JX	BTGO0345-00002470 가사로부터 가사 /NNG + 로부터/JKB

이와 반대로 ‘S’ 패치는 서로 다른 문장으로 분리되어야 하는 어절임에도 하나의 문장으로 분석된 오류에 적용된다. 적용 후에는 표1처럼 두 어절 사이에 문장 분리 표지가 추가되어 각각의 문장으로 분리된다.

‘+’ 패치와 ‘-’ 패치는 오류로 인해 해당 어절을 추가하거나 삭제한 경우에 적용된다.

‘+’ 패치는 세종 형태 분석 말뭉치에 없던 새로운 어절이 생성되었을 때 적용되는데 새로운 어절은 문장 및 어절 분리 오류를 수정하는 과정에서 생성된다. 예를 들어서, 문장 및 어절 분리 오류가 나타난 다어절은 각각의 어절로 분리되는 과정에서 새로 어절이 생성되기 때문에 ‘+’ 패치가 적용된다. 이러한 어절은 표 1의 예처럼 ‘-1’ 이라는 새로운 어절 번호를 할당받는다.

‘-’ 패치는 단독 어절로 쓰일 수 없는 형태가 단독 어절로 나타난 경우, 그 어절을 삭제할 때 적용된다. 표1의 예처럼, ‘-’ 패치가 적용된 ‘는’ 은 원문을 살펴보면, 어미 ‘라는'의 일부이다. 따라서 ‘-’ 패치 적용 결과, ‘는’ 이 단독 어절로 나타난 어절은 삭제되고, ‘는’ 은 선행 어절에 결합되어 분석된다.

‘=’ 패치는 철자 오류, 불일치 오류, 분석 오류, 형식 오류에 적용된다. 표1의 예는 원어절을 구성하는 형태소가 형태 분석 결과에 누락된 오류로, ‘=’ 패치 적용 결과 누락된 형태소가 형태소 분석 결과에 생성되었다.

5. 결론

본 논문에서는 가급적 원문을 훼손하지 않고, 원래의 세종 지침에 따라 형태 중심의 분석을 따르면서 지침을 벗어나는 오류들을 수정하였다. 수정된 오류 유형으로는 문장 및 어절 분리 오류, 철자 오류, 불일치 오류, 분석 오류, 형식 오류가 있었다.

수정된 내용이 반영된 패치 파일을 패치 적용 및 생성 스크립트와 함께 배포하여, 패치 파일을 배포하는 커미터와 사용자가 서로 협업하여 지속적으로 세종 말뭉치의 오류를 개선할 수 있는 방법을 제안하였다.

향후에는 동일한 어절을 일관성 없게 분석한 오류 유형을 수정할 것이며, 말뭉치 대상을 확대해 문어체 말뭉치뿐 아니라 구어체 말뭉치에 대해서도 지속적으로 오류를 검증하고 수정해 나갈 것이다.

참고문헌

[1] 김재훈, 서형원, 전길호, 최명길, “세종말뭉치의 오류 수정 방법", 한국마린엔지니어링학회 학술대

회 논문집, pp.435-436, 2010.
[2] 최명길, 서형원, 권홍석, 김재훈,“한국어 품사 부착 말뭉치의 오류 검출 및 수정", 한국마린엔지니어링학회지, 제 37 권, 제 2 호, pp.227-235, 2013.
[3] 홍진표, 차정원,“품사 태거와 빈도 정보를 활용한 세종 형태 분석 말뭉치 오류 수정”, 정보과학회논문지: 소프트웨어 및 응용, 제 40 권, 제 7 호. pp.417-428, 2013.
[4] 김일환, 이도길, 강범모, “SJ-RIKS Corpus : 세종 형태의미 분석 코퍼스를 넘어서”, 민족문화연구, 제 52 권, pp.373-403, 2010.
[5] 이미경, 정한민, 성원경, 박동인. “품사 표지 부착 말뭉치 검증", 한국정보과학회 언어공학연구회 학술발표 논문집, pp.145-150, 2005.

Sequence-to-Sequence 모델을 이용한 신문기사의 감성 댓글 자동 생성*

박천용[0], 박요한, 정혜지, 김지원, 최용석, 이공주
충남대학교
sdpcy0520@gmail.com, happy005012@naver.com, hyeji6138@naver.com,
top9076@cnu.ac.kr, yseokchoi@cnu.ac.kr, kjoolee@cnu.ac.kr

Automatic Generation of Emotional Comments on News-Articles using Sequence-to-Sequence Model

Chun-Young Park[0], Yo-Han Park, Hye-Ji Jeong, Ji-Won Kim, Yong-Seok Choi, Kong-Joo Lee
Chungnam National University

요 약

본 논문은 신문기사의 감성 댓글을 생성하기 위한 시스템을 제시한다. 감성을 고려한 댓글 생성을 위해 기존의 Sequence-to-Sequence 모델을 사용하여 긍정, 부정, 비속어 포함, 비속어 미포함 유형의 4개의 감성 모델을 구축한다. 하나의 신문 기사에는 다양한 댓글이 달려있지만 감성 사전과 비속어 사전을 활용하여 하나의 댓글만 선별하여 사용한다. 분류한 댓글을 통해 4개의 모델을 학습하고 감성 유형에 맞는 댓글을 생성한다.

주제어: 감성, 댓글, Sequence-to-sequence

1. 서론

딥 러닝 기법이 발달하면서 자연어 처리 분야 또한 눈에 띄게 많은 발전을 이루었다. 그 중, Sequence-to-Sequence(seq2seq)모델은 기계번역 분야에서 뛰어난 성능을 보이고 있다[1]. 그밖에도 QA 시스템에서 질문에 대한 응답, 문서에 대한 질문, 신문기사의 제목, 프로그램 코드에 대한 주석을 생성하는 데에도 다양하게 활용된다[2-6]. 영어권에서는 seq2seq 모델을 이용하여 감성에 따른 응답을 생성하는 챗봇[7]에 관한 연구가 있었으나 국내에서는 아직 감성을 고려한 생성에 관한 연구가 많지 않았다. 본 논문에서는 seq2seq모델을 이용하여 감성이 표현된 문장을 생성해 보고자 한다.

모델의 학습 데이터로는 인터넷 기사와 댓글을 수집하여 사용한다. 왜냐하면 댓글은 “좋다”, “싫다”, “잘했다”, “못했다” 등과 같이 글쓴이의 감성을 포함하고 있는 경우가 많기 때문이다. 또한 기사와 댓글의 양이 많고 수집도 용이하기 때문에 대량의 데이터 수집도 가능하다. 자동으로 수집한 댓글을 감성사전을 이용하여 긍정이나 부정의 의미를 가진 댓글로 분류한다. 또한 비속어 사전을 이용하여 비속어 포함 댓글과 그렇지 않은 댓글로도 분류하여 모델의 학습 데이터로 사용해 본다.

본 연구에서는 긍정, 부정, 비속어 포함, 비속어 미포함, 4개의 모델을 각각 구축하여 감성이 표현되어 있는 댓글을 자동 생성해 보고자 한다.

* 이 논문은 2016년도 정부(미래창조과학부)의 재원으로 한국연구재단의 지원을 받아 수행된 기초연구사업임(No.NRF-2015R1C1A2A01051685)

2. 댓글 자동 생성

2.1 전체 시스템

감성을 고려한 댓글을 자동 생성하기 위해 그림 1과 같이 네 가지 감성 유형의 모델을 사용한다. 각 모델의 입력은 기사 제목과 첫 문장이며, 출력은 입력을 토대로 감성 유형에 맞게 생성된 댓글이다.

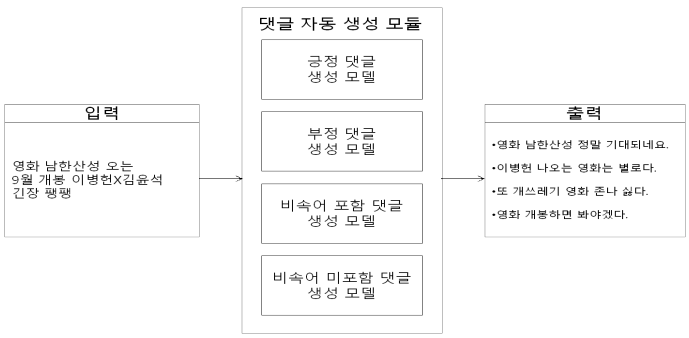


그림 1. 감성을 고려한 댓글 생성 모델

2.2 Seq2seq 모델

본 연구에서는 댓글 생성을 위해 seq2seq 모델을 사용한다. 그림 2는 입력단에 “특이한 고양이 있다”가 들어갔을 때 “참 좋은 소식이네요”라는 댓글을 생성하는 예시이다. 인코더의 입력은 형태소의 임베딩 벡터를 사용한다. 그리고 인코더의 출력은 입력 문장의 의미가 포함된 벡터이다. 디코더는 인코더가 출력한 벡터와 Bahdanau[8]의 주의집중(attention) 알고리즘을 바탕으로 계산한 가중치를 입력받는다. 디코더의 출력은 Softmax 함수를 사용하여 가장 높은 확률의 단어를 선택한다.

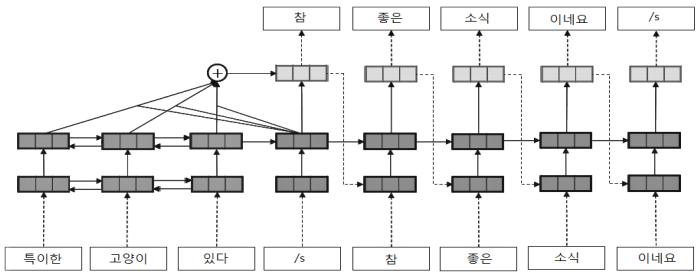


그림 2. Sequence-to-sequence Model

2.3 데이터 수집 및 처리

seq2seq모델을 이용하기 위해서는 대량의 데이터가 필요하다. 따라서 다음(Daum) 신문기사[2]를 수집하였다. 신문기사는 독자의 이해를 돕기 위해 대체적으로 두괄식 구조를 채택하고 있다[9]. 그러므로 본 연구에서는 신문기사의 전문이 아닌 제목과 첫 문장만을 모델의 입력으로 사용한다. 표 1은 수집한 데이터의 개수와 평균 어절 수이다.

표 1. 분야별 신문기사의 수

단위 : 개

분야	기사	댓글	기사[1] 평균 어절	댓글 평균 어절
it	6,079	137,544	28.78	13.16
문화	8,129	282,506	24.72	13.32
연예	43,928	675,755	23.64	9.78
스포츠	13,326	2,192,384	22.35	13.47
경제	6,848	273,429	30.03	14.80
사회	12,126	756,754	30.81	12.62
기타	49,517	12,233,498	31.41	14.47
총합	139,953	16,551,870	27.39	13.08

본 논문은 감성사전[10]과 비속어사전[10]을 이용하여 댓글을 긍정, 부정, 비속어 포함, 비속어 미포함 댓글로 분류하였다. 감성사전에서 1,647개의 긍정 표현과 4,573개의 부정 표현을 사용하였고 비속어 사전에서 409개의 비속어 단어를 사용하였다. 표 2는 긍정, 부정, 비속어 단어들의 예시이다.

표 2. 유형별 단어 예시

유형	예시
긍정	가슴 뛰리는 기분, 딱 좋다, 감동 받다, 걱정 말아요, 가치 있다, 너무 보기 좋다, 매우 좋다, 보기 좋다
부정	거부감 들다, 골치 아프다, 너무 후회, 답답해 보이다, 미심쩍다, 분노하다, 마음에 안 들다, 눈꼴사납다
비속어	시발, 병신, 놈, 전나, 쪽발이, 개새끼, 지랄, 쥐뿔

하나의 신문기사에는 평균 118개 이상의 댓글이 달려있다. 본 연구에서는 각 유형에 맞게 선별된 댓글과 그 댓글이 달린 신문기사를 학습데이터로 사용한다.

댓글을 선별할 때에는 감성사전과 비속어 사전에서 추출한 단어가 댓글에 포함되어 있는지 확인한다. 그 중 댓글의 어절 수가 14 어절 이하이고, 기사의 제목과 첫 줄에서 사용된 단어가 가장 많이 포함된 댓글을 최종적으로 선택한다. 그 후 문장의 모든 특수기호를 제거하고, 숫자는 "<NUM>"로, 공백은 "<SP>"로 정규화 한다. twitter-korean-text[11]를 이용해 형태소 분석을 하고 입력단의 조사만 제거한다. 표 3은 이와 같이 추출한 유형별 댓글 수이다. 각 유형별로 선별된 댓글 중 100개는 평가 데이터이고 그 외의 댓글은 모두 학습데이터이다.

표 3. 유형별 선별된 댓글 수

단위 : 개

유형	선별된 댓글 수
긍정	51,472
부정	64,240
비속어 포함	70,043
비속어 미포함	101,383

3. 실험 환경 및 결과

3.1 실험환경

모델을 학습할 때 사용한 파라미터는 표 4와 같다. 네 가지의 모델은 동일한 파라미터를 사용하였다. seq2seq 모델은 OpenNMT[3]를 활용하여 학습 시킨다.

1) 기사의 평균어절 수는 제목+첫 문장의 평균 어절 수이다.
2) http://media.daum.net
3) http://opennmt.net/

표 6. 평가항목-2 기준

유형	입력 (제목+신문기사)	출력 (댓글)
입력 문장의 키워드가 출력 문장에 포함된 경우	고개 숙인 **종근당** 회장 운전기사 폭언 ...	**종근당** 불매운동 해야 한다
입력 문장과 관련된 어휘가 출력된 경우	타격의 팀 기아 타이거즈와 **SK 와이번스**가 역대급 명승부를 펼쳤다	**김성근 선수** 수고 많았습니다 내년엔 좋은 결과 있길 바랍니다
문맥상 의미가 적합한 경우	300mm 폭우 피해 눈덩이 **사망** 4명 **실종** 2명 이재민 517명 ...	삼가 **고인**의 **명복**을 빕니다 잊지 않겠습니다
두 문장의 관련성이 없는 경우	호텔에 등 달고 변기 뜯고 천상천하 유아독존	서세원이 더 나쁘다는 말이 아니라 이게 무슨 의미가 있다는 것 자체가 한심하다

표 4. 모델에 사용한 파라미터

Parameter	Value
Word embedding size	256
RNN size	512
Encoder/Decoder layer	2
Encoder/Decoder type	GRU
Dropout	0.3
Optimizer	Adam
Learning rate	0.001
Epoch	200

3.2 실험 평가 기준

seq2seq 모델을 이용한 대부분의 생성 결과는 정답과 동일한 표현이 얼마나 발생했는지를 평가의 기준으로 사용한다[1,4,12]. 그러나 본 연구의 경우, 정답으로 간주할 수 있는 댓글이 각 신문기사 당 118개 이상이기 때문에 정답 댓글과의 정량적인 비교가 어렵다. 그러므로 정량적인 평가보다는 정성적인 평가를 수행해 보고자 한다.

각 모델의 대한 평가는 다음의 3가지 기준을 사용한다. 평가 항목-1은 출력된 문장의 완성도와 의미를 평가한다. 댓글의 경우 표 5와 같이 여러 유형의 문장이 존재하기 때문에 이를 고려하여 평가한다. 문장이 온전하며 의미전달이 명확할 때 최고점을 부여한다. 문장이 이루어지지 않거나 명사로 끝나는 경우라도 의미가 전달된다면 높은 점수를 부여한다. 의미가 명확히 전달되지 않으면 낮은 점수를 부여한다.

표 5. 인터넷 기사의 댓글 유형

유형	예시
문장이 온전한 경우	쉬고 싶은 숲 너무 보기 좋습니다.
문장이 이루어지지 않는 경우	목숨 걸고 갈 만큼 가치 있는지
문장이 명사로 끝나는 경우	읽어보고 왜 인기 있는지 이해가 안 되었던 책

평가항목-2는 입력 문장과 출력 문장 사이의 의미 관련성을 평가한다. 입력 문장의 주요 키워드가 출력 문장에 포함되어 있으면 높은 점수로 평가한다. 그 외에 관련 어휘가 포함되었을 경우와 문맥상의 의미가 적합할 경우에는 그보다 낮은 점수로 평가한다. 입력문장과 출력문장이 연관성이 없을 경우 낮은 점수로 평가한다. 평가예시는 표 6와 같다.

평가항목-3은 출력된 문장이 감성유형에 적합한지를 평가한다. 긍정 댓글 생성기의 경우 출력된 문장이 긍정 표현에 가까운 지를 평가한다. 부정 댓글 생성기의 경우 출력 문장이 부정표현에 가까운 지를 평가한다. 비속어 포함 모델 및 비속어 미포함 모델의 경우는 평가항목-3은 평가하지 않았다.

평가의 공정성을 확보하기 위해 한 모델 당 평가자 2명씩 총 8명이 평가를 실시하였다. 평가할 댓글은 100개의 신문기사에 대해서 모델이 생성한 댓글과 실제 기사의 댓글을 섞어 총 200개의 댓글을 함께 평가한다. 각 항목의 점수는 1-5점 범위이다. 평가자는 어떤 결과가 실제 사람이 작성한 댓글인지 시스템이 자동 생성한 댓글인지 모른 채 평가를 수행하도록 하였다.

3.3 실험 결과 및 분석

표 7은 실제 인터넷 기사 댓글, 표 8은 댓글 자동 생성 모듈이 생성한 댓글에 대한 각 평가항목별 평균점수이다. 예상했던 것과 같이 실제 인터넷 기사 댓글이 거의 모든 항목에서 좋은 점수를 받았다.

감성적합성의 부문에서는 모델이 생성한 댓글의 점수가 더 높았다. 부정 유형의 경우, 모델이 생성한 댓글이 “짜증난다.”, “개판이다.” 등 의미가 명확한 단어가 인터넷 기사의 댓글에 비해 많이 나타났기 때문인 것으로 판단된다. 긍정 유형의 경우 실제 인터넷 기사의 댓글에서

표 9. 각 모델의 좋은 예와 나쁜 예

		유형	입력 (제목+신문기사)	출력 (댓글)
좋은 예	(1)	긍정	TV 예능 프로그램 동상이몽 통해 기존 샤프 이미지 벗고 부드러운 면서도 속정 깊은 아저씨로 새롭게 인식되고 있는 이재명 성남시장 침체 된 한국 관광 활성화 나섰다	우와 좋은 모습 보기 좋아요
	(2)	부정	이철성 음주 사고 은폐 민정 수석실 알고도 넘어갔다 청와대 민정 수석실 이철성 경찰청장 후보자 과거 음주운전 사고 당시 신분 속여 징계 받지 않았다는 사실 알고도 이를 문제 삼지 않은 것으로 드러났다	진짜 짜증난다 진짜 나라꼴이 개판이다 이 나라의 미래는 암울하다
	(3)	비속어 포함	박주영 리더 대신 그림자를 택하다 홍명보호의 맏형 격인 박주영이 리더 대신 조력자 역할을 선택했다	홍명보 같은 놈들은 뭐냐 양심도 없는 놈들이네
	(4)	비속어 미포함	귀국한 어보 박수치는 문 대통령 내외 배재만 기자 미국 방문을 마친 문재인 대통령이 2일 오후 서울공항에 도착해 함께 전용기편으로 들어온 조선현종어보 문정왕후 어보를 보며 박수치고 있다	문재인 대통령님 감사드립니다
나쁜 예	(5)	긍정	옥에 티 얼룩진 흰 운동화 하얗게 만들려면 김대리가 생활 속 꿀팁을 전합니다	저런 사람들이 살아있는 사람들은 알고 있는거 아닌가 뭔 의미가 있나
	(6)	부정	관광버스 참사 속에서 가장 먼저 탈출한 운전기사 지난 13일 경부고속도로 언양 분기점 인근을 달리던 전세버스에 화재 사고가 발생한 직후 운전기사 이 모씨가 가장 먼저 버스에서 탈출했다는 경찰 조사 결과가 나왔다	제목 좀 내려라 너무 비싸다 좀 내려라
	(7)	비속어 포함	MLB 류현진 다음등판 장담 못해 발톱 생각보다 나빠 호주 개막시리즈에서 기분 좋은 칫승을 신고했던 류현진이 다음 이어지는 본토 개막시리즈 등판을 장담할 수 없게 됐다	기자놈아 전범기가 아니라 전범기가 아니라 전범기가 아니라 전범기가 아니라 쪽바리 색히들아
	(8)	비속어 미포함	김상곤 송영무 엄호 조대엽은 떨떠름 낙마사유는 없어 강병철 한지훈 기자 더불어민주당은 2일 김상곤 송영무 후보자를 엄호하면서 조대엽 후보자에 대해서는 여론을 주시했다	노회찬 의원님 존경합니다 고인의 명복을 빕니다

표 7. 실제 인터넷 기사 댓글

	평가항목-1 문장완성도	평가항목-2 의미관련성	평가항목-3 감성적합성
긍정	3.78	3.945	3.02
부정	4.82	3.88	4.13
비속어 포함	4.64	4.205	-
비속어 미포함	4.58	4.25	-

표 8. 감성 모델이 생성한 댓글

	평가항목-1 문장완성도	평가항목-2 의미관련성	평가항목-3 감성적합성
긍정	3.37	2.205	3.585
부정	3.615	3.1	4.265
비속어 포함	3.525	3.08	-
비속어 미포함	3.3	3.85	-

긍정적인 단어가 나타나더라도 반어적인 의미로 사용된 경우가 있기 때문이다.

표 9는 각 모델이 생성한 문장 중 좋은 예와 나쁜 예이다. (5)와 (6)의 경우 입력과 관련이 없는 댓글이 생성되었다. (7)의 경우 seq2seq의 반복 생성 문제를 살펴볼 수 있었다. (8)의 예에서는 같은 분야지만 입력에서 거론되지 않은 이름이 댓글에서 언급된 것을 살펴볼 수 있다.

각 모델의 출력에서 사용된 단어와 실제 댓글에서 사용된 고유 단어 개수와 평균 어절 수는 표 10과 같다. 출력 결과의 고유한 단어 개수가 실제 댓글의 단어 개수보다 적다. 본 연구에서는 다양한 표현을 생성할 수 있기를 기대하였으나 실제 출력에서는 단어의 사용이 한정적인 것을 확인하였다.

표 10. 댓글에 쓰인 고유 단어 개수와 평균 어절 수

	실제 댓글		모델이 생성한 댓글	
	단어 수	평균 어절	단어 수	평균 어절
긍정	941	8.49	287	6.67
부정	1,014	9.08	321	8.57
비속어	1,041	9.05	285	7.93
비속어 미포함	1,007	8.9	375	9.76

4. 결론

본 논문에서는 감성을 고려한 댓글을 자동으로 생성하였다. 이를 위해 신문 기사를 수집하고 감성 사전과 비속어 사전을 토대로 댓글을 분류하였다. seq2seq 모델을 이용하여 긍정, 부정, 비속어 포함, 비속어 미포함 모델을 구성하였다. 실험의 결과로 기사 내용에 따라 감성이 잘 나타나는 문장 생성이 가능하다. 그러나 모델이 자동 생성한 댓글은 실제 인터넷 기사의 댓글에 비해 맞춤법과 띄어쓰기가 더 정확하였다. 실제 인터넷 기사의 댓글처럼 생성하기 위해서는 추가적인 연구가 필요하다. 또한 모델이 자동 생성한 댓글은 신문기사 내용과 의미관련성이 적은 경우가 많았고, 학습한 단어 수에 비해 다양한 표현들이 댓글에 많이 나타나지 않았다. 향후 기사 분야를 좁혀 학습시킬 경우 의미관련성이 향상될 것으로 기

대된다.

참고문헌

[1] SUTSKEVER, Ilya; VINYALS, Oriol; LE, Quoc V. Sequence to sequence learning with neural networks. In: *Advances in neural information processing systems*. 2014. p. 3104-3112.

[2] HE, Shizhu, et al. Generating Natural Answers by Incorporating Copying and Retrieving Mechanisms in Sequence-to-Sequence Learning. In: *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. 2017. p. 199-208.

[3] SERBAN, Iulian Vlad, et al. Multiresolution Recurrent Neural Networks: An Application to Dialogue Response Generation. In: *AAAI*. 2017. p. 3288-3294.

[4] YUAN, Xingdi, et al. Machine Comprehension by Text-to-Text Neural Question Generation. *arXiv preprint arXiv:1705.02012*, 2017.

[5] Lei Xu, Ziyun Wang, Ayana, Maosong Sun. Topic Sensitive Neural Headline Generation. 2016.

[6] HU, Xing, et al. CodeSum: Translate Program Language to Natural Language. *arXiv preprint arXiv:1708.01837*, 2017.

[7] ZHOU, Hao, et al. Emotional Chatting Machine: Emotional Conversation Generation with Internal and External Memory. *arXiv preprint arXiv:1704.01074*, 2017.

[8] BAHDANAU, Dzmitry; CHO, Kyunghyun; BENGIO, Yoshua. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*, 2014.

[9] 박재영. 뉴스 평가 지수 개발을 위한 신문 1 면 머리기사 분석. *한국의 뉴스 미디어*, 2006, 147-220.

[10] CHO, Young Hwan; LEE, Kong Joo. Automatic affect recognition using natural language processing techniques and manually built affect lexicon. *IEICE transactions on information and systems*, 2006, 89.12: 2964-2971.

[11] PARK, Eunjeong L.; CHO, Sungzoon. KoNLPy: Korean natural language processing in Python. In: *Proceedings of the 26th Annual Conference on Human &Cognitive Language Technology*. 2014. p. 133-36.

[12] BAHDANAU, Dzmitry; CHO, Kyunghyun; BENGIO, Yoshua. Neural machine translation by jointly learning to align and translate. arXiv preprint arXiv:1409.0473, 2014.

구글 학술 검색 기반의 질병과 바이오마커 관계 분석

오병두○, 김유섭

한림대학교, 융합소프트웨어학과
iambd822@gmail.com, yskim01@hallym.ac.kr

Relation Analysis of Disease and Biomarker based on Google Scholar

Byoung-Doo Oh○, Yu-Seop Kim
Hallym University, Convergence Software

요 약

본 논문에서는 구글 학술 검색 기반의 데이터를 이용하여 질병과 폐질환과 관련된 바이오마커 단어의 유사도를 계산하는 방법을 제안한다. 질병과 바이오마커의 유사도를 계산할 때, 각 단어의 구글 학술 검색의 검색 결과를 이용하였다. 이를 통해 폐질환 관련 바이오마커와 다른 질병간의 관계를 파악하고자 하며, 의료 전문가에게 폐질환 관련 바이오마커와 다른 질병간의 새로운 관계를 제시하고자 한다. 이러한 데이터를 이용하여 계산한 결과, Wor2Vec의 결과를 이용한 코사인 유사도의 결과와 상관 계수가 약 0.64로 상당히 높은 상관 관계를 확인할 수 있었다. 따라서 이 방법을 통해 질병과 바이오마커의 관계를 파악하고자 하였다. 또한 Word2Vec을 이용한 질병과 바이오마커 단어의 벡터 값과 단어 유사도 계산 방법의 결과를 이용한 Deep Neural Networks (DNNs) 모델을 구축하고자 하며, 이를 통해 자동적으로 유사도를 분석하고자 하였다.

주제어: 질병, 바이오마커, 단어 유사도 분석, Deep Neural Networks (DNNs)

1. 서론

바이오마커는 일반적으로 단백질이나 DNA, RNA 등을 이용해 신체 내부의 변화를 알아낼 수 있는 지표를 의미한다. 이러한 바이오마커는 일반 생물처리 과정, 질병을 유발하는 과정, 그리고 질병을 치료하기 위한 약리학의 과정을 측정하거나 평가하는 부분에도 쓰이게 된다.[1] 바이오마커를 발굴할 때에는 대부분 임상적인 실험 또는 연구를 통해 발굴하게 된다. 이러한 임상적인 실험은 다양한 단계(또는 과정)를 통해 진행되어 많은 시간과 비용을 소모하게 되며, 원하는 결과를 얻지 못할 수도 있다.

기존부터 자연어처리 기술을 이용한 단어 관계 분석이 이루어졌다. 그 중 Information Content를 통한 단어 관계 분석은 꾸준히 연구되고 있는 분야이다. Information Content는 코퍼스에서 단어의 확률을 사용하는 방법이다. 이 방법은 연구마다 다양한 코퍼스의 단어 확률을 계산하는 방법들을 제시하고 있다. 또한 워드 임베딩은 문서에서 단어의 관계를 계산하여 각 단어들을 벡터로 표현해주는 방법이다. 비슷한 단어들은 유사한 벡터로 표현되었으며, 이를 통해 비슷한 의미를 가진 단어들을 알아낼 수 있었다. 또한 워드 임베딩의 결과인 벡터를 이용하여 코사인 유사도 계산을 통해 단어의 관계를 계산할 수 있다. 이러한 방법들을 이용하여 자연어처리 분야, BioNLP 분야 등에서 좋은 성능을 보였다.[1-3]

본 논문에서는 기존의 단어 간의 유사도 측정보다 단어에 대한 구글 학술 검색의 결과에 기반한 단어 간의 유사도를 측정하는 방법을 논한다. 질병은 암을 포함한 37가지의 질병을 선정하였고, 바이오마커는 폐질환과 관련되어 있다고 알려진 27가지의 바이오마커를 선정하였다. 이를 통해, 폐질환과 관련된 바이오마커 중 폐질환이 아닌 다른 질병과의 새로운 관계를 파악하고자 한다. 따라서 의학 분야의 전문가들에게 질병과 폐질환 관련 바이오마커의 기존에 알려지지 않은 관계에 대해 제시하고자 한다. 질병과 바이오마커의 유사도 방법을 계산할 때, 워드 임베딩 방법 중 하나인 Word2Vec[4]의 결과를 통해 계산한 코사인 유사도의 결과와의 상관관계를 비교하였다. 이 때, 상관계수는 약 0.64의 결과를 얻을 수 있었다. 또한 이러한 계산 방법을 통해 나온 결과를 DNNs를 이용한 학습 모델을 구축하여 다른 질병과 폐질환 관련 바이오마커의 관계를 분석하는데 특화된 학습 모델을 만들고자 한다.

2. 관련 연구

[5]는 워드 임베딩을 이용하여 질병과 미생물의 관계를 분석하였다. 이 연구에서는 워드 임베딩 방법 중 하나인 CCA (Canonical Correlation Analysis)를 이용해 문서의 단어들을 벡터화하고, 코사인 유사도를 계산하여, t-SNE를 통해 2차원으로 만들어 질병과 미생물들의 관계를 분석하였다. 이를 통해, 질병과 미생물들의 관계를 제시하였다.

[6]에서는 WordNet의 특성(경로의 길이, 단어의 깊이)을 이용해 두 단어의 유사도를 측정할 수 있는 방법을 활용하였다. 이러한 유사도 측정은 WordNet에서 단어의 깊이와 최단 경로를 균형 있게 조정할 수 있는 장점을 가지고 있다. 따라서 단어의 Information Content를 활용하여, 두 단어쌍의 동일한 경로와 깊이의 유사점을 찾

[1] https://ko.wikipedia.org/wiki/바이오마커

아낼 수 있는 similarity metric을 제안하였다.

3. 방법론

3.1 데이터

본 논문에서는 구글 학술 검색에서 단어를 검색한 결과의 개수를 이용하였다. 검색한 단어는 질병과 바이오마커를 선택하였다. 이를 통해, 해당 질병과 바이오마커의 유사도를 계산하고자 하였다.

또한 PubMed의 문서 데이터를 활용하여 Word2Vec의 결과를 얻었고, 그 결과를 이용해 코사인 유사도를 계산하고, DNNs 모델을 만들 때 입력 데이터로 사용하였다. 질병과 바이오마커의 종류는 다음의 표1과 같다.

표 1. 실험에 사용한 질병 및 바이오마커 종류

질병 (37 개)	바이오마커 (27 개)
만성폐쇄성폐질환, 천식, 부정맥, 심부전, 심근경색, 심근증, 뇌경색, 고지혈증, 동맥경화, 신부전, 신장결석, 당뇨병, 갑상샘과다증, 백혈병, 간경변, 고혈압, 유방암, 자궁경부암, 위암, 대장암, 폐암, 피부암, 간암, 자궁암, 전립선암, 췌장염, 백색증, 골연골이형성증, 혈우병, 신경섬유종증, 뇌종양, 혈소침착증, 식도암, 후두암, 쓸개암, 고환암	CC-16, rbp, cea, asph, Calprotectin, saa, sp-d, Igfbp-2, Endoglin, Endostatin, trail, Cyfra21-1, Ghrelin, Leptin, nse, pai-1, Angiostatin, ip-10, Adiponectin, il-10, Paraoxonase, C9, Eotaxin-1, dr5, ldl, Nf-kb, il-2

3.2 Word2Vec

Word2Vec[4]은 2013년, 구글의 Mikolov가 제안한 방법으로 인공신경망 모델을 기반으로한 방법이다. Word2Vec은 Skip-gram과 CBOW (Continuous Bag-of-words), 2 가지 모델이 있다. Skip-gram 모델은 하나의 단어를 통해 주변의 단어들을 예측하고, CBOW 모델은 주변의 단어들을 통해 해당 단어를 예측한다. 본 논문에서는 Skip-gram을 이용하여 PubMed의 문서 데이터에서 질병과 바이오마커에 대한 단어 벡터 값을 얻었다. Skip-gram 모델의 구조는 그림 1과 같다.

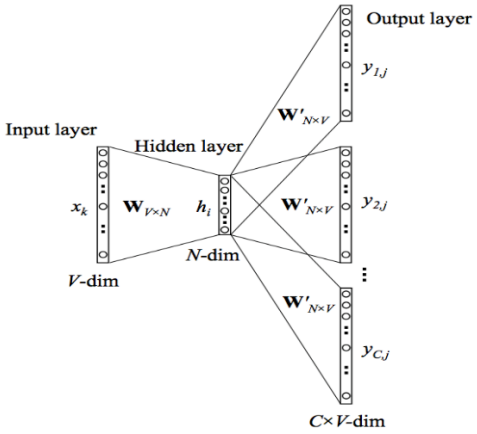


그림 1. Word2Vec의 Skip-gram 모델 구조

3.3 코사인 유사도

코사인 유사도는 내적 공간의 두 벡터간 각도의 코사인 값을 이용하여 측정된 벡터간의 유사한 정도를 측정하는 방법으로, 이 방법은 다차원의 공간에서의 유사도 측정에 자주 이용된다. 코사인 유사도는 두 벡터 X, Y의 내적, 벡터의 크기 등을 이용해 표현하게 된다. 본 논문에서는 코사인 유사도의 결과를 이용해 두 단어의 유사도 계산과의 상관관계를 비교하여 신뢰도를 측정하였다. 코사인 유사도의 계산은 다음의 (1)과 같다.

$$\text{similarity} = \cos\theta = \frac{X \cdot Y}{\|X\|\|Y\|} = \frac{\sum_{i=1}^{n} X_i \times Y_i}{\sqrt{\sum_{i=1}^{n}(X_i)^2} \times \sqrt{\sum_{i=1}^{n}(Y_i)^2}} \quad (1)$$

3.4 단어 유사도 계산

본 논문에서는 구글 학술 검색 기반의 데이터를 이용하여 계산을 하기 위한 계산 방법을 제안한다.

먼저 질병의 검색 개수(W_d)와 바이오마커의 검색 개수(W_m)를 더한 후, 질병과 바이오마커를 동시에 검색했을 때의 검색 개수(W_{d+m})를 나누어 준다. 이 식은 다음의 (2)과 같다.

$$p(W_d, W_m, W_{d+m}) = \frac{W_{d+m}}{W_d + W_m} \quad (2)$$

그 후, $p(W_d, W_m, W_{d+m})$의 결과에 상용 로그 함수를 취한다. 이 식은 다음의 (3)와 같다.

$$L(W_d, W_m, W_{d+m}) = log_{10}{}^{p(W_d, W_m, W_{d+m})} \quad (3)$$

마지막으로, $C(W_d, W_m, W_{d+m})$의 결과를 시컨트(SEC) 함수를 이용하여 계산한다. 이 식은 다음의 (4)과 같다.

$$f(W_d, W_m, W_{d+m}) = sec\, L(W_d, W_m, W_{d+m}) \quad (4)$$

이러한 3 단계의 계산을 통해 질병과 바이오마커의 유

사도를 분석하였다.

3.5. Deep Learning

Deep Neural Networks (DNNs)는 인공 신경망 기술로, 데이터 기반의 예측 방법 중 하나이다. Deep Learning은 현재 자연어처리, 컴퓨터비전 등 다양한 분야에서 좋은 성능을 보이고 있다. 이 방법은 데이터의 양이 많을수록 좋은 성능을 보이며, 또한 매개변수들을 어떻게 설정하는지에 따라 다양한 성능을 보이게 된다. Deep Learning에는 다양한 알고리즘이 존재한다. 본 논문에서는 그 중 DNNs를 이용하였다. 본 논문에서는 DNNs의 구조는 그림 2와 같다.

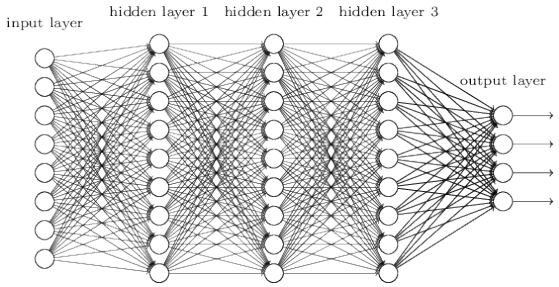


그림 2. DNN의 구조 예시

4. 실험 및 결과

4.1 실험

본 논문에서는 위와 같은 단어의 유사도를 계산하기 위해 질병과 바이오마커의 단어를 Word2Vec을 이용하여 단어에 대한 벡터 값을 얻었고, 이 결과를 이용해 코사인 유사도를 먼저 계산하였다. 그 후, 우리는 코사인 유사도의 결과와 구글 학술 검색 기반의 계산 결과에 대한 상관관계를 계산하여 위와 같은 계산 방법을 선택하였다. 이 때, COPD(만성폐쇄성폐질환)과 27 개의 바이오마커를 선택하여 코사인 유사도를 계산하였고, 이 결과와의 상관관계를 계산하였다. 상관관계를 계산할 때, 코사인 유사도와 3.4의 유사도의 결과에 대한 Rank를 각각 구하였고, 이러한 Rank에 대한 상관관계를 계산하였다. 그 예는 표 2과 같다.

표 2. Rank 상관관계 계산의 예

단어 쌍 (COPD)	코사인 유사도	Rank	계산 결과	Rank
CC-16	0.912140484	1	1.0294	9
SP-D	0.85229974	7	1.2003	16
Leptin	0.76273251	14	1.4872	23
C9	0.618226102	22	4.6947	26

이 때, 코사인 유사도와 단어 유사도 계산과의 상관계수 결과는 약 0.64로 상당히 높은 결과를 얻었다. 따라서 우리는 이러한 계산 방법을 이용하여 유사도를 계산해보고, 이를 기반으로 하여 DNNs 도구 중 하나인 TensorFlow를 이용해 이러한 결과를 학습한 모델을 구축하고자 하였다.

DNNs 모델을 구축할 때, 입력 데이터는 질병과 바이오마커의 단어의 Word2Vec을 이용한 벡터값으로 지정하였다. 그리고 출력 데이터는 계산 결과로 지정하였다. Training data는 질병 29 개, 바이오마커 27로 총 783 개의 데이터를 사용하였다. 그리고 Test data는 질병 8 개, 바이오마커 27개로 총 216 개의 데이터를 사용하였다.

4.2 실험 결과

본 논문에서는 구글 학술 검색 기반에 질병과 바이오마커 단어의 유사도를 계산할 때, 37 가지의 질병 단어와 27 가지의 폐질환과 관련된 바이오마커 단어의 유사도를 계산하였다. 이를 통해 기존에 알려지지 않은 질병과 폐질환 관련 바이오마커와의 관계를 파악하고자 하였다.

계산 방법은 3.4의 계산 방법을 통해 계산하였으며, 폐질환과 관련되지 않은 3 가지의 질병에 대한 계산 결과를 예로 제시한다. 그 예는 다음의 표 3와 같다.

표 3. 3 가지 질병의 유사도 결과 상위 4 가지

질병	바이오마커	유사도 결과
부정맥	ASPH	3.636684
	C9	2.546032
	TRAIL	-1.000802
	SP-D	-1.011931
동맥경화	SAA	34.982258
	Paraoxonase	22.201981
	I1-10	14.631821
	Adiponectin	3.945321
당뇨병	DR5	-1.004244
	SP-D	-1.005012
	Eotaxin-1	-1.011437
	CC-16	-1.017126

또한 질병과 바이오마커 단어의 벡터 값을 입력 데이터로 선정하고, 질병과 바이오마커의 단어에 대한 3.4의 계산 결과를 출력 데이터로 하여 TensorFlow를 이용해 DNNs 모델을 만들고자 하였다. 이를 통해, 3.4의 계산방법을 자동적으로 할 수 있는 모델을 만들고자 하였다. 이 때, 입력 데이터는 질병과 바이오마커의 단어로 각각 2 차원의 벡터, 5 차원의 벡터, 10 차원의 벡터로 총 3 가지의 데이터셋을 구축하였다. 이러한 3 가지의 데이터셋을 가지고 TensorFlow를 통해 어떤 차원의 수를 가진 데이터셋이 학습이 더 잘되는지 실험하였다. Training data로 학습을 한 후, Test data를 통해 예측한 값과 Test data의 정답과의 상관관계를 계산하여 성능을 평가

하였다. 이 때, 각 실험 중 가장 좋은 성능은 다음의 표 4와 같다.

표 4. DNN 모델의 Hyper parameter 및 성능

	2 차원	5 차원	10 차원
Num of hidden layer	5	3	5
Batch size	30	전체	전체
Num of node	80	10	130
Num of epoch	501	501	501
상관 계수	0.1933	0.1951	0.21

실험 결과, 코사인 유사도와 단어 유사도 계산의 상관관계만큼의 성능을 얻을 수 없었다. 3.4의 계산 결과와 예측된 값의 상관 계수는 약 0.2로 상당히 낮은 성능을 보였다.

5. 결론 및 향후 연구 방향

본 논문에서는 구글 학술 검색 기반의 질병과 바이오마커의 유사도를 계산하고자 하였다. 이 때, 구글 학술 검색 기반의 단어 유사도 계산방법은 단어의 벡터 값을 이용한 코사인 유사도의 결과와의 상관 계수가 약 0.64로 상당히 좋은 결과를 얻을 수 있었다. 이러한 문서에서의 질병 단어와 바이오마커의 유사도 계산을 통해 의료 전문가들이 어떠한 질병에 대한 새로운 영향을 가진 바이오마커를 제시하여 의학 분야에 도움이 될 것이라 판단된다. 그러나, DNN의 학습을 통한 예측 성능에서는 약 0.2로 낮은 성능 결과를 얻었다.

현재는 해당 질병과 바이오마커에 대한 단어 벡터값과 유사도 계산 결과를 학습하여 모델을 구축하려하였다. 그러나 향 후, 단어 벡터값과 유사도 계산을 학습하는 것이 아닌 질병과 바이오마커의 단어를 통해 자동적으로 유사도 계산이 가능한 질병과 바이오마커의 관계 분석에 특화된 모델을 만들고자 한다.

참고문헌

[1] T. Pedersen, Information content measures of semantic similarity perform better without sense-tagged text." Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics, Association for Computational Linguistics, 2010.

[2] Muneeb, T.H., Sahu, S.K. and Anand, A., Evaluating distributed word representations for capturing semantics of biomedical concepts, Proceedings of ACL-IJCNLP, 158, 2015.

[3] T. Slimani, Descirption and evaluation of semantic similarity measures approaches, arXive preprint arXiv:1310.8059, 2013.

[4] T. Mikolov, et al., Efficient estimation of word representations in vector space, arXiv preprint arXiv:1301.3781, 2013.

[5] 윤영신·김유섭, “워드 임베딩을 이용한 미생물 관계 분석”, 한국정보과학회 학술발표 논문집, pp.461-463, 2016.

[6] Atoum, I., Bong, C.H., Joint Distance and Information Content Word Similarity Measure, Soft Computing Applications and Intelligent Systems SE-22, 378, 257-267, 2013.

기계 학습형 사용자 맞춤 추천 앱 '눈치 코칭_문화' 개발

전재환O, 이대영, 강현규1)
건국대학교 컴퓨터공학과
wjswoghks1@naver.com, tmsektmsek@naver.com, hkkang@kku.ac.kr

An Android App Development - 'Noonchi Coaching' Which has function of recommendation based on machine learning

Jeon-Jae HwanO, Lee-dae young, Hyun-Kyu Kang*
Department of Computer Engineering, Konkuk University

요 약

본 논문은 공공 데이터 Open API와 사용자의 과거 행동과 주변 상황정보를 토대로 사용자가 선호하는 문화를 맞춤 추천하는 어플리케이션인 '눈치 코칭_문화'의 설계 및 구현에 대하여 서술한다. '눈치 코칭_문화'는 사용자가 쉽게 문화를 추천 받을 수 있도록 만들어진 어플리케이션으로 기존의 필터링 방식으로 사용자가 검색하는 방식의 어플리케이션들과 달리 사용자의 주변 상황과 사용자의 취향 분석을 통해 최적의 문화 Contents를 어플리케이션을 통해 제공한다. 사용자의 별도의 상세검색이나 검색, 좋아요 기능, 주변 위치와 같은 상황 정보를 어플리케이션 사용 로그를 저장 후 데이터 전처리를 하여 사용자에게 다시금 피드백 되는 어플리케이션이다. 지속적인 알림을 통해 사용자에게 문화를 추천하도록 만들었다. 또한, 사용자에게 문화의 날 정보와 사용자 주변 위치의 문화센터를 추천하여 사용자의 문화 활동을 지향한다.

주제어: 추천, 학습형 어플리케이션, 문화, 기계 학습형

1. 서론

우선 본 연구에서 언급하는 문화라는 의미를 설명하고자 한다. 포괄적인 문화의 정의보다 눈치코칭_문화의 문화는 영화, 뮤지컬, 연극, 콘서트, 국악, 무용, 전시, 미술, 문화의 날 정보를 문화라고 칭한다. 그리고 뮤지컬, 연극, 콘서트, 국악, 무용, 전시, 미술을 공연이라고 칭한다.

최근 들어 소비자들의 정보탐색과정과 행동에도 많은 변화가 있다. 모바일 기기를 통한 온라인 접속을 통해 사용자들은 편리하게 필요 정보를 탐색할 수 있게 되었으며, 이로 인해 소비자들의 정보탐색 비중은 점차 확대되고 있다.

하지만 요즘 수많은 문화 contents가 많이 쏟아져 나오고 있다. 사용자는 많은 contents 중에서 자신의 취향에 맞는 contents를 찾는 데는 시간적으로 많은 시간이 소요되고 불필요한 검색도 많다. 그리고 단편적인 정보로 제공되는 contents들 사이에는 어떤 contents가 나에게 맞는 것인지 파악하기는 힘들다. 이러한 흐름 속에서 4차 산업혁명으로 다양한 분야에서 사용자들의 시행착오를 덜어주고, 직접 검색에 드는 시간과 노력의 비용을 아껴주는 매칭 시스템이 각광 받고 있다. 이에 '눈치코칭_문화' 어플리케이션에서는 자동적으로 사용자의 어플리케이션 이용을 데이터로 저장하여 사용자 취향에 맞는 정보를 학습하고 사용자의 양방향성 데이터와 프로파일을 이용하여 사용자에게 맞춤 문화를 추천하는데 목적이 있다.

본 논문의 2장에서는 시스템 구성을 설명하며, 3장에서는 설계 및 구현 내용에 대해 설명한다. 마지막으로 4장에서 결론을 맺는다.

2. 시스템 구성

우리가 개발 하려는 '눈치코칭_문화' 는 모바일 앱 시스템으로 구성하였다. 서버는 사용자가 어플리케이션을 사용한 이용 로그를 데이터화 하여 내부 데이터베이스에 저장한다. 데이터베이스는 JSP 스크립트로 구성하였다.Device에선 외부 데이베이스에서 Open API 정보를 받아온다. 이는 스마트폰으로 3G나 WIFI등으로 연결이 되도록 한다. <그림 1>은 전체 시스템을 간략하게 보여준다.

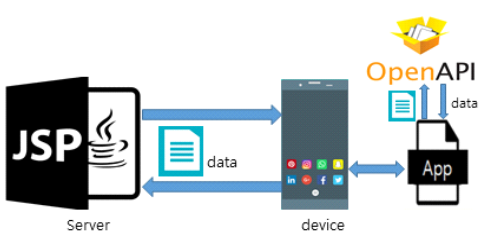


그림 1 시스템 전체 구상

1) Corresponding Author

3. 설계 및 시스템 구현

<그림 5>와 같은 방법으로 영화, 공연에 대한 선호도 조사로 입력받은 데이터를 내부 데이터베이스에 누적한다. 내부 데이터베이스에 저장된 데이터는 사용자가 어플리케이션 초기부터 어느 정도 적절한 추천을 받는데 사용된다. 이후 사용자가 어플리케이션을 사용함에 따라 해당 로그를 내장 데이터베이스에 누적하게 된다. 상황에 따라 사용자 맞춤 추천을 할 때 수집된 데이터를 분석하고, 외부 데이터베이스에서 가져온 contents중 일치율이 높은 순서로 sort하여 다음 추천에 있어 가장 최적의 추천을 예측한다.

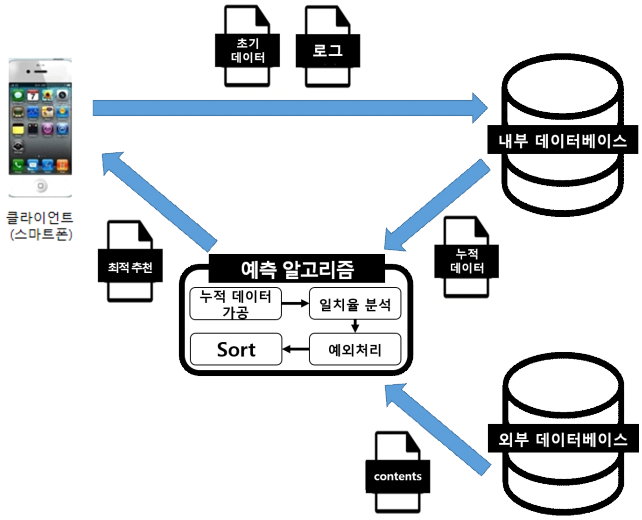


그림 2 시스템 구현도

본 어플리케이션은 사용자 프로파일에 기반을 두어 상황별 선호정보를 예측 추천해주는 어플리케이션 서비스이다. 사용자 프로파일을 기반으로 하여 추천을 하는 어플리케이션이기 때문에 어플리케이션을 설치 후 최초로 설치한 경우 <그림3> 와 <그림 4> 과 같이 사용자의 기본 정보를 입력받는다.

그림 3 성별 입력

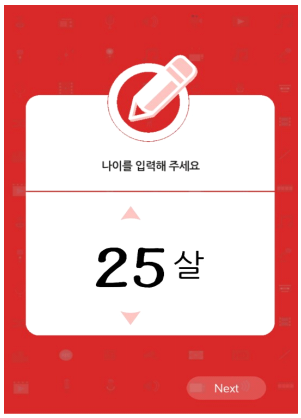


그림 4 나이 입력

또한 <그림 5> 와 같이 영화/공연에 대한 최초 선호도도 입력받는다.

그림 5 영화 장르 선호도 입력

3.0 Splash

어플리케이션을 실행하면 Splash을 화면에 출력한다. Splash화면에서는 최초사용자와 기존사용자 구분, 사용자 선호도 질의, 영화 및 공연 API 질의, 추천알고리즘의 4가지 일을 하게 된다.

먼저, 사용자의 기기 고유 ID를 내부 데이터베이스에 존재 여부를 질의한다. 사용자의 고유 ID가 내부 데이터베이스 안에 없다면 <그림 3>, <그림 4>, <그림 5>에서 언급한 최초 기본정보를 입력 받기 위한 Activity를 실행한다. 만약 사용자가 이전에 본 어플리케이션을 사용한 기록이 있다면 다음 단계인 사용자 선호도 질의 단계로 넘어간다.

두 번째, 사용자 선호도 질의 기능은 기존 사용자가 본 어플리케이션을 사용함으로써 증가/감소 된 선호도를 내부 데이터베이스에 질의하여 본 어플리케이션으로 가져온다. 내부 데이터베이스로부터 가져오는 선호도로는 우선 영화와 공연 중 무엇을 더 선호하는지, 어떤 장르의 영화를 선호하는지, 어떤 분야의 공연을 선호하는지에 대한 정보이다.

세 번째, 이전 단계에서 가져온 사용자 선호도를 기반으로 외부 데이터베이스에 영화, 공연 정보를 질의한다. 질의할 때 사용자 선호도를 기반으로 질의하기 때문에 너무 방대하거나 쓸모없는 데이터는 제외하여 본 어플리케이션으로 가져올 수 있다.

마지막으로, 추천알고리즘에서는 세 번째 단계에서 가져온 데이터와 사용자 선호도간에 가장 많이 일치하는 최적의 contents를 선정하여 Main Activity를 실행한다.

그림 6 Splash activity 화면

3.1 Main Activity

Main Activity에서는 사용자의 선호도 가중치를 이용하여 사용자의 취향 맞춤 contents를 가장 상단에 표시하고 그 뒤를 이어 그 다음으로 사용자 선호도와 일치되는 contents를 화면에 표시한다. 본 화면은 사용자의 선호도가 증가/감소함에 따라 어플리케이션 실행마다 새로운 contents를 화면에 표시한다. 또한 영화/공연간의 선호도에 따라 영화만 출력하거나 공연만 출력할 수도 있고 위치는 유동적으로 변경된다. 우측 하단에는 취향 맞춤 버튼이 위치하고 있다.

그림 7 Main Activity 화면

3.2 Navigation bar

Navigation바는 영화 activity, 공연 activity, 문화의 날 정보 activity, 내 주변 activity로 이동할 수 있다. 또한 초기 실행 시 설정한 자신의 성별과 나이 정보를 확인할 수 있다.

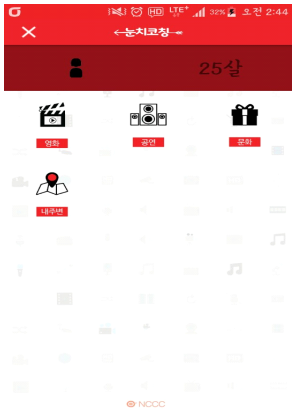


그림 8 Navigation 바

3.3 영화

영화 Activity에는 추천, 신규, 검색 텝이 존재한다. 최초로 영화 Activity에 들어오게 되면 추천 텝을 우선으로 출력한다.

3.3.1 영화 - 추천 텝

영화 Acitivity의 추천 텝에서는 영화 선호도 질의, 영화 정보 질의, 추천 알고리즘의 3가지 일을 한다.

먼저, 사용자의 고유 ID를 이용하여 내부 데이터베이스에 영화 장르에 대한 선호도를 질의한다.

두 번째, 사용자의 선호도를 기반으로 외부 데이터베이스에 영화 정보를 질의한다. 질의할 때 사용자 선호도를 기반으로 질의하기 때문에 너무 방대하거나 쓸모없는 데이터는 제외하여 본 어플리케이션으로 가져올 수 있다.

마지막으로, 추천 알고리즘을 통해 가져온 영화 정보와 사용자 선호도간에 많은 부분이 일치되는 영화들을 선정하여 출력한다.

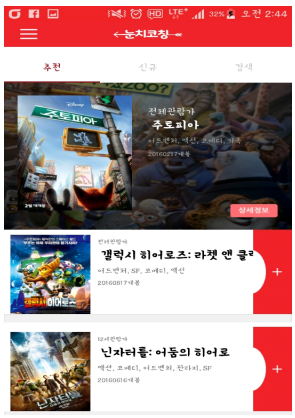


그림 9 영화 Activity 추천 텝

3.3.2 영화 - 신규 텝

영화 Activity의 신규 텝에 들어오게 되면 외부 데이터베이스에 현재 날짜를 기준으로 15일 이전부터 15일 이후 내에 개봉하는 영화를 질의하여 화면에 표시한다.

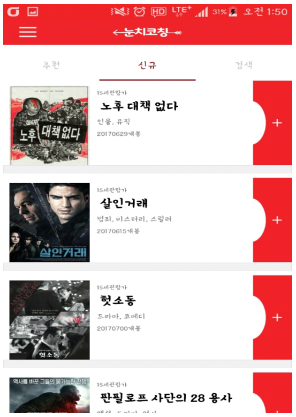


그림 10 영화 Activity 신규 탭

3.3.3 영화 - 검색 탭

영화 Activity의 검색 탭에서는 사용자가 특정 키워드를 직접 입력하여 일치하거나 포함된 영화를 검색할 수 있다.

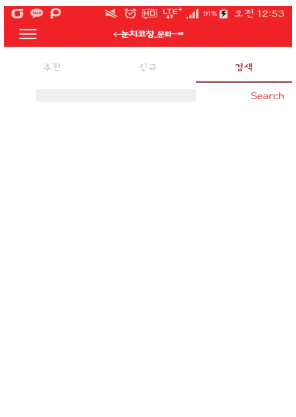


그림 11 키워드 검색 전

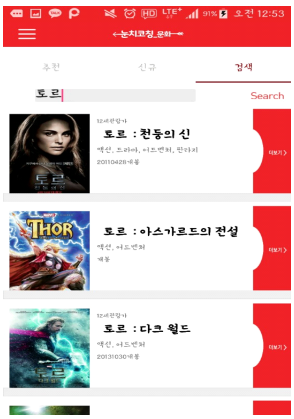


그림 12 키워드 검색 후

3.4 공연

공연 Activity에는 추천, 지역, 분야 탭이 존재한다. 최초로 공연 Activity에 들어오게 되면 추천 탭을 우선으로 출력한다.

3.4.1 공연 - 추천 탭

공연 Acitivity의 추천 탭에서는 공연 선호도 질의, 공연 정보 질의, 추천 알고리즘의 3가지 일을 한다.

먼저, 사용자의 고유 ID를 이용하여 내부 데이터베이스에 공연에 대한 선호도를 질의한다. 질의하는 내용에는 분야, 지역 등 공연과 관련된 모든 선호도를 질의한다.

두 번째, 사용자의 선호도를 기반으로 외부 데이터베이스에 공연 정보를 질의한다. 질의할 때 사용자 선호도를 기반으로 질의하기 때문에 너무 방대하거나 쓸모없는 데이터는 제외하여 본 어플리케이션으로 가져올 수 있다.

마지막으로, 추천 알고리즘을 통해 가져온 공연 정보와 사용자 선호도간에 많은 부분이 일치되는 공연들을 선정하여 출력한다.

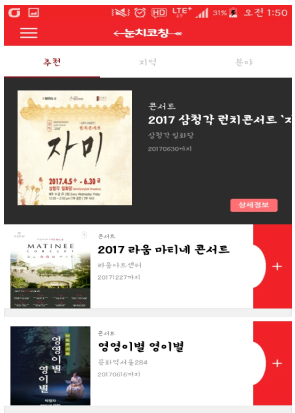


그림 13 공연 Activity 추천 탭

3.4.2 공연 - 지역 탭

공연 Activity의 지역 탭에서는 사용자가 이전에 가장 많이 활동한 지역 질의, 외부 데이터베이스 질의 두 가지 일을 한다.

먼저, 사용자의 고유 ID를 이용하여 내부 데이터베이스로부터 사용자의 공연 이용 기록을 확인하여 가장 많이 공연을 관람한 지역을 파악한다.

두 번째. 가장 공연 관람을 많이 한 지역에서 하는 공연을 외부 데이터베이스에 질의하여 화면에 출력한다.

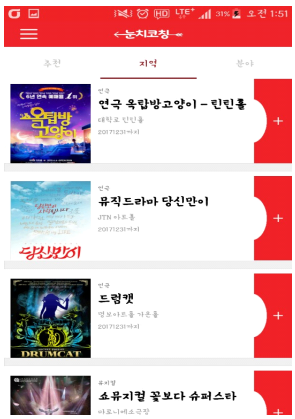


그림 14 공연 Activity 지역 탭

3.4.3 공연 - 분야 탭

공연 Activity의 지역 탭에서는 사용자가 이전에 가장 많이 관람한 공연 분야(연극, 뮤지컬, 음악회 등) 질의, 외부 데이터베이스 질의 두 가지 일을 한다.

먼저, 사용자의 고유 ID를 이용하여 내부 데이터베이스로부터 사용자의 공연 이용 기록을 확인하여 가장 많이 공연을 관람한 공연 분야를 파악한다.

두 번째. 가장 많이 관람한 공연 분야를 외부 데이터베이스에 질의하여 화면에 출력한다.

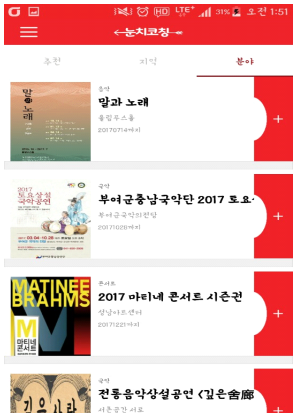


그림 15 공연 Activity 분야 텝

3.5 문화의 날

문화의 날 Activity에서는 당월 문화의 날(매달 마지막 주 수요일) 행사 정보를 외부 데이터베이스에 질의하여 화면에 출력한다.

그림 16 문화의 날 Activity

3.6 내 주변

내 주변 activity에서는 사용자 주변의 문화 센터를 확인할 수 있다. 사용자 주변의 문화센터를 확인하기 위해서 GPS사용권한과 GPS 사용 설정을 요청하여 GPS사용이 가능한 경우에만 실행한다. GPS가 사용 가능한 경우 외부 데이터베이스에 주변 정보를 질의하여 화면에 출력한다.

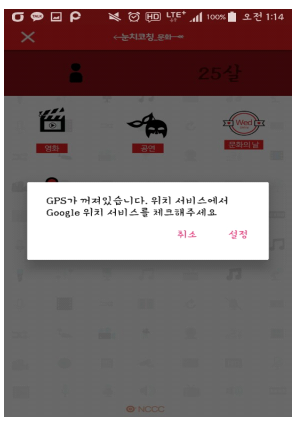


그림 17 GPS사용 요청

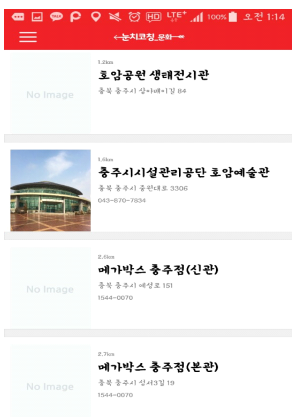


그림 18 내 주변

3.7 맞춤 추천

맞춤 추천 Activity에서는 사용자가 자신의 원하는 조건을 직접 선택 및 입력하여 영화/공연을 검색할 수 있다. 영화의 검색 옵션으로는 장르, 감독, 배우가 있고, 공연의 검색 옵션으로는 지역과 분야가 있다. 장르, 지역, 분야는 다중 선택이 가능한 옵션이다. 모든 선택이 완료된 뒤에는 해당 조건을 이용하여 외부 데이터베이스에 질의를 하고 어플리케이션 이용 기록을 내부 데이터베이스에 저장한다.

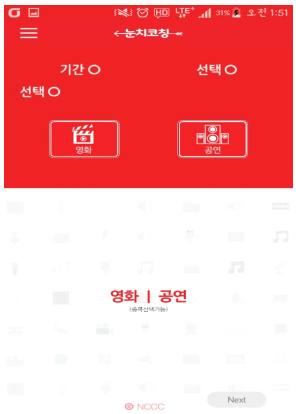


그림 19 맞춤 추천

그림 20 장르 선택

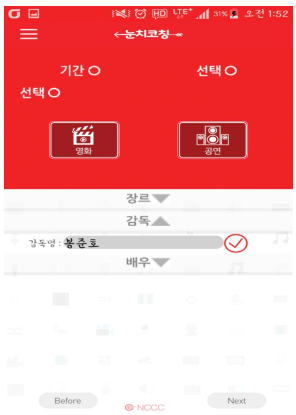


그림 21 감독 옵션

그림 22 지역 옵션

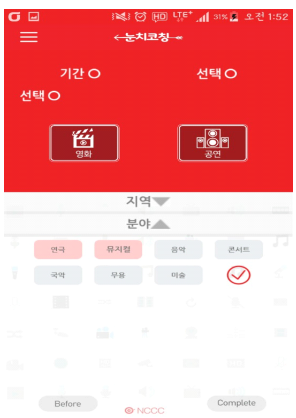


그림 23 분야 옵션

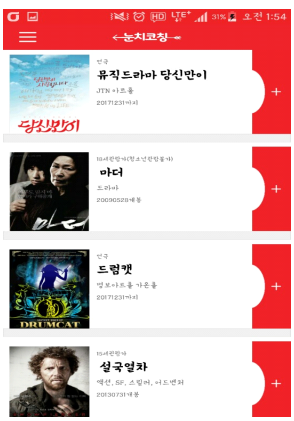


그림 24 검색 결과

3.8 상세정보

상세정보 Activity는 Main, 영화, 공연, 문화의 날, 맞춤 검색 결과 activity에서 contents의 상세정보를 보기 위한 Activity다. 상세정보 Activity에서는 '찜'을 통해 내부 데이터베이스의 해당 contents와 관련된 선호도를 증가/감소시킬 수 있다.

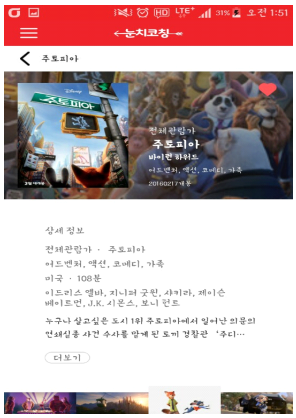


그림 25 상세정보

3.9 알림

'눈치코칭_문화' 어플리케이션은 매주 주말 전 주말간 문화 활동 추천 알림, 특정 공휴일 전 문화 활동 추천 알림, 문화의 날 일주일 전 문화 활동 추천을 사용자에게 알림 하는 서비스를 제공한다.

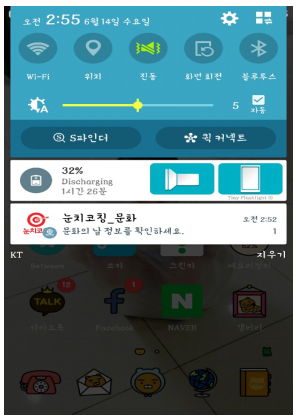


그림 26 알림 서비스

4. 결론

모바일 기기를 통한 온라인 접속을 통해 사용자들은 편리하게 필요 정보를 탐색할 수 있게 되었으며, 이로 인해 소비자들의 정보탐색 비중은 점차 확대되고 있다. 수많은 contents속에 나에게 딱 맞는 contents를 찾는 것은 시간적으로 많은 시간이 소요되고 불필요한 검색이 많다. '눈치 코칭_문화'은 유사 어플리케이션들과는 달리 일부 contents 대해서만 정보를 제공하지 않는다. 폭넓고 다양한 contents를 전 연령층과 성별을 고려하여 상황별 선호 정보를 사용자 각각에 맞는 맞춤을 추천해주는 부분에서 차별화 된 특징을 갖는다.

'눈치 코칭_문화'는 데이터베이스에 저장되어 있는 기존의 사용자 고유 ID를 바탕으로 추천알고리즘에서 가져온 데이터와 사용자 선호도간에 가장 많이 일치되는 contents를 선정하여 사용자 문화 선호도에 맞는 정보를 추천해준다. 이렇게 누적된 데이터는 어플리케이션 접근 시 첫 Activity에 보여 사용자에게 신선한 재미를 선사한다. 그리고 많은 방대한 정보 속에서 나에게 맞는 content를 찾는데 걸리는 시간을 절약하는데 유익함을 준다. 또한 문화의 날 정보, 위치를 파악할 수 있다면 인근에 있는 문화센터 알려주어 사용자의 문화생활에 장려에 보조적인 도움을 줄 수 있게 한다. 마지막으로 주기적인 알람을 통해 어플리케이션이 사용자를 지속적으로 관찰하고 관심을 갖고 있다는 인상을 남겨 친숙하게 다가간다.

이러한 기능들을 바탕으로 '눈치 코칭_문화' 어플리케이션은 공공데이터의 활용성을 높여 살아있는 신선한 데이터들을 자동적으로 모바일에서 얻어올 수 있다. 사용자의 주변 상황정보와 누적된 데이터를 통해 사용자에게 맞는 추천을 하는 점에서 유용함을 보였다.

참고문헌

[1] 김상형, 안드로이드 프로그래밍 정복1, 한빛미디어, 2016.

[2] 정재곤, Do it! 안드로이드 앱 프로그래밍 , 이지스퍼블리싱, 2016.

[3] 한소희, 오동옥, 박지환, 이주경, 류승완. "뮤지컬 공연 관객의 사회관계망 서비스를 이용한 공연정보 추구가치 분석", 예술경영연구, 40, 63-90, 2016.

[4] 송주홍, 김영환, 문남미. "사용자 맞춤형 스마트폰 어플리케이션 추천 시스템 설계", 한국방송미디어공학회, 학술발표대회, 논문집, 156-159, 2010.

[5] 문이해성, 권준희. "상황 정보와 폭소노미를 이용한 협업 필터링 모바일 콘텐츠 추천 어플리케이션", 한국정보기술학회, 7(2), 132-140, 2009.

[6] 김용수, "개인화 서비스를 위한 추천 시스템의 연구동향", ie 매거진, 19(1), 37-42, 2012.

[7] 문은영, 윤회진, 최병주. "상황인식 어플리케이션 개발의 컴포넌트 기반 접근법", 한국정보과학회 학술발표논문집, 33(2C), 422-426, 2016.

위키피디아 QA를 위한 질의문의 정답제약 추출

왕지현○, 허정, 이형직, 배용진, 김현기

한국전자통신연구원
{jhwang, jeonghur, leehj, yongjin, hkk}@etri.re.kr

Answer Constraints Extraction on User Question for Wikipedia QA

JiHyun Wang○, Jeong Heo, Hyungjik Lee, Yongjin Bae, Hyunki Kim
Electronics and Telecommunications Research Institute

요 약

질의응답 시스템에서 정답을 제약하기 위한 위키피디아 영역의 정답제약 9개를 정의하고 질문 문장에서 제약표현을 추출하는 방법을 제안한다. 다어절의 정답제약 표현을 추출하기 위해서 언어분석 결과를 활용하여 정답제약 후보를 생성하며 후보단위로 정답제약 표현을 학습하기 위한 자질을 제시한다. 기계학습 방법을 이용하여 정답제약 후보 별로 정답제약 태그를 분류하여 정답제약 표현을 추출한다. 성능 실험은 각 정답제약 태그 별로 F1-Score 평가를 수행하였다.

주제어: 질의응답, 질문분석, 정답제약

1. 서론

질의응답(QA) 시스템은 자연어 질문에 대해 정답을 찾는 시스템이다. 사용자 질의문으로부터 정확한 검색 의도를 알기 위해서는 질문에 포함된 질문중심어휘(Question Centric Words; 이하 QCW로 표기)와 QCW를 수식하는 질문 내의 표현들이 정답을 찾는 주요 근거가 된다. QCW는 질문에서 정답의 유형을 표현한 어휘를 말하며 이를 정답으로 대치하는 경우 정답가설(Answer Hypothesis)이 된다. QCW를 수식하여 정답을 제약하는 표현들을 '정답제약' 이라고 한다.

(1) Q : "세계에서 가장 높은 빌딩은?"
A : "부르즈할리파"
X='부르즈할리파' <- QCW='빌딩'
<- 정답제약='가장 높다','세계'

(2) Q : "'10일간의 이야기'라는 뜻의 소설집은?"
A : "데카메론"
X='데카메론' <- QCW='소설집'
<- 정답제약='10일간의 이야기'

본 논문은 위키피디아 영역의 자연어 질문에서 출현하는 정답제약 유형을 정의하였으며, 규칙과 기계학습을 사용하여 정답제약 표현을 추출하고 정답제약 유형을 결정하는 방법을 제안한다.

2. 정답제약 유형

위키피디아 영역의 자연어 질문에 대해 고빈도로 출현하는 9개의 정답제약 유형을 정의하였다. 제약마다 1개 이상의 서브필드명(sub-fieldname)이 있다. 예를 들어, 정의제약은 mean과 origin으로 구성되어 있다.

(1) 시간제약 : 정답과 관련된 시간표현
예) "조선 시대 안정복이 지은 역사책은?"
=> time='조선 시대', (QCW='역사책')

(2) 공간제약 : 정답과 관련된 구체적인 장소를 나타내는 공간표현
예) "전한을 세운 사람은?"
=> loc=전한, (QCW=사람)

(3) 별칭제약 : 정답을 달리 부르는 이름
예) "호가 '고산자'인 이 사람은 누구인가?"
=> alias=고산자 (QCW=사람)

(4) 정의제약 : 정답의 의미와 기원
예) "아랍어로 메뚜기를 뜻하며 아랍 에미리트 연방에 속한 도시"
=> mean=메뚜기
=> origin=아랍어, (QCW=도시)

(5) 언어제약 : 정답의 언어(language)
예) "'금성'을 가리키는 순우리말은?"
=> lang=순우리말, (QCW=순우리말)

(6) 부정제약 : 부정하는 대상
예) "독도를 가리키는 말이 아닌 것은 무엇일까?"
1) 우산도, 2) 삼봉도, 3) 가지도, 4) 관음도
=> neg=독도, (QCW=말)

(7) 공칭제약 : 부여된 공식명칭 및 번호. '국보', '보물', '중요무형문화재' 등
예) "이것은 천연기념물 218호로 지정된 곤충이다."
=> type=천연기념물, => number=218호, (QCW=곤충)

(8) 순서제약 : 정답과 관련된 순서 및 최상급 표현

예) “학급에서 끝에서 두 번째로 큰 학생은?”
=> domain=학급, startingPoint=끝, type=두 번째, predicate=큰, target=학생, (QCW=학생)
예) “세계 최초의 인공위성은?”
=> domain=세계, type=최초, target=인공위성, (QCW=인공위성)

(9) 차이제약 : 정답과의 비교대상
예) “사자성어 중에 ‘양’의 의미가 다른 것은 무엇일까?”
=> domain=사자성어, comp=‘양’의 의미, (QCW=것)

3. 정답제약 추출

정답제약을 추출하는 접근방법은 처리 대상 질문 문장에서 정답제약 후보들을 생성하고 각 후보 별로 자질들을 추출하여 기계학습 모델로 학습한 후, 학습된 모델을 사용하여 각 후보가 정답제약인지 여부를 판별하는 것이다(그림1). Sequence Labeling방식을 사용하지 않고 정답제약 후보를 생성하는 이유는 정답제약 추출값의 경계가 구(phrase) 단위를 넘어서는 긴 표현의 경우에 학습량이 많아지게 되고 추출 대상 표현의 중간에서 끊겨서 온전한 추출이 실패할 가능성이 높기 때문이다.

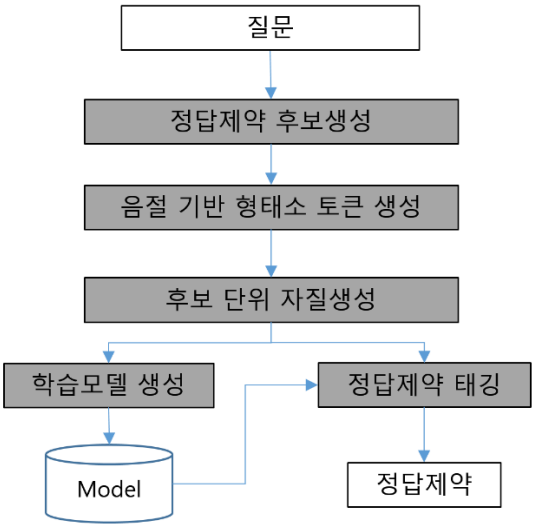


그림 1. 정답제약 추출 흐름도

2.1 정답제약 후보생성

정답제약 후보는 자연어 질문을 언어분석하여 어절, 개체명, 단일명사, 명사구 청크, 절의 수식을 받는 명사구, 기호문자로 둘러싼 인용구문을 대상으로 후보를 생성하였다[1][2]. 예를 들어, 질문 “흰 사슴이 물을 마시던 연못이라는 뜻을 가진 한라산 정상의 화구호 이름은?”이라는 문장에서 정답제약 후보를 생성하면 다음과 같이 생성하게 된다.

(1) 어절 : C1=(흰), C2=(사슴이), C3=(물을),..C11=(이름은)
(2) 개체명 : C12=(한라산)
(3) 단일명사 : C13=(사슴), C14=(정상), C15=(화구호), C16=(이름)
(4) 명사구 청크 : C17=(흰 사슴), C18=(물), C19=(연못), C20=(뜻), C21=(한라산 정상), C22=(화구호 이름)
(5) 절의 수식을 받는 명사구 : C23=(흰 사슴이 물을 마시던 연못), C24=(흰 사슴이 물을 마시던 연못이라는 뜻), C25=(흰 사슴이 물을 마시던 연못이라는 뜻을 가진 한라산), C26=(흰 사슴이 물을 마시던 연못이라는 뜻을 가진 한라산 정상), C27=(흰 사슴이 물을 마시던 연못이라는 뜻을 가진 한라산 정상 화구호 이름은)

위의 후보 생성 규칙은 4,358개 질문 문장으로 구성된 정답제약 태그드코퍼스에 대해서 약 70.83%의 커버리지를 보였다. 후보 유형별 비율을 보면, 명사구 청크가 48.72%로 가장 많은 비중을 차지하였고 개체명이 37.75%, 절의 수식을 받는 명사구가 13.51%의 순으로 커버하였다.

2.2 음절 기반 형태소 토큰 생성

질문 문장에 정답제약을 태깅할 때, 음절 경계에 태깅을 한다. 예를 들어, (흰) (사슴)(이) (물)(을) (마시)(던) (연못)(이)(라는) (뜻)(을) (가진) (한라산) (정상)(의) (화구)(호) (이름)과 같이 최소 분해 단위인 형태소 단위의 토큰을 생성하되 어미활용을 복원하지 않고 음절 단위의 태깅을 한다. 이와 같이 하는 이유는 생성한 정답제약 후보의 경계와 형태소 단위 토큰의 경계를 쉽게 정렬하기 위한 것으로, 수작업한 태그드코퍼스의 정답제약 태깅 위치를 형태소 토큰의 위치에 쉽게 매핑할 수 있다.

2.3 정답제약 자질생성

생성된 각각의 정답제약 후보에 대해서 자질을 생성한다. 후보별로 생성된 자질들은, 학습단계에서 정답제약 레이블과 함께 학습하여 기계학습 모델을 생성한다. 태깅단계에서는 기계학습 분류기의 입력이 되어 정답제약 레이블을 출력하게 된다.

생성하는 자질셋은 2가지가 있으며, 학습단계와 태깅단계에서 (2)의 정답제약 후보 단위 자질을 사용한다.

(1) 토큰 단위 자질
- 형태소 : m-2, m-1, m0, m+1, m+2
- 품사 : p-2, p-1, p0, p+1, p+2
- 개체명 : n-2, n-1, n0, n+1, n+2
- 어휘의미 : s-2, s-1, s0, s+1, s+2

(2) 정답제약 후보 단위 자질
- 좌측 경계 토큰의 토큰 단위 자질
- 우측 경계 토큰의 토큰 단위 자질
- 후보 중심어의 지배소의 형태소와 품사
- 후보 중심어의 술어 지배소의 형태소와 품사
- 후보 중심어의 술어 지배소의 대상격(obj) 논항의 형태소와 품사
- 후보 중심어의 의미역(SRL)
- 좌측 경계 토큰의 피지배소 개수

정답제약 태그드코퍼스 4,358 문장에서 출현하는 숫자 표현들 중에서 순서와 순차적 표현을 나타내는 다음과

같은 숫자들의 형태소 길이 분포를 조사하였다.

예) “제201-1호”, “제11항”, “1,2호”, “제19대”, “두 번째”, “18호”, “2층” 등

코퍼스 내에서 약 95% 이상이 +-2 내의 토큰 거리 안에 포함됨을 알 수 있었다. 이와 같은 이유로 토큰 단위 자질의 거리를 +-2로 정하였다.

2.4 정답제약 학습 및 태깅

정답제약 유형은 총 9개 유형이다. 9개의 정답제약에 정의된 전체 서브필드명을 고려하면 총 16개의 태그가 만들어 지며, 정답제약이 아닌 NOT태그까지 고려하여 17개의 태그를 만들었다. 각 정답제약 후보가 생성하는 자질들에 대해서 17개의 태그를 Structural SVM [3]으로 학습하고 태깅하였다.

3. 성능평가

위키피디아 영역의 장학퀴즈형 자연어 질문 4,358개 문장(평균어절수는 약 18어절)을 개발셋으로 학습하고 별도의 657개 문장을 블라인드 평가셋으로 실험하였다. 정확한 추출경계와 정답제약의 각 서브필드명의 태그를 잘 맞췄는지를 F1-Score로 평가하였다.

표1. 장학퀴즈형 자연어 질문의 성능평가

제약명	개발셋	블라인드셋	제약명	개발셋	블라인드셋
시간	92.15%	80.10%	부정	84.68%	82.17%
공간	94.01%	65.01%	공칭	76.12%	83.87%
별칭	90.27%	65.55%	순서	84.65%	72.39%
정의	79.55%	77.70%	차이	77.97%	92.31%
언어	90.91%	76.60%	-	-	-

위키피디아 영역의 단문형 자연어 질문 2,878개 문장(평균어절수는 약 3어절)을 개발셋으로 하여 학습하고 평가를 하였다.

표2. 단문형 자연어 질문의 성능평가

제약명	개발셋	제약명	개발셋
시간	95.20%	부정	-
공간	92.99%	공칭	64.29%
별칭	96.97%	순서	80.61%
정의	85.71%	차이	-
언어	97.22%	-	

단문형 평가셋에서는 부정제약과 차이제약이 출현하지 않아서 학습과 평가를 할 수 없었다.

4. 결론 및 토의

본 논문은 질의응답 시스템을 위한 질문분석 과정에서 정답을 제약하기 위한 위키피디아 영역의 정답제약 9개를 정의하였고 질문 문장에서 제약표현을 추출하는 방법을 제안하였다. 비교적 긴 형태의 장학퀴즈 질문셋과 짧은 단문형 질문셋에 대한 성능평가를 수행했다.

의미기반의 태깅에 있어서 어렵고 중요한 이슈는 학습 코퍼스를 수작업으로 구축할 때 모호한 점을 최소화하기 위해 태그 별로 명확한 태깅 기준을 정하는 것이고 태깅 일관성을 유지하는 것이다. 정답제약 태깅도 표현에 따라 태깅 여부가 상당히 애매한 경우가 많았으며 코퍼스 구축 작업이 진행될수록 일관성을 유지하기가 쉽지 않았다.

사사문구

본 연구는 과기정통부 및 정보통신기술진흥센터의 정보통신·방송 연구개발 사업의 일환으로 하였음. [2013-0-00131, (엑소브레인-1세부) 휴먼 지식증강 서비스를 위한 지능진화형 WiseQA 플랫폼 기술 개발]

참고문헌

[1] 임수종, 김현기, “의미 정보를 이용한 한국어 의미역 인식 연구”, 제27회 한글 및 한국어 정보처리 학술대회 논문집, pp.18-22, 2015.

[2] 임준호, 배용진, 김현기, 김윤정, 이규철, “의존 구문분석을 위한 한국어 의존관계 가이드라인 및 엑소브레인 언어분석 말뭉치”, 제27회 한글 및 한국어 정보처리 학술대회 논문집, pp.234-239, 2015.

[3] Y.Altun, T.Hofmann and I.Tsochantaridis, SVM Learning for Interdependent and Structured Output Spaces, Proceedings of the ICML, 2004.

워드 임베딩을 이용한 COPD와 암 관련 바이오마커의 상관관계 분석

윤병훈○, 김유섭

한림대학교, 융합소프트웨어 학과
yqudgns1222@gmail.com, yskim01@hallym.ac.kr

Correlation Analysis of Cancer Biomarkers and COPD Using the Word Embedding

Byeong-Hun Yoon○, Yu-Seop Kim
Department of Convergence Software, Hallym University

요 약

본 연구에서는 COPD와 기존에 연관이 있는 것으로 알려진 바이오마커 이외의 새로운 바이오마커를 찾고자 한다. Pubmed Data에서 선정한 암 관련 바이오마커를 추출하여 COPD와 암 관련 바이오마커의 관계를 파악하는 데이터로 사용한다. 그리고 워드 임베딩 모델 중 Word2vec을 사용하여 워드 임베딩 한다. 워드 임베딩한 K차원의 COPD와 암 관련 바이오마커를 t-SNE를 사용하여 시각화한다. 또한 코사인 유사도를 이용하여 COPD와 암 관련 바이오마커의 유사도를 측정한다. 그리고 코사인 유사도와 t-SNE 결과를 이용하여 COPD와 암 관련 바이오마커와의 상관관계를 파악할 수 있으며, 암 관련 바이오마커와 COPD 관련 바이오마커를 비교 하여 기존의 COPD와 연관이 있다고 알려진 바이오마커 이외의 새로운 바이오마커를 찾을 수 있다.

주제어: Word-embedding, COPD, Biomarker, Word2vec

1. 서론

COPD(Chronic Obstructive Pulmonary Disease, 만성 폐쇄성 폐질환)는 천식과 비슷하게 호흡곤란, 기침, 가래 등의 기도 질환 증상을 나타내다가 폐 기능을 악화시켜 사망에 이르게 하는 질병이다[1]. 이와 관련하여 건강 상태를 확인하는데 지표가 되는 바이오마커(Biomarker)[2]를 이용하여 여러 질병을 예측하고, 몸 안의 변화를 알아내는 연구가 진행되고 있다.[3]

바이오마커란 질병을 발견, 모니터링하거나 치료하는 데에 사용되는 신체의 변화를 감지할 수 있는 척도가 되는 생체지표를 가리킨다.

이와 같이, 질병과 바이오마커에 대한 연구가 증가하면서 특정 질병과 관련 있는 바이오마커로 다른 질병의 바이오마커를 찾기 위한 다양한 연구들이 진행되고 있다[4][5].

본 논문에서는 COPD와 관련이 있는 바이오마커 이외에 새로운 바이오마커를 찾기 위하여 발병률이 높은 암(Cancer) 관련 바이오마커를 선정하여 상관관계를 파악한다. Pubmed Data[1]의 전체 문서를 Word2vec[6]을 이용하여 워드 임베딩(Word-Embedding)하고, COPD와 암 관련 바이오마커 38개[2]를 추출한다. 또한, 워드 임베딩한 K차원의 COPD와 암 관련 바이오마커 데이터를 t-SNE(t-distributed Stochastic Neighbor Embedding)[7]를 이용하여 2차원으로 매핑하고 결과를 시각화한다. 그리고, COPD와 암 관련 바이오마커를 코사인 유사도(Cosine Similarity)를 이용하여 COPD와 암 관련 바이오마커 간의 유사도를 측정한다. 마지막으로, 코사인 유사도와 t-SNE 결과를 이용하여 COPD와 암 관련 바이오마커와의 상관관계를 파악한다. 그리고, 암 관련 바이오마커와 COPD 관련 바이오마커를 비교하여, COPD와 밀접한 연관성을 띄는 대체 바이오마커를 추정하고자 하는 것이다.

2. 방법론

본 논문에서는 전체 807,821개의 Pubmed Data의 문서를 Word2vec을 이용하여 워드 임베딩하고, COPD와 바이오마커 38개를 추출하여 실험 데이터로 사용한다. 코사인 유사도 및 t-SNE 결과를 통하여 COPD와 바이오마커의 상관관계를 파악하고, 암 관련 바이오마커와 COPD 관련 바이오마커를 비교한다.

2.1 데이터

본 논문에서는 실험에 사용할 데이터인 암 관련 바이오마커를 선정하기 위하여 국가정보암센터[3] 사이트의 통계자료를 바탕으로 발병률이 높은 암을 선정하여 아래 [표 1]과 같이 암 관련 바이오마커 38개 선정하였다.

1) ftp://ftp.ncbi.nlm.nih.gov/pub/pmc/oa_package/

2) https://en.wikipedia.org/wiki/Tumor_marker
http://www.bea.hi-ho.ne.jp/~ahcc/ahcc18.htm

3) http://www.cancer.go.kr/

[표 1] 암 관련 바이오 마커 38개

바이오 마커	P53, EGFR, ACT, I1-6, Brca1, Brca2, Kras, PSA, Braf, I1-8, CRP, AFP, CD117, MIF, Cytokeratin, CA125, FSH, HER-2/NEU, CD20, CA19-9, TTR, MMP-7, ALK, SOD, SP1, Cortisol, PAP, OPN, HCG, Prolactin, TPA, PTHRP, PDGFR, IAP, HE4, TK, ugt1a1

예를 들어, P53[8]은 암 억제 단백질로 알려져 있으며, 유전자의 돌연변이가 나타나면 암 발병 확률이 증가하게 된다. EGFR[9]은 표피성장인자 수용체로 변이를 통한 과다 발현이나 과다반응은 폐암, 항문암, 등의 암 발생과 관련있다고 알려져 있다. 이외에도, PSA, PAP는 전립선암, HE4는 난소암, SCFR은 위장관 간질종양, CRP는 폐암, Cortisol은 유방암, IAP는 면역력에 중요한 조절인자로 알려져 있다.

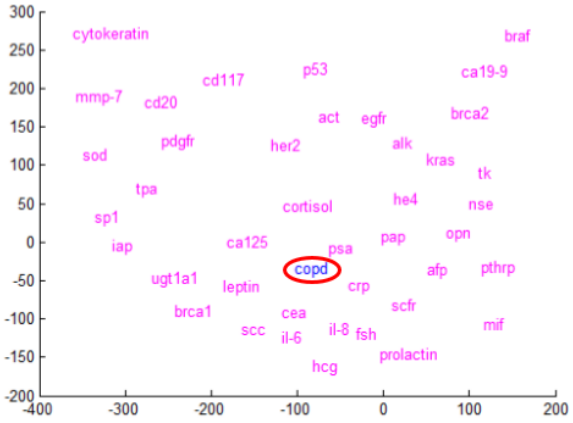
2.2 Word2vec

워드 임베딩은 문서 내에 있는 모든 단어들에 대해 벡터 값을 부여하여 벡터 표현을 학습하는 기술이다.

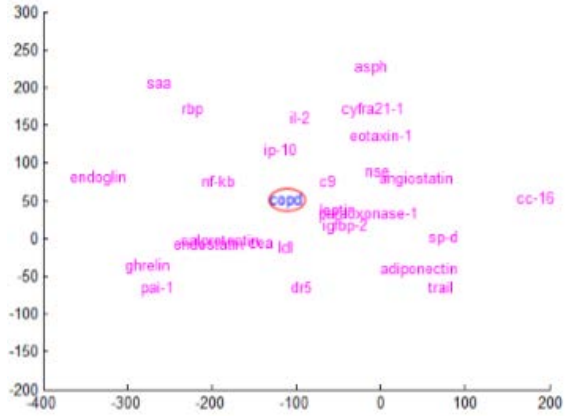
본 논문에서는 워드 임베딩 모델 중 Word2vec을 사용한다. Word2vec은 2013년 구글에서 발표된 연구 모델로 같은 맥락을 지니고 있는 단어는 서로 가까운 의미를 지니고 있다는 전제를 가지고 있다. 또한, 가장 흔하게 텍스트로 된 문장을 이해할 때에 사용된다. word2vec 모델의 학습방법에는 Skip-gram방식과 CBOW(Continous Bag Of Words)방식이 있다[10]. 하지만, 대규모 데이터 셋에서는 Skip-gram이 더 정확한 것으로 알려져있다. 따라서, 본 논문에서는 Skip-gram 방식을 사용하여 5차원, 10차원, 15차원, 20차원, 100차원으로 워드 임베딩한다.

2.3 t-SNE

본 논문에서 워드 임베딩한 고차원의 암 관련 바이오마커의 벡터를 가지고 코사인 유사도를 계산 하게 되면 정확한 유사도를 구하기 어렵다. 따라서, t-SNE를 사용하여 2차원으로 매핑하고 결과를 시각화 한다. 아래 [그림 1]은 암 관련 바이오마커 100차원의 벡터 값을 2차원으로 맵핑하여 가시화시킨 결과이다. [그림 1]에서 파란색은 COPD를 나타내며, 마젠타색은 암 관련 바이오마커를 나타낸다. [그림 2]는 COPD 관련 바이오마커이며, 파란색은 COPD, 마젠타색은 COPD 관련 바이오마커를 나타낸다. [그림 1 ~ 2]을 보면, COPD 기준으로 바이오마커가 어떻게 분포 되어 있는지를 한눈에 알 수 있다.



[그림 1] 암 관련 바이오마커 t-SNE 결과



[그림 2] COPD 관련 바이오마커 t-SNE 결과

2.4 코사인유사도

본 논문에서는 COPD와 바이오마커 간의 유사도 계산하기 위하여 코사인 유사도를 사용하여 유사도를 측정한다. 코사인 유사도는 내적 공간의 두 벡터간 각도를 코사인 값을 이용하여 측정된 벡터간의 유사도이다. 이는 Vector Space Model에서 가장 많이 사용되는 문서와 질의어 간의 유사도 계산법이다. 벡터 A와 벡터 B가 주어졌을 때, 내적과 벡터의 크기 등을 이용하여 표현된다. 계산된 유사도는 -1에서 1사이의 값을 가진다.

$$\text{Similarity} = \text{Cos}(\theta) = \frac{A \cdot B}{||A||||B||} = \frac{\sum_{i=1}^{n} A_i \times B_i}{\sqrt{\sum_{i=1}^{n}(A_i)^2} \times \sqrt{\sum_{i=1}^{n}(B_i)^2}} \quad (1)$$

3. 실험결과

본 논문에서는 코사인 유사도를 사용하여 COPD와 암 관련 바이오마커의 유사도를 계산한다. 아래 [표 2]는 각각 5차원, 10차원, 15차원, 20차원으로 유사도 값를 계산한 결과의 상위 5개씩을 정리한 것이다.

[표 2] 암 관련 바이오마커 유사도 상위 5개

차원	바이오마커	유사도	차원	바이오마커	유사도
5	PSA	0.879	15	PSA	0.795
	PAP	0.870		CRP	0.775
	HE4	0.810		Cortisol	0.753
	SCFR	0.792		SCFR	0.734
	CA125	0.762		PAP	0.714
10	PSA	0.782	20	CRP	0.805
	SCFR	0.762		PSA	0.775
	CRP	0.744		Cortisol	0.744
	Cortisol	0.737		SCFR	0.719
	IAP	0.726		MIF	0.692

그리고, 암 관련 바이오마커와 COPD 관련 바이오마커를 비교하기 위하여 COPD 관련 바이오마커의 각각 5차원 10차원, 15차원, 20차원 유사도 결과를 아래 [표 3]과 같이 정리 하였다.

[표 3] CODP 관련 바이오마커 유사도 상위 5개

차원	바이오마커	유사도	차원	바이오마커	유사도
5	CC-16	0.935	15	CC-16	0.799
	Calprotectin	0.838		Calprotetin	0.768
	Cyfra21-1	0.800		SAA	0.749
	Endoglin	0.770		leptin	0.717
	Leptin	0.763		PON-1	0.699
10	CC-16	0.892	20	CC-16	0.773
	Calprotectin	0.825		Calprotectin	0.769
	PON-1	0.795		SAA	0.752
	Leptin	0.768		Leptin	0.706
	SAA	0.759		PON-1	0.687

[표 3]을 기준으로 [표 2]와 비교하여 암 관련 바이오마커 중 COPD 관련 바이오마커 보다 높은 유사도 계산 결과를 보이는 바이오마커를 추출한다. [표 2] 5차원의 경우, PSA, PAP, HE4, SCFR가 [표 3] 5차원의 CC-16을 제외한 유사도 상위 4개 보다 높게 나왔다. [표 2] 10차원의 경우, PSA, SCFR이 [표 3] SAA보다 유사도 값이 높게 나왔다. [표 2] 15차원의 경우, PSA, CRP, Cortisol, SCFR이 [표 3] Leptin, PON-1보다 유사도 값이 높게 나왔다. 마지막으로, [표 2] 20차원의 경우, CRP, PSA, Cortisol, SCFR, MIF가 [표 3] PON-1보다 유사도 값이 높게 나온 것을 알 수 있다.

[표 2]와 [표 3]을 비교한 결과, 암 관련 바이오마커 중 PSA, PAP, HE4, SCFR, Cortisol, CRP가 COPD 관련 바이오마커와 유사도 값이 비슷하거나 높은 것을 알 수 있었다. 또한, 암 관련 바이오마커 PSA, PAP, HE4, SCFR, Cortisol, CRP 6개가 COPD와 상관관계가 높을 것으로 추정할 수 있다.

4. 결론

본 논문에서는 Pubmed Data의 문서를 Word2vec을 이용하여 워드 임베딩하고, 코사인 유사도 및 t-SNE 결과를 통하여 COPD와 바이오마커의 상관관계를 파악한다. 그리고, 암 관련 바이오마커와 COPD 관련 바이오마커를 비교하여 COPD와 암 관련 바이오마커 사이의 관계를 파악한다.

특정 질병과 유의한 관계를 갖는 바이오마커를 찾는 것은 의료계에서 매우 의미 있는 일이다. 바이오마커는 질병의 사전 진단이나 진행 정도를 모니터링하는데 있어서 매우 중요하다. 또한 특정 질병과 관련이 있을 것으로 추정되는 바이오마커에 대하여 섣불리 임상연구를 진행 할 수 없다는 점을 고려할 때, 문서의 사전 정보에서 관련성을 입증하는 것은 아주 큰 의미를 가진다. 실험결과 상관관계가 높지만 현재 연구되고 있지 않은 COPD 바이오마커 쌍에 대해서는 임상연구를 통하여 보다 정확한 관련성을 입증하는 것이 좋을 것이다. 향후 연구에서 유사도에 대한 분명한 검증을 하여 그 유용성을 입증할 것이고, 더 나아가서 현재까지 관련이 있다고 알려지지 않은 특정 질병과 특정 바이오마커의 조합을 찾을 수 있는 방법을 제시하겠다.

참고문헌

[1] Vestbo J, Hurd SS et al, " Global Strategy for the Diagnosis, Management, and Prevention of Chronic Obstructive Pulmonary Disease", American Journal of Respiratory and Critical Care Medicine, 187(4) : 347-65, 2013.

[2] Aronson, Jeffrey, "Biomarkers and surrogate endpoints", British Journal of Clinical Pharmacology, 59 (5): 491-494, 2005.

[3] Craig E. Wheelock, Victoria M. Goss et al, "Application of'omics technologies to biomarker discovery in inflammatory lung diseases", European Respiratory Journal, 42 : 802-825, 2013.

[4] Young-shin Youn, Chan-youngPark, Jong-daeKim, Hye-jeongSong, Yu-seopKim. "New Biomarker Discovery of Specific Disease using Word Embedding", IMETI, 2016.

[5] Byeong-Hun Yoon, Young-shin Youn, Hye-jeongSong, Jong-dae Kim, Chan-young Park, Yu-seopKim. " Correlation Analysis of BioMedical Entities using Word Embedding", ICBEI, 2017.

[6] T Mikolov, I Sutskever, K Chen, GS Corrado, J Dean, "Distributed representations of words and phrases and their compositionality", Advances in neural information processing systems, 2013.

[7] Maaten, Laurens van der, and Geoffrey Hinton. "Visualizing data using t-SNE.", Journal of Machine Learning Research 9.Nov, 2579-2605, 2008.

[8] Huarte Maite, Guttman Mitchell et al, "A Large Intergenic Noncoding RNA Induced by p53 Mediates Global Gene Repression in the p53 Response", CELL, Volume 142, Issue 3, 2010.

[9] Zhang H, Berezov A, Wang Q, Zhang G, Drebin J,

Murali R, Greene MI, "ErbB receptors: from oncogenes to targeted cancer treatment", The Journal of Clinical Investigation, 117 (8): 2051-8, 2007.

[10] T Mikolov, I Sutskever, K Chen, GS Corrado, J Dean, " Efficient estimation of word representations in vector space", arXiv preprint arXiv:1301.3781, 2013.

문장 벡터와 전방향 신경망을 이용한 스팸 문자 필터링

이현영○, 강승식
국민대학교 소프트웨어학부
le32146@gmail.com, sskang@kookmin.ac.kr

Spam Text Filtering by Using Sen2Vec and Feedforward Neural Network

Hyun-Young Lee○, Seung-Shik Kang
School of Software, Kookmin University

요 약

스팸 문자 메시지를 표현하는 한국어의 단어 구성이나 패턴은 점점 더 지능화되고 다양해지고 있다. 본 논문에서는 이러한 한국어 문자 메시지에 대해 단어 임베딩 기법으로 문장 벡터를 구성하여 인공신경망의 일종인 전방향 신경망(Feedforward Neural Network)을 이용한 스팸 문자 메시지 필터링 방법을 제안한다. 전방향 신경망을 이용한 방법의 성능을 평가하기 위하여 기존의 스팸 문자 메시지 필터링에 보편적으로 사용되고 있는 SVM light를 이용한 스팸 문자 메시지 필터링의 정확도를 비교하였다. 학습 및 성능 평가를 위하여 약 10만 개의 SMS 문자 데이터로 학습을 진행하였고, 약 1만 개의 실험 데이터에 대하여 스팸 문자 필터링의 정확도를 평가하였다.

주제어: 스팸 문자 필터링, 단어 임베딩, 문장 벡터, 전방향 신경망

1. 서론

스마트폰은 사용자의 생활을 편리하고 윤택하게 만들어주는 만큼 빠른 대중화를 이루었다. 이에 반해 스마트폰 대중화에 의한 스팸 문자 메시지 양도 폭발적으로 증가하는 추세이다. 그러한 스팸 문자 메시지의 내용은 성인광고, 대출광고, 게임광고 등이 주를 이루며, 수신자로 하여금 불쾌감을 유발하고 불편을 가중시킨다[1,2]. 스팸 문자 메시지를 접하는 사용자가 늘어나는 만큼 사용자들은 스팸 문자 메시지를 통한 소액결제 및 개인 정보 유출 등에 쉽게 노출이 된다. 이에 따라 스팸 문자 메시지 필터링의 중요성은 점점 커지고 있다.

기존의 스팸 문자 메시지를 자동 차단하는 방식은 크게 '단어 기반 사전을 통한 키워드 매칭' 방식과 '나이브 베이지안(Naive Bayesian), SVM(Support Vector Machine)' 등을 이용한 기계학습 방식으로 구분할 수 있다. 단어 기반 사전을 통한 키워드 매칭 방식은 구현하기가 쉽고 컴퓨터 자원 소모가 적지만 이를 통한 스팸 문자 메시지 필터링은 사용자가 스팸 번호, 스팸 단어 등을 직접 입력해야 하므로 사용자의 편의성이 낮다[3].

스팸 단어의 경우에는 인위적으로 조작되는 경우도 존재한다. 예를 들어, "경마"라는 단어는 "경ㅁr", "야동"이란 단어는 "ㅇk동", "0ㅑ동"으로 표현할 수 있다. 이렇게 의도적으로 조작할 수 있는 단어의 가지 수는 수없이 많기 때문에, 단어 기반 사전을 통한 스팸 문자 메시지 필터링 방식은 효율성이 떨어진다[4,5].

또 다른 스팸 문자 메시지 필터링 방식 중 하나인 기계학습 방법으로는 나이브 베이지안, SVM(Support Vector Machine) 등이 있으며, 이를 사용하기 위해서는 스팸 문자 메시지와 햄 문자 메시지(스팸이 아닌 문자 메시지)를 샘플 데이터로 하여 해당 문자 메시지를 구분할 수 있는 특징 벡터(Feature Vector)를 생성해야 한다. 예를 들어, 특징 벡터를 생성하고자 할 때, 일반적으로 추출하는 특징으로는 특수문자의 빈도, 반복되는 단어의 빈도, 명사의 빈도, 품사 태깅(POS-tagging) 등의 특징을 바탕으로 해당 문자 메시지를 표현하는 특징 벡터를 만든 후 기계학습 방법을 이용하여 스팸 문자 메시지를 필터링한다.

이러한 특징 벡터는 해당 문자 메시지를 표현하는 특징의 종류가 다양할수록 차원 수는 증가하고, 벡터는 희소한(sparse) 형태가 된다[6]. 그림 1을 통해 희소한 형태의 특징 벡터를 확인할 수 있다.

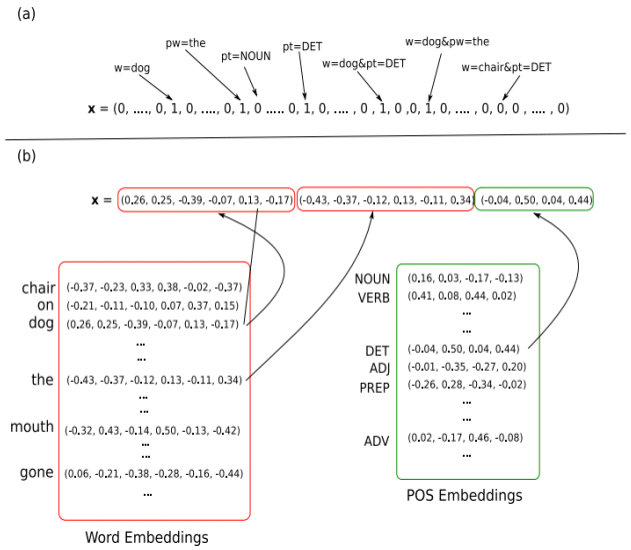


그림 1. 희소한 특징 벡터(위)와 밀집한 특징 벡터(아래)

본 논문에서는 밀집한 특징 벡터 생성을 위해 신경망

언어 모델의 단어 임베딩(Word Embedding)을 이용하여 단어 벡터를 생성하고 이를 기반으로 문자 메시지의 문장 벡터(Sentence Vector)를 생성한다. 그리고 그 문장 벡터와 인공신경망(Artificial Neural Network)의 일종인 전방향 신경망을 이용하여 스팸 문자 메시지를 필터링하는 방법을 제안한다.

2. 단어 벡터와 단어 임베딩

인공신경망의 일종인 딥러닝(Deep Learning)은 컴퓨터 비전, 패턴 인식 그리고 음성 인식 분야에서 뛰어난 성과를 보여주고 있다. 그리고 자연어 처리에서도 그림 2와 같이 딥러닝을 활용한 연구는 빠르게 증가하는 추세이다[7].

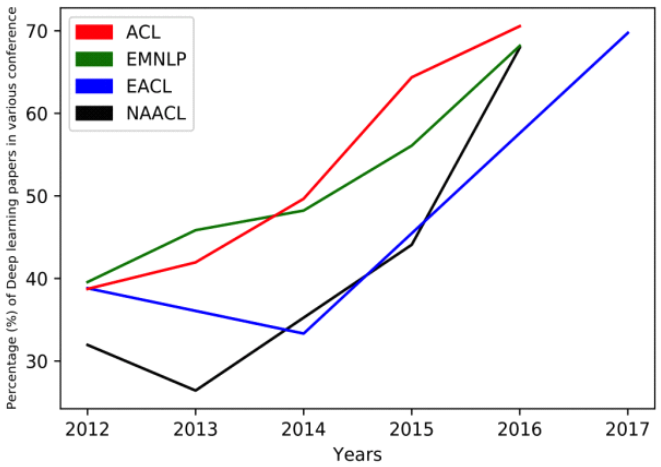


그림 2. 딥러닝 관련 자연어 처리 논문 증가 추세

그림 1과 같이 통계적 기법에서 사용하는 단어 임베딩은 희소한 형태의 단어 벡터를 생성한다. 이는 자연어 처리에서 말하는 차원의 저주인 차원 수 문제와 연산속도 및 메모리 부분에서 효율적이지 못하다.

하지만 인공신경망을 이용한 단어 임베딩은 단어 벡터의 차원 축소 및 확대에 있어서 자유롭고, 연산속도 및 메모리 공간 활용에서도 효율적이다. 이뿐만 아니라 단어 간의 유사도 분석과 단어 사이의 의미적 관계 분석에서도 우수한 효과를 보여 준다.

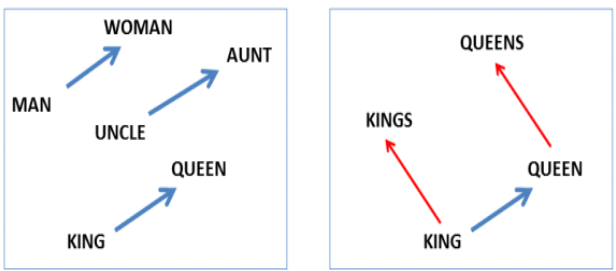


그림 3. 단어 공간에서 단어 벡터 간의 상관관계

그림 3에서 <man-woman>은 man과 woman이라는 단어의 차이 벡터인

word_vector(“man”) - word_vector(“woman”)

을 말한다. <man-woman>의 차이 벡터와 유사한 벡터는 다음과 같다.

word_vector(“uncle”) - word_vector(“aunt”)
word_vector(“king”) - word_vector(“queen”)

즉, 단어 임베딩을 통한 단어 벡터 공간에 각 단어 벡터는 성별이라는 특성을 포함하고 있어, <man-woman>의 차이 벡터는 두 단어의 성별 차이를 나타낸다. 또한 <uncle-aunt>, <king-queen>라는 두 개의 차이 벡터는 <man-woman>의 성별 차이와 유사함을 보여준다. 이러한 인공신경망을 통한 단어 임베딩은 성별, 복수 명사, 단수 명사, 시제 등과 같은 문법적, 의미적 특성을 내포하는 단어 벡터를 생성한다. 이러한 단어 벡터 효과는 자연어 처리 연구에서 우수한 효과를 보여준다[8,9,10].

인공신경망의 구조 중 하나인 전방향 신경망은 선형적 특성과 비선형적 특성으로 이루어진다. 식 (1)은 전방향 신경망의 선형적 특성을 포함하는 식이고, 식 (2)는 식 (1)을 활성화하는 식으로 비선형적 특성을 포함한다. 전방향 신경망 구조는 이 두 식의 조합을 통해 비선형적 특성을 활용한 분류를 가능하게 한다.

F(x) = Wx + b (1)
G(x) = activation(F(x)) (2)

본 논문에서는 단어 임베딩을 통해 생성한 단어 벡터를 이용하여 문장 벡터를 생성하고, 전방향 신경망을 통해 스팸 문자 메시지 필터링 문제에 비선형 분류 기법을 적용하였다.

3. 전방향 신경망을 이용한 스팸 문자 필터링

본 논문에서 제안하는 스팸 문자 메시지 필터링 방법은 전처리 과정으로 자동 띄어쓰기를 한 후, 단어 임베딩을 통해 단어 벡터를 생성하고 문장들을 구성하는 단어의 벡터 합으로 문장 벡터를 생성한다. 최종적으로는 문장 벡터와 전방향 신경망을 이용하여 그 문장이 스팸 문자 메시지인지 아닌지를 필터링하는 방법을 제안한다.

단어 벡터 토큰을 생성할 때 구분자는 공백 문자로 한다. 즉, “01야ㄱ1 사과” 라는 문장에서 단어 벡터 토큰은 각각 “01야ㄱ1”, “사과” 라는 두 개의 토큰으로 나누어진다. 단어 벡터 토큰 생성 시에는 공백 문자만을 구분자로 하여 특수기호, 숫자 등을 이용하여 의도적으로 변환한 “사♥랑”, “♥축하”, “경★ㅁr”, “ㅇk동”, “0ㅏ동” 과 같은 단어 패턴 변화에도 단어 임베딩을 적용하여 단어 벡터를 생성하였다.

전체적인 스팸 문자 메시지 필터링 과정은 그림 4와 같다.

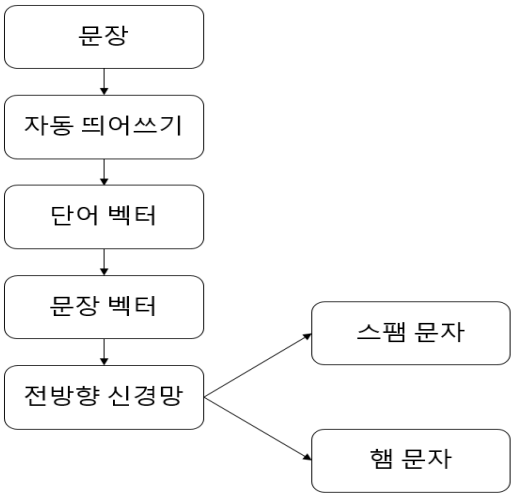


그림 4. 스팸 문자 메시지 필터링 과정

3.1 자동 띄어쓰기

문장을 구성하는 단어 패턴은 다양한 구성을 보여준다. 즉, 하나의 문장은 숫자, 특수문자, 한글, 영어 등의 복합적인 구성이다. 본 논문에서는 복합적인 단어 패턴에서 공백 문자를 구분자로 하여 단어 벡터 토큰을 생성하였다. 하지만 문자 메시지에서 일반 사용자들은 단어를 띄어쓰기하지 않고 한 줄에 연이어 문장을 작성하는 경우가 여러 존재하였다. 이러한 경우에 공백 문자를 기준으로 단어 벡터 토큰을 선택하려 할 때, 문제가 발생하였다. 즉, 띄어쓰기를 하지 않은 긴 길이의 문장도 하나의 단어로 처리하게 되고 이는 희소한 단어를 이용하여 의미 없는 단어 벡터를 생성하게 된다. 이와 같은 문제를 해결하기 위하여 의미있는 단어 벡터를 생성하기 위한 전처리 과정으로 자동 띄어쓰기를 적용하였다.

sentence = “좋은밤되세요내용없음”
word_vector(“좋은밤되세요내용없음”)

자동 띄어쓰기를 적용한 후, 문장은 다음과 같다.

sentence = “좋은 밤 되세요 내용 없음”

자동 띄어쓰기를 적용한 문장의 단어 벡터는 다음과 같다.

word_vector1(“좋은”), word_vector2(“밤”),
word_vector3(“되세요”), word_vector4(“내용”),
word_vector5(“없음”)

3.2 단어 벡터와 문장 벡터

단어 벡터는 신경망 언어 모델 중 하나인 CBOW(Continuous Bag-of-Words) 모델을 기반으로 생성하였다. 그림 5와 같이, CBOW는 단어 W(t)를 중심으로 오른쪽, 왼쪽에 있는 단어를 윈도우 크기만큼 이용하여 중심단어 W(t)의 단어 벡터를 생성한다. 그림 5는 윈도우 크기를 2로 하여, 단어 W(t)를 중심으로 오른쪽 2개의 단어, 왼쪽 2개의 단어를 이용하여 중심단어 W(t)의 단어 벡터를 생성하는 것을 보여준다.

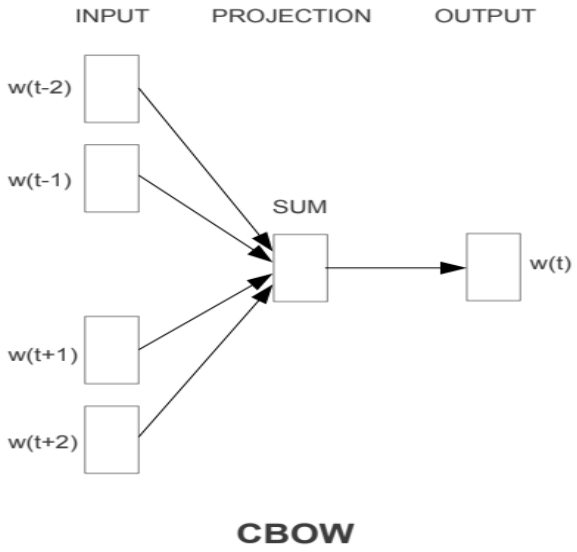


그림 5. CBOW(Continuous Bag-of-Words)

예를 들어 “한글 및 한국어 정보처리 학술대회”라는 문장에서 그림 5와 같이 윈도우 크기는 2이고 CBOW를 이용한다면 “한국어” 단어 벡터는 “한국어”의 오른쪽 2개의 단어(“한글”, “ 및”), 왼쪽 2개의 단어(“정보처리”, “학술대회”)를 이용하여 “한국어” 단어 벡터를 생성한다.

본 논문에서는 단어 벡터 생성을 위해 학습 데이터와 평가 데이터의 총 11만 문장에 포함된 단어들을 기반으로 윈도우 크기가 8인 CBOW를 통해 단어 벡터를 생성하였고, 이 단어 벡터들을 문장 벡터로 합성하였다.

예를 들어, “한글 및 한국어 정보처리 학술대회”의 문장 벡터를 생성하는 과정은 다음과 같다.

x = “한글 및 한국어 정보처리 학술대회”
word1 = word_vector(“한글”),
word2 = word_vector(“및”)
word3 = word_vector(“한국어”)
word4 = word_vector(“정보처리”)
word5 = word_vector(“학술대회”)

Sen2Vec(x) = word1 + word2 + word3 + word4 + word5

3.3 스팸 문자 메시지 필터링을 위한 신경망 구조

전방향 신경망은 완전히 연결된 신경망(FNN: Fully connected Neural Network)으로 계층적 구조로 객체를 분류하는 문제를 해결하는데 사용하는 기본적인 신경망 구조이다. 본 논문에서 사용한 전방향 신경망의 구조는 그림 6과 같다.

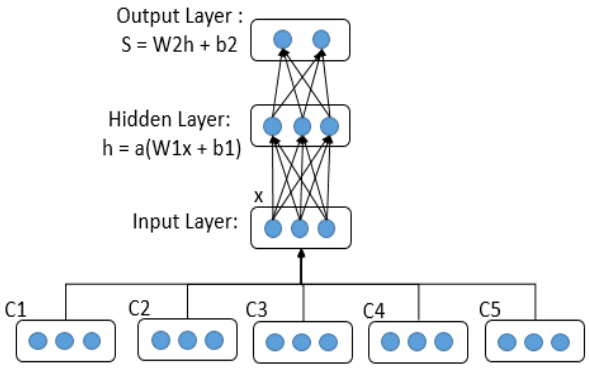


그림 6. 스팸 문자 필터링을 위한 전방향 신경망 구조

이 신경망 구조의 특징은 은닉층(Hidden Layer) 수, 은닉층의 뉴런 개수 조절이 자유롭고, 이를 조절하여 객체 분류 정확도를 개선한다. 그리고 최종 출력(Output)의 값을 이용하여 비용 함수(Cost Function)를 평가하고 객체 분류를 수행한다. 최적의 분류를 수행하기 위해 신경망 구조 기계 학습 모델은 역전파(Backpropagation) 알고리즘을 이용한다. 이 알고리즘은 비용함수의 값을 최소화하는 방향으로 신경망의 인수(W, b)를 학습한다.

본 논문에서 사용한 비용 함수는 크로스 엔트로피(Cross Entropy), 그리고 역전파 알고리즘으로는 경사하강법(Gradient Descent)을 사용하였다.

4. 실험 및 결과

본 논문은 기존의 기계학습 모델인 SVM light와 인공신경망의 일종인 전방향 신경망을 이용한 실험에서 학습 데이터 및 평가데이터, 단어 벡터, 문장 벡터의 차원 수를 표1과 같이 동일하게 사용하였다.

표 1. 데이터 기본 설정

학습데이터	spam	약 5만 문장
	ham	약 5만 문장
평가데이터	spam	약 5천 문장
	ham	약 5천 문장
단어 벡터의 차원수		300 차원
문장 벡터의 차원수		300 차원

*총 어절 수: 약 910,000어절
*한 문장 당 평균 어절 수 : 8.27

표 2는 학습 데이터와 평가 데이터의 총 합인 11만 문장을 기반으로 단어 벡터를 생성하고, 문장을 구성하는 단어의 벡터 합을 통해 생성된 문자 메시지의 문장 벡터를 SVM light와 전방향 신경망을 사용하여 스팸 문자 메시지인지 아닌지를 필터링한 정확도를 보여주고 있다.

기존의 SVM light를 이용한 이진 분류(Binary Classification)의 정확도 보다 전방향 신경망을 통한 이진 분류가 높은 정확도를 보여준다.

또한, 신경망을 이용한 분류 정확도는 은닉층의 수를 증가함으로써 정확도가 개선되는 것을 확인하였다. 하지만 은닉층의 수가 증가하는 만큼 정확도의 증가 폭도 비례하게 증가하기 보다는 감소하였다. 즉 다시 말해, 은닉층의 수가 1에서 2로 증가함에 따라, 정확도의 증가 폭은 1.2로 나오는 반면에 은닉층 수가 2에서 3으로 증가 시, 정확도의 증가 폭은 오히려 0.58로 감소하였다.

표 2. 실험 결과

		정확도			
SVM light		95.25 %			
	활성화 함수	비용 함수	최적화 알고리즘	층수	정확도
전방향 신경망	Sig	크로스 엔트로피	경사 하강법	2	93.72%
				3	94.92%
				4	95.50%

*Sig : Sigmoid Function
*층수: 은닉층 + 출력층

5. 결론

본 논문은 신경망 언어 모델 중 하나인 CBOW를 한국어에 적용하여 생성된 단어 벡터를 문장 벡터로 합성하고, 전방향 신경망을 이용한 스팸 문자 필터링 방법을 제안하였다. 실험 결과를 통해 이진 분류 기능에서 신경망을 활용한 방법이 기존의 SVM light보다 우수할 수 있음을 보여주었다.

신경망 구조에서는 은닉층 수에 따라 정확도를 향상할 수 있음을 확인하였다. 하지만, 은닉층의 수가 증가하는 폭만큼 정확도가 비례하여 증가하지 않았음을 보여주었다. 이를 통해, 전방향 신경망을 이용한 스팸 문자 필터링에 효율적인 은닉 층의 수를 계산하는 방안이 필요할 것으로 생각된다.

본 논문에서는 단어 임베딩을 통해 문장 벡터를 생성하고 전방향 신경망으로 스팸 문자 필터링 방법을 제안했지만, 정확도 향상을 위해 다양한 단어 임베딩(skip-gram, GloVe)을 통한 단어 벡터 생성과 CNN(Convolution Neural Network)을 이용한 문장 벡터 생성 등 다양한 시도를 통해 스팸 문자 필터링의 정확도를 분석하는 시도가 필요할 것으로 생각된다[10].

참고문헌

[1] 강승식, “메일 주소 유효성과 제목-내용 가중치 기법에 의한 스팸 메일 필터링”, 멀티미디어학회 논문지, Vol.9, No.2, pp.255-263, 2006.
[2] 손대능, 이정태, 이승욱, 신중휘, 임해창, “문자 메시지의 특성을 고려한 한국어 모바일 스팸필터링 시스템”, 한국산학기술학회논문지, 제11권, 제7호, pp.2595-2602, 2010.
[3] 이승재, 최덕재, “사용자 맞춤형 스팸 문자 필터링 시스템”, Vol.11, No12, 2011.

[4] 강승식, 장두성, “SMS 변형된 문자열의 자동 오류 교정 시스템”, 정보과학회논문지, 35권, 6호, pp.386-391, 2008.
[5] 김성윤, 차태수, 박제원, 최재현, 이남용, “통계적 기법을 이용한 스팸메시지 필터링 기법”, 한국IT서비스학회지, 제 13권, 제3호, pp. 299-308, 2014.
[6] Goldberg, Yoav. “A Primer on Neural Network Models for Natural Language Processing." Journal of Artificial Intelligence Research(JAIR) 57, pp.345-420, 2016.
[7] Young, T., Hazarika, D., Poria, S., & Cambria, E., “Recent Trends in Deep Learning Based Natural Language Processing,” arXiv preprint arXiv:1708.02709, 2017.
[8] Mikolov, Tomas, Wen-tau Yih, and Geoffrey Zweig. “Linguistic Regularities in Continuous Space Word Representations,” Hlt-Naacl. Vol.13. 2013.
[9] Mikolov, Tomas, et al. “Efficient Estimation of Word Representations in Vector Space,” arXiv preprint arXiv:1301.3781, 2013.
[10] Pennington, Jeffrey, Richard Socher, and Christopher Manning, “Glove: Global Vectors for Word Representation,” Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), pp.1532-1543 2014.

채식주의자: 랭귀지 모델 접근

김재준[0], 권준혁, 김유래, 박명관, 송상헌

동국대학교, 인천대학교

kjj8286@naver.com, sanghoun@inu.ac.kr

A Language Model Approach to "The Vegetarian"

Jaejun Kim[0], Junhyeok Kwon, Yoolae Kim, Myung-Kwan Park, and Sanghoun Song

Dongguk University, Incheon National University

요 약

This paper is to broaden the possible spectrums of analyzing the Korean-written novel "The Vegetarian" by using the computational linguistics program. Through the use of language model, which was usually used in bi-gram analysis in corpus linguistics, to the International Man Booker award winning novel, the characteristics of "The Vegetarian" is investigated by comparing it to the English-written novel "A Little Life".

주제어: Language Model, N-gram Analysis, The Vegetarian

1. Introduction

In this research, the comparison between the 2016 International Man Booker award winning "The Vegetarian" written by Han Kang and the 2015 Man Booker award nominee "A Little Life" written by Hanya Yanagihara is conducted by using the language model analysis. Using "A Little Life" as the reference novel, the translation of the original text, "The Vegetarian" is thoroughly investigated.

As the language models are now being applied in many different kinds of linguistic areas, this study aims to contribute to the area of translation. So far, the researches using the language models compared only the limited expressions between the novels, whereas in this research, comprehensive analysis is conducted. The reference novel and the target novel are different in that their original languages differ, one being English and the other being Korean. Thus, comparing these books can be meaningful in terms of translation. Pattern of sentences or phrases from each novel can also be investigated through the application of the language model.

There are many kinds of language models such as N-gram model, structured model, and class-based model. Among these forms of language modeling, N-gram model will play a pivotal role as a basic foundation. In other words, deep and various analyses of "The Vegetarian" is carefully observed.

2. Background

As mentioned above, when interpreting the literature works, researches using various types of analyses compared only the limited expressions, a word level. Thus, in this research, comprehensive analysis is applied to observe the distinctive patterns within the novel. Since "The Vegetarian" is the first Korean novel that was awarded for the International 2016 Man Booker award, extensive analysis through distant reading approach is necessary in order to find its special aspects.

There are three main reasons for choosing "The Little Life" as a reference novel. First, the general plots between the novel are similar. Both novels have the contents of psychic trauma in

the past, desire, and suicide of the main character. Second, the text itself is about 2.5 times larger than "The Vegetarian". Thus, the larger size of the text can provide excellent standard for being the reference text. Third, "A Little Life" is originally written in English. Since "The Vegetarian" is originally written in Korean, there may be certain characteristics of translated novels. In that way, "A Little Life" can display a criterion of the originally English-written text. For these reasons, "A Little Life" can play an important role as a reference text for "The Vegetarian".

3. Previous Analysis (Roh (2017))

According to Moretti (2005), reading literature writings are divided by two big categories, distant reading and close reading. Other than the traditional way of reading literatures, Moretti (Ibid.) proposed to visualize the text by counting texts and making graphs and maps, as a main characteristic of distant reading. In distant reading, quantified results, which cannot be obtained by close reading, are presented by analyzing the text as a whole.

By implementing distant reading of "The Vegetarian", Roh (2017) divided the text into three different parts where the viewpoint slightly changes in each part. Rho (Ibid.) mainly investigated frequent words of both a whole text and each parts such as (1) and (2).

(1) Frequent Word List (Entire Text)

	Word	Frequency		Word	Frequency
1	like	208	7	face	122
2	just	190	8	eyes	120
3	time	190	9	he'd	120
4	yeonghye	160	10	she'd	116
5	wife	141	11	body	100
6	inhye	125	12	It's	82

(2) Frequent Word List (Part 1)

	Word	Frequency
1	wife	105
2	just	51
3	meet	46
4	time	40
5	life	39
6	face	38

As can be seen from (1) and (2), her work mainly focuses on the unigram analysis. It is noticeable that the frequent word lists slightly changes depending on which part is analyzed.

However, there are several shortcomings to this analysis. First, the POS (Part Of Speech)-tagging was not implemented from the beginning. Words such as '*like*' are used as different POS, depending on the sentences. However, in Roh (Ibid.)'s analysis, '*like*' was investigated according to its different POS usages after already being counted as one as in (1). Therefore, the accurate count of the word '*like*' is somehow confounded. Second, in Roh (Ibid.)'s work, neither BOS (Begin Of the Sentence) nor EOS (End Of the Sentence) were considered. Third, similar to the second reason, Rho (Ibid.) excluded exclamation marks and question marks. The last two reasons are all connected with each other. Roh (Ibid.) did not distinguish the sentences because the main focus was on the unigrams. However, when it comes to n-grams other than just the unigrams, considering the BOS and the EOS is highly important. The bigrams should be limited to sentences because the two consecutive words from different sentences must not be counted as bigram words. Thus, dividing the text as sentence by sentence is highly significant.

4. Methodology

4.1 Language Modeling

Language model refers to the statistical probability of sequential words combination. Among many types of language model, three representative models exist in language processing which include

n-gram model, class-based model, and structured model. Among these types, n-gram model will play a pivotal role as a basic foundation, as mentioned earlier. In this way, n-gram probabilities can help find specific word patterns for "The Vegetarian". Our research used SRILM (Stanford Research Institute Language Modeling) toolkit in applying n-gram language model.

4.2 Procedures

Before applying language model to the text, the text must be preprocessed in order to prevent erroneous results and interpretations. First, tokenization is the start of the preprocessing the given text. Tokenization includes lowering all the capitalized letters, tagging the POS of the words, and removing hyphens. In this way, the words from the text are equally counted and their information is also properly calculated. POS-tagging was conducted by using Standard POS tagger. By implementing these procedures, the text is ready for the language model analysis.

After tokenizing the text, computing the language model is the next step. Thus, by running language model, the frequencies and probabilities of the words can be investigated. One possible problem can occur when counting the frequencies of the words. Since the size of the text can differ from each other, the absolute frequencies cannot be the suitable indicator for the analysis. In language model, this problem is removed because the language model itself reflects the relative frequencies of the words. Thus, it is possible for the researchers to compare frequencies between texts. As a final procedure, smoothing is required in order to avoid possible underflow problem. As its tool, extended interpolated Kneyser-Ney is applied.

5. N-Gram Analysis

As a outcome of the language model, it is shown as an ARPA format such as (3) and (4). APRA format allows us to briefly analyze the result of the language model in a certain form. In APRA format, it is divided into two main parts. Left column show the basic form of the column on the right. The right column describes the actual results of the text. The right column is composed of the number of n-grams at the top. On the bottom, the numbers present the probability of the certain words. As mentioned earlier, the probability is relatively calculated. To the right, the words and their POS-tagging are displayed. Thus, this format allows us to analyze the data in a more efficient and convenient way.

(3) ARPA Format of "The Vegetarian"

/data/	/data(type)/	
ngram 1=n1	ngram 1=6741	
ngram 2=n2	ngram 2=30121	
…	ngram 3=4572	
ngram N=Nn	ngram 4=1829	
/1-grams:	ngram 5=745	
P w [bow]	/1-grams:	
…	-2.435441	like_in
/2-grams:	-3.82404	like_vb
P w1 w2 [bow]	-4.23738	like_vbp
…	…	
/N-grams:	/3-grams:	
P w1 … wN	-0.1808866	there_ex 'll_md be_vb
…	/4-grams:	
/end/	-0.3314916	the_dt nurses_nns '_pos room_nn
	/end/	

In (3), as can be seen under *data(type)*, there is a reduction in the frequently used word combinations from *2-grams* to *5-grams*. It means that since the language model distinguished all the sentences, the number of n-grams decreases depending on the number of the n-grams. Another notable aspect of (3) is that the different POS-tagged usage of *like* is presented in detail under the different POS categories. The different POS of *like* such as *preposition, base form verb,* and *non-4rd person singular present verb* are presented. The prepositional usage of *like* shows the highest probability among other usages, followed by verb usages. On top of that, in 3-grams and 4-grams, the result shows that the language model works perfectly, depending on the given text. Especially in 4-grams, one of the most frequently used word combinations include the word *nurse* because the main content of "The Vegetarian" involves hospital. Thus, the sentences that the language

model calculated are affected by the contents.

(4) ARPA Format of "A Little Life"

```
/data/                  /data(type)/
ngram 1=n1              ngram 1=17638
ngram 2=n2              ngram 2=110999
...                     ngram 3=31373
ngram N=Nn              ngram 4=18793
/1-grams:               ngram 5=9922
P w [bow]               /1-grams:
...                     -2.769982       like_in
/2-grams:               -4.023671       like_vb
P w1 w2 [bow]           -4.304081       like_vbp
...                     -4.78211        like_jj
/N-grams:               ...
P w1 ... wN             /3-grams:
...                     -0.5745651      how_wrb 'd_md you_prp
                        ...
/end/                   /end/
```

In terms of (4), the number of *data(type)* of "A Little Life" is considerably higher than "The Vegetarian", as it is chosen for the reference text. Thus, the number of n-grams reflect the actual volume of the text. Different from (3), it is noticeable to take a look at the variety usage of the word *like*. In this novel, which is originally written in English, another POS of *like* is included. In here, usage of adjective of *like* is added, but its usage is the lowest among *like*. The probability of seeing *like* from both novels displays similar outcome. The most frequently used POS of *like* is the same. Thus, even though the detailed usage of *like* is different in that "A Little Life" used *like* as an adjective, the overall trend of using the word is almost similar.

(5) Bi-grams of "The Vegetarian" and "A Little Life"

The Vegetarian	Probability	A Little Life	Probability
such_pdt a_dt	-0.07123	wo_md n't_rb	-0.02217
able_jj to_to	-0.07909	ca_md n't_rb	-0.04107
unable_jj to_to	-0.07909	able_jj to_to	-0.05225
wo_md n't_rb	-0.08060	lack_nn of_in	-0.05487
trying_vbg to_to	-0.08276	supposed_vbn to_to	-0.06016
want_vbp to_to	-0.10461	hundreds_nns of_in	-0.06022
began_vbd to_to	-0.10617	lots_nns of_in	-0.06022
ca_md n't_rb	-0.10668	kinds_nns of_in	-0.07481
kind_nn of_in	-0.11140	unable_jj to_to	-0.08094
does_vbz n't_rb	-0.11628	version_nn of_in	-0.09753

On top of the ARPA format of both texts, simple bi-gram comparison depending on the probabilities display another similarity as in (5). Since there are two many word combinations in both texts, (5) only briefly lists the words. Among the top ten bi-grams that are likely to appear, both texts had quite a lot of overlapping word combinations. From the top ten bi-grams, four combinations were used in both text with a high probability. About 40 percent of the used word combinations are overlapped.

6. Conclusion

As a conclusion, there is no distinctive characteristics of "The Vegetarian" itself. Although "A Little Life" can show more diverse usages of words, the overall usage of the context is similar. The reason for that is because the originally Korean-written novel "The Vegetarian" and English-written novel "A Little Life" did not display differences in word sequence usages. Even though "The Vegetarian" is translated into English after written in Korean, the translation was focused on conveying the meaning of the original sentence, not on sentence-to- sentence correspondence. From this point of view, we observed that the sketchy features of the novels can be investigated through the means of the language model. With the help of the language model approach, researchers can distinguish whether the novel is written in a English-like way or not in a way.

This research is understandably not perfect in terms of choosing the reference text. In order to develop this analysis of applying language model to the literature works, the reference text needs to be bigger than "A Little Life" in order to set the genuine standards. Thus, this research can be supplemented with the help of the bigger reference text.

References

[1] Moretti, F. (2005). *Graphs, maps, trees: abstract models for a literary history*. Verso.

[2] 노은주. (2017). 문학 작품 "멀리서 읽기":한강의<채식주의자> 번역본 텍스트 분석. *언어와 언어학*, *74*, 75-104

음성인식 기반 리마인더를 위한 시간 표현 분석 기법

박재성, 이상원, 장재나, 강상우
가천대학교, 소프트웨어학과
tmakdlfwotjd@gc.gachon.ac.kr, yohan5050@gc.gachon.ac.kr, jaena96@gc.gachon.ac.kr, swkang@gachon.ac.kr

Time Expression Analysis For Reminder Applications Using Speech Recognition

Jaeseong Park, Sangwon Lee, Jaena Jang, Sangwoo Kang
Department of Software, Gachon University

요 약

본 연구는 리마인더 앱을 위한 효과적인 시간 표현 분석 방법을 제안한다. 시간 표현 분석을 위한 정규식 패턴을 이용하여 사용자 발화 텍스트로부터 시간 정보를 분석하고 시간 표현 유형에 따라 절대적 시간 정보로 변환한다. 제안한 방법은 정규식 패턴을 이용한 시간 표현 분석 기법으로 시스템의 유지 관리가 용이하고 정보량이 많은 패턴과의 매칭을 위해 효과적이다.

주제어 : 리마인더, 정규식, 시간 표현 분석, 음성 인식

1. 서 론

리마인더의 사전적 정의는 잊었던 약속 혹은 해야 할 일 등을 생각나게 하도록 도움을 주는 것을 의미한다[1]. 리마인더는 전통적으로 수기를 통한 메모를 사용하였지만 스마트폰이 보급된 이후로는 다양한 리마인더 앱들이 개발되었다. 초기의 리마인더 앱들은 텍스트를 기반으로 개발되었지만 2015년 이후로는 음성인식을 기반으로 한 리마인더 앱들이 주목 받기 시작하였다. 그 이유로는 음성인식률이 비약적으로 향상되었기 때문인데 통계적으로 2013년 단어 에러율(word error rate) 23%인 것에 반하여 2015년에는 8%로 비율이 현저히 낮아졌다[2-3].

리마인더 앱은 먼저 사용자가 알림 받을 시간과 메모를 작성하면 설정된 시간이 되었을 때 알람을 통해 사용자에게 리마인더 기능을 제공한다. 최근 가장 대표적인 리마인더 기능을 갖는 앱은 “빅스비”와 “시리”와 같은 가상 비서 앱이다. “빅스비”와 “시리”는 리마인더 기능 이외에도 다양한 기능들이 포함되어 있어 리마인더 기능만을 위해서는 여러 가지 제약이 있다.

첫 번째로 가상 비서 앱들은 “리마인드”혹은 명시적인 시간표현이 없으면 리마인드 명령으로 인식 되지 않는 문제가 있다. 두 번째로는 분석 가능한 시간 표현이 매우 제한적이다. 예를 들어, “사흘 후” 혹은 “보름” 등과 같은 표현을 “빅스비”와 “시리”를 대상으로 테스트한 결과 이 두 가지 표현은 분석이 불가함을 확인할 수 있었다. 가상 비서 이외에 리마인더 기능만을 위한 앱들은 음성 인식을 통한 메모 기능은 가능하지만 시간 표현을 자동으로 분석하는 기능은 대부분 지원하지 않았다. 안드로이드에서 음성 인식이 가능한 리마인더 앱들(Google Keep, Just reminder)을 조사한 결과, 알림의 내용은 음성 인식이 되었지만 알림 시간은 사용자가 직접 설정해야만 리마인더 기능이 정상적으로 동작하였다.

본 연구는 Memorade 프로젝트의 일환인 Memory Trigger 앱을 위해 진행되었으며 사용자 발화 텍스트로부터 효과적인 시간 표현 분석을 위한 방법을 제안한다. 다음 장은 다양한 한국어 시간 표현 패턴 구축과 정확한 시간 표현 분석을 위한 매칭 기법 및 절차를 설명한다.

2. 정규식을 이용한 시간 표현 분석

본 연구는 음성 인식을 위하여 Google Speech API를 사용하였으며 시간 표현 분석을 위해 정규식(regular expression)을 사용한다. 그리고 상대 시간 표현의 인식 범위는 최대 한 달로 정하였다.

2.1 전 처리 및 정규식을 이용한 분석 기법

그림 1은 제안한 방법의 처리과정을 보여준다. 사용자의 발화는 Google Speech API를 통해 인식한다. 인식된 텍스트는 부정확한 띄어쓰기를 가지고 있는 경우가 많으며 부정확한 띄어쓰기는 시간 표현 분석의 정확성을 저하시키는 요인이 되므로 인식된 텍스트의 공백을 모두 제거한다.

사용자의 발화에는 유사한 의미의 시간 표현이 여러 형태로 표현될 수 있으므로 이 경우는 모두 하나의 시간

본 연구는 과학기술정보통신부 및 정보통신기술진흥센터의 SW중심대학지원사업의 연구결과로 수행되었음 (2015-0-00932)

표현으로 통일한다. 예를 들어, “오전”과 비슷한 의미를 가진 단어에는 “아침” 등이 있는데, 이 단어들은 모두 하나의 심볼로 변환하여 모호성을 최소화하고 분석의 효율성을 높인다.

공백 제거와 유사 의미가 통합된 사용자 발화 텍스트는 시간 표현 분석을 위한 정규식 패턴들과 비교하여 절대 시간을 추출한다. 비교 과정에서 텍스트가 두 개 이상의 정규식 패턴과 매칭되는 경우 정보량이 가장 많은 정규식 패턴과 매칭하여야 하기 때문에 정규식들 간의 포함관계에서 가장 상위에 있는 정규식을 선택해야 한다. 아래는 포함관계에 있는 정규식의 예이다.

“(오전|오후)([1-2][0-9])시([1-5][0-9])분”
“([1-2][0-9])시”

사용자 발화 텍스트가 위 정규식들과 매칭된다면, “(오전|오후)([1-2][0-9])시([1-5][0-9])분”이 매칭되어야 한다. 이 경우 사용자 발화 텍스트는 정보량이 가장 많은 정규식 패턴부터 비교한다. 따라서 사용자 발화 텍스트가 정규식 패턴과 최초로 매칭되었을 때 나머지 정규식 패턴들과는 비교 할 필요 없이 정확한 시간 표현을 얻을 수 있다.

최종적으로 분석된 시간 정보를 이용해 자동으로 리마인더 앱에 등록을 진행하며 녹음된 음성 파일, 변환된 텍스트, 알림 예정 시각이 함께 저장된다. 분석을 위한 사용자 발화 텍스트는 띄어쓰기가 제거된 결과이므로 자동 띄어쓰기 모듈을 이용하여 자연스러운 문장을 사용자에게 제공한다.

그림 1. 시간 표현 분석의 처리 과정

2.2 시간 표현 유형 분석 및 처리

사용자 발화 텍스트에 나타난 시간 표현의 유형을 다음과 같이 세 가지로 분류하고 처리 방법을 설명한다.

2.2.1 절대적 시간 표현

절대적인 시간 정보는 리마인드 등록에 필요한 시간 정보가 “월”, “일”, “시”, “분” 그리고 12시간제의 경우는 “오전” 혹은 “오후” 정보가 명시되어 있는 경우를 말한다. 사용자 발화 텍스트에서 절대적 시간 정보가 모두 추출된 경우 추가적인 처리 없이 리마인드 등록이 가능하다.

2.2.2 상대적 시간 표현

절대적인 시간 정보 뒤에 “후”, “뒤”, “지나서” 등의 패턴이 추가되거나 “내일”, “모레”와 같은 시간 표현이 나타난 경우 절대적 시간 정보를 얻기 위한 처리 과정이 필요하다. 현재 시간을 기준으로 상대적 시간 표현에 대응하는 시간 정보를 결합하여 절대적 시간 정보로 변환한다.

2.2.3 정보가 부족한 시간 표현

리마인드 등록에 필요한 절대적 시간 정보 중 하나라도 부족하다면 아래와 같이 세 가지 유형으로 분류하여 처리한다.

[방안1] (다이얼로그 창을 이용하여)사용자에게 부족한 시간 정보에 대해 물어본다.
[방안2] 기준 시간을 설정하고 그를 기준으로 부족한 시간 정보를 결정한다.
[방안3] 사용할 수 있는 정보(문맥, 현재 시간, 지난 일정 등)를 이용하여 부족한 시간 정보를 추측한다.

[방안1]은 정확한 정보를 얻을 수 있는 방안이지만 사용자 편의를 중요시 하는 경우는 적절하지 않을 수 있다. [방안2]의 경우는 사용성은 증가하지만 기준 시간에 따라 정확도가 감소할 가능성이 있다. [방안3] 역시 사용성을 높일 수 있지만 개인 정보 및 많은 리소스를 사용해야 하므로 제약이 많다. 본 연구에서는 모바일 환경 및 리마인더 특성을 고려하여 [방안2]를 선택하였다. [방안2]에서의 기준 시간 설정은 보편적으로 사람들의 활동이 많고 기억할 것이 많은 오전 8시부터 오후 7시 59분까지로 설정한다.

3. 성능평가

표1은 제안 모델과 비교 모델의 성능을 비교한다. 테스트를 위한 사용자 발화는 총 300개이며 남녀 20대부터 60대까지 골고루 분포되어 있다. 평가 결과는 “빅스비” 62%, “시리” 58%의 분석 정확률을 기록하였고 제안 모델은 84%의 성능을 보여주었다

표 1. 제안 모델과 비교 모델의 성능 평가

	제안 모델	시리	빅스비
정확률	84%	58%	62%

표2는 사용자 발화를 제안 모델과 비교 모델들이 분석한 예를 보여준다. “사흘 후”, “보름 후”, “자정” 등과 같은 사용 빈도수가 낮은 상대적 시간 표현들은 비교 모델들에서 분석 하지 못하였다. “익일” 같은 어려운 상대적 시간 표현의 경우 국외에서 제작된 어플리케이션인 “시리” 는 분석 하지 못하였다. 하지만 위의 상대적 시간 표현들은 제안 모델에서 모두 분석이 가능하였다.

표 2. 제안 모델과 비교 모델의 분석 결과 예

* 2017년 9월 29일 오후 5시 기준

시간 표현	정답	제안 모델	시리	빅스비
사흘 후 정오	10월 3일 오후 12시	10월 3일 오후 12시	9월 29일 오후 12시	9월 29일 오후 12시
보름 후 2시	10월 14일 오후 2시	10월 14일 오후 2시	9월 30일 오후 2시	9월 30일 오후 2시
익일 오전 6시	10월 1일 오전 6시	10월 1일 오전 6시	9월 30일 오전 6시	10월 1일 오전 6시
다음주 금요일 자정	10월 6일 오전 0시	10월 6일 오전 0시	10월 6일 오전 9시	10월 6일 오전 9시
2일 후 오전 7시	10월 1일 오전 7시	9월 30일 오전 7시	9월 30일 오전 7시	9월 30일 오전 7시

4. 결론

본 연구는 시간 표현 패턴에 기반한 정규식을 이용하여 사용자 발화 텍스트로부터 시간 정보를 분석하고 시간 표현 유형에 따라 최종적으로 절대적 시간 표현을 계산한다. 정규식 패턴을 이용한 분석은 패턴의 추가와 제거가 용이하여 유지 관리에 이점이 있다.

향후에는 서버 클라이언트 모델로 확장하여 정보가 부족한 시간 표현을 예측하기 위하여 사용자 모델(문맥, 현재 시간, 지난 일정 등)을 이용한 분석 기법을 연구할 계획이다.

5. 감사의 글

본 논문을 위해 연구 방향을 제시해 주시고 세심한 조언을 아끼지 않으신 김원 교수님께 감사의 뜻을 전합니다. 아울러 이형철 교수님께도 감사의 말씀을 드립니다.

참고문헌

[1] http://www.oxfordlearnersdictionaries.com/definition/english/reminder?q=reminder

[2] Malid Shokouhi, Umut Ozertem, Nick Craswell, Did you say U2 or Youtube? Inferring Implicit Transcripts from Voice Search Logs, *Proceedings of the 25th International Conference on World Wide Web*, pp. 1215-1224.

식당 예약 대화 시스템 개발을 위한 한국어 데이터셋 구축

김경민○, 이동엽, 허윤아, 임희석
고려대학교 컴퓨터학과
totoro4007@gmail.com, dongyub63@gmail.com, yj72722@korea.ac.kr, limhseok@gmail.com

Development of Korean Dialogue Dataset for Restaurant Reservation System

GyeongMin Kim○, DongYub Lee, YunA Hur, HeuiSeok Lim
Dept. of Computer Science and Engineering, Korea University

요 약

대화 시스템(dialogue system)은 사용자의 언어를 이해하고 그 의도를 분석하여 사용자가 원하는 목적을 달성할 수 있게 도와주는 시스템이다. 인간과 비슷한 수준의 대화를 위해서는 대량의 데이터가 필요하며 데이터의 양질에 따라 그 결과가 달라진다. 최근 페이스북에서 End-to-end learning 방식을 기반으로 한 영어로 구성된 식당 예약 학습 대화 데이터셋(The 6 dialog bAbI tasks)을 구축하여 해당 모델에 적용한 연구가 있다. 대화 시스템에서 활용 가능한 연구가 활발히 진행되고 있지만 영어 기반의 데이터와는 다르게 식당 예약 시스템에서 다른 연구자들의 연구 목적으로 공유한 한국어 데이터셋은 아직까지도 미흡하다. 본 논문에서는 페이스북에서 구축한 영어로 구성된 식당 예약 학습 대화 데이터셋을 이용하여 한국어 기반의 식당 예약 대화 시스템에서 활용 가능한 한국어 데이터셋을 구축하고, 일상생활에서 발생 가능한 발화(utterance)에 따른 형태 변화를 통해 한국어 식당 예약 시스템 데이터셋 구축 방법을 제안한다.

주제어: 식당 예약 대화 시스템, 한국어 데이터셋, 개체, 다양성

1. 서론

대화 시스템(dialogue system)은 컴퓨터가 사용자의 언어를 이해하고 그 의도를 분석하여 사용자가 원하는 목적을 달성할 수 있도록 도와주는 시스템이다. 컴퓨터는 인간과 비슷한 수준의 대화가 가능하도록 대량의 데이터를 가지고 있어야하기 때문에 대화 시스템은 이러한 데이터의 양질에 따라 결과가 달라진다. 페이스북은 장, 단기 메모리를 통합하는 메모리 네트워크의 지식 기반 연구에서 객관적이고 주관적인 학습효율 향상을 가져온 영화 데이터셋과[1], 대용량의 자연어 및 개체 처리를 위한 간단한 질문 응답 형식의 멀티 태스킹 및 전송 학습의 영향을 수행한 연구[2]에서 사용된 데이터셋을 구축 및 공개하였다. 최근 페이스북에서 end-to-end learning 방식을 기반으로 한 목적 지향형 대화(goal-oriented dialog) 식당 예약 시스템의 모델 학습을 위해 영문으로 구성한 학습 대화 데이터셋(the 6 dialog bAbI tasks)을 구축하여 해당 모델에 적용한 연구가 있다[3]. 대화 데이터를 활용한 연구가 활발히 진행되고 있지만 영어 기반의 데이터와는 달리 식당 예약 시스템에서 활용 가능한 데이터셋을 다른 연구자들이 연구할 목적으로 공유한 한국어 데이터셋은 아직까지 존재하지 않는다. 최근 hybrid code network 구조를 이용하여 영어로 구축된 페이스북의 학습 데이터셋을 한국어로 번역하여 대화 데이터셋을 구축 및 학습시킨 연구[4]가 있으나 단순 번역한 데이터셋을 활용한 정도이다.

본 논문에서는 end-to-end learning 방식의 식당 예약 대화 시스템 개발을 위해 페이스북에서 구축한 영어로 구성된 식당 예약 학습 대화 데이터셋(The 6 dialog bAbI tasks)을 이용하여 한국어 식당 예약 대화 시스템에서 활용 가능한 한국어 데이터셋을 구축하고, 일상생활에서 발생하는 발화(utterance)에 따라 다양한 방식으로 시도한 질의응답 데이터 구축 방식을 제안한다.

2. 데이터 구성

2.1 식당 예약 대화 시스템 데이터 번역

순서	대화 순서	영어	한국어
1	인사	hello	반가워
2	예약 진행	can you make a restaurant reservation with (food, location, party_size, price)	(음식, 장소, 인원, 가격) 으로 예약 할래
3	필요 예약 정보 추가	for (food, location, party_size, price) please	~ 으로 할래
4	예약 수정	instead could it be with ~	~ 로 바꿀 거야
5	식당 리스트 변경	no, this does not work for me	다른 걸로, 아닌 거 같아
6	추가 변경 거부	no	아니
7	예약 확정	it's perfect	마음에 들어, 완벽해
8	장소 확인	can you provide the address	식당 주소 뭐야?, 주소 좀 알려줘
9	연락처 확인	may I have the phone number of the restaurant	식당 연락처 뭐야?, 연락처 좀 알려줘
10	감사 인사	thank you	고마워, 고마워 넌 최고야
11	끝맺음	no thank you	아니

●(음식, 장소, 인원, 가격),(food, location, party_size, price) = '~' 로 대체

[표 1] 대화 시스템 원문 데이터 번역본

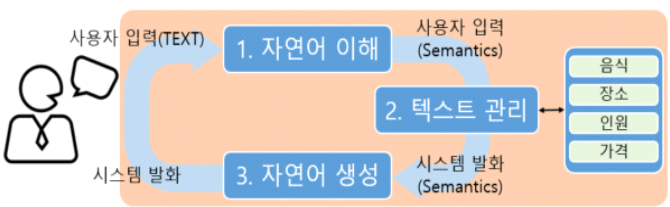
사용자 순서	발화의 목적	발화 형식	다양한 발화 형식의 형태 변화
1	인사	반가워	(생략 가능), 안녕, 안녕하세요, 반갑다, 반가워요, 반갑습니다, 좋은 아침
2	예약	(음식, 장소, 인원, 가격) 으로 예약 할래	~ 가능한 식당 예약 할래, ~ 추천해줘, ~ 가고 싶어, ~ 먹고 싶어, ~ 할게, ~ 하려고, N명이서 외식할거야, ~해줄래?, ~예약 하고 싶어, ~찾아줘, 예약하고 싶은데 찾아 줄래?, ~할 만한 식당이 있을까?
3	~ 예약	~로 바꿀 거야 or ~	~로 해줘, ~ 좋겠어, ~가 좋을 것 같아, ~를 찾고 있어
4	예약 수정	~ 로 바꿔줘 or ~	~말고 ~로 해줘, ~로 바꿀래, 바꾸는 게 좋을 것 같아, 바꾸고 싶어, 바꿀 수 있어? ~로 바꾸자, ~로 바꿔줘, ~ 할게
5	예약 가능성	별로야, 다른 건?	다른 것도 있어?, 마음에 안 들어, 나랑 맞지 않는 것 같아, 너무 별로야 끔찍해, 다른 리스트도 있지?, 내 스타일이 아니야, 다른 건 뭐야, 다른 건 없어?
6	추가 변경 거부	아니	됐어, 없어, 싫어, 충분해, 이 정도면 충분해, 괜찮아, 이게 다예요, 이제 없어 이게 다야, 없는 것 같아
7	예약 확정	좋아, 훌륭해, 마음에 들어, 완벽해	아주, 응(좋아, 훌륭해, 마음에 들어, 완벽해), 진행해줘, 완벽해요, 좋아 보여, 이건 좋아, 아주 완벽해, 바로 이거야, 이게 좋네, 이거로 가자
8	식당 장소 확인	식당 주소 좀 알려줘, 식당 주소를 알 수 있을까	식당 주소는 뭐야?, 주소 좀 알려줄래?, 식당 주소가 어떻게 되죠? 식당 주소가 필요해, 주소 알려줄 수 있지?, 주소 알아?, 식당 위치가 어디야?
9	식당 연락처 확인	식당 연락처 좀 알려줘, 식당 연락처를 알 수 있을까	식당 연락처는 뭐야?, 연락처 좀 알려줄래?, 식당 전화번호 알아? 전화번호를 알고 싶어, 연락처도 있어?, 연락처도 알 수 있지?, 연락처는?
10	감사 인사	고마워, 고마워 넌 최고야	정말 고마워, 알겠어, 응, 최고예요, 넌 정말 최고야
11	추가 도움 거부	아니	됐어, 없어, 싫어, 충분해, 이 정도면 충분해, 괜찮아, 아니 괜찮아, 이걸로도 충분한 것 같아, 훌륭했어, 이거면 충분해요, 필요 없어

●(음식, 장소, 인원, 가격) = '~' 로 대체

[표 2] 발화 형태에 따른 다양한 데이터 변화

위의 [표 1]에서처럼 대화 순서에 따른 영어 원문을 한국어로 번역하여 학습 데이터셋을 구성하였고 각각의 발화(utterance) 순서는 초기 데이터 형식을 이용하였으나, 본 연구의 최종 목표인 데이터셋 활용에서는 발화(utterance)의 순서와 상관없이 학습이 이루어졌다. 한국어 식당 예약 시스템의 대화 데이터셋 구성을 위해 페이스북에서 구축한 영어로 구성된 식당 예약 학습 대화 데이터셋(the 6 dialog bAbI tasks)을 이용하였으며 연구를 위해 사용된 데이터셋은 깃 허브(Git Hub)를 통해 공유하였다[5].

대화 시스템은 사용자의 발화를 입력받아 사용자로부터 식당 예약에 필요한 속성이 될 수 있는 개체(entity)를 추출하는데, 개체 추출을 위해서는 각 개체들의 해당 개체리스트를 정의하고 문자열 매칭 알고리즘을 통해 문자열 속의 개체를 추출할 수 있다[4]. 대화 시스템은 텍스트 관리 과정에서 필요한 개체 정보를 사용자에게 요구하며 누락된 정보는 반복적인 질의 행위를 통해 획득할 수 있다. 대화 시스템 구조와 각 개체에 해당하는 내용은 [그림 1]과 같다.



[그림 1] 대화 시스템 구조

2.2 발화 방식의 다양한 변화

목적에 따른 발화 형식을 바탕으로 한 대화 시스템에서의 발화 형태 변화를 [표 2]를 통해 볼 수 있다. 실제 대화는 대화 시스템에서와 다르게 인사 생략 등 단순 의미 전달 만으로도 대화가 가능한 것을 고려하여 발화 형식 별로 6 ~ 13 개의 다양한 형태로 변화시켰으며, [표 2]에 표기된 사용자 순서 필드와 상관없이 실제 대화가 일어날 수 있는 발화의 목적에 따른 상황에서의 순서를 고려하여 데이터셋을 구축하였다.

식당 예약 시스템과 사용자 간 발화 정보는 [표 3]을 통해 확인할 수 있다. 발화의 주체는 사용자(user)와 시스템(system)으로, 시스템은 사용자로부터 원하는 정보 획득을 위해 반복적인 질의를 이용한다.

발화의 주체	발화	발화의 주체	발화
U(User)	안녕	S	이 식당은 어떤가요? <식당 1>
S(System)	안녕하세요. 어떻게 도와 드릴까요	U	별로야, 다른 곳은 없어?
U	파리에서 3명이 인도 요리를 먹을 수 있는 식당 예약해줘	S	다른 식당 리스트를 보여 드릴게요
S	알겠습니다. 가격의 범위는 어느 정도로 생각하세요?	S	이 식당은 어떤가요? <식당 2>
U	적당한 가격으로	U	아주 만족스러워
S	좋아요 몇 가지 리스트를 보여 드릴게요	S	예약을 진행해드리도록 하겠습니다
S	api_call 인도 프랑스 세 명 적당한	U	식당 연락처 좀 알려줄래?
U	저렴한 가격으로 바꿔줘	S	전화번호는 --- 입니다
S	네 또 변경하실 사항이 있나요?	U	고마워
U	아니	S	또 도와드릴 게 있나요?
S	api_call 인도 프랑스 세 명 저렴한	U	아니 없어
		S	감사합니다

[표 3]식당 예약 시스템과 사용자 간 발화 정보

원하는 개체를 입력받은 시스템은 사용자로부터 개체 변경 의사에 대한 질의를 통해 해당 개체를 수정할 수

있으며 사용자에게 원하는 식당에 대한 정보를 제공함으로써 대화를 마무리한다.

3. 결론

본 논문에서는 한국어 식당 예약 시스템 구축을 위한 식당 예약 학습 데이터셋 수집 방식과 실제 대화에서 일어날 수 있는 발화 방식에 따른 데이터 형식의 형태 변화에 대해 제안하였다. 식당 예약 시스템의 영어 기반 데이터가 구축된 것에 비해 식당 예약 시스템의 한국어 데이터 연구는 아직까지 미흡하며 연구를 목적으로 한 데이터 공유가 존재하지 않는다. 이번 연구에서 사용된 한국어 식당 예약 시스템 데이터를 깃 허브(Git Hub)를 통해 공유함으로써 이번 연구를 토대로 해당 분야의 다른 연구자들에게 많은 기여가 될 것으로 기대한다.

Acknowledgement

본 연구는 문화체육관광부 및 한국콘텐츠 진흥원의 2017년도 문화기술 연구개발 지원 사업으로 수행되었음. [2017. 전통문화 융복합 지원을 위한 지능형 검색 플랫폼 구축]

참고문헌

[1] J. Dodge, A. Gane, Xiang Zhang, A. Bordes, S. Chopra, H.Miller, A. Szlam and J. Weston, Evaluating Prerequisite Qualities for Learning End-to-End Dialog Systems, 2016.

[2] A. Bordes, N. Usunier, S. Chopra and J. Weston, Large-scale Simple Question Answering with Memory Networks, 2015.

[3] A. Bordes, Y. Boureau & J. Weston, Learning End-to-End Goal-Oriented Dialog, 2017.

[4] 이동엽, 허윤아, 임희석, "Hybrid Code Network를 이용한 한국어 식당 예약 시스템 모델", 한국컴퓨터교육학회, 제 21권, 제 2호, pp.57-59, 2017.

[5] 이동엽, 김경민, “Korean Restaurant Reservation,” (2017), GitHub repository, https://github.com/JudeLee19/korean_restaurant_reservation

한국어 학습자 작문 자동 평가를 위한 평가 항목 선정

곽용진○
㈜이르테크
silhuett@iirtech.co.kr

Evaluation Category Selection For Automated Essay Evaluation of Korean Learner

Yong-Jin Kwak○
IIRTECH Inc

요 약

본 연구는 한국어 학습자 작문의 자동 평가 시스템 개발의 일환으로, 자동 평가 결과에 대한 설명과 근거가 될 수 있는 평기 기준 범주를 선정하기 위한 데이터 구축과 선정 방법을 제시한다. 작문의 평가 기준의 영역과 항목은 평가체계에 대한 이론적 연구에 따라 다양하다. 이러한 평가 기준은 자동 평가에서는 식별되기 어려운 경우도 있고, 각각의 평가 기준이 적용되는 작문 오류의 범위도 다양하다. 그러므로 본 연구에서는 자동 평가 기준 선정의 문제는 다양한 평가 기준에 중 하나를 선정하는 분류의 문제로 보고, 학습데이터를 구축, 기계학습을 통해 자동 작문 평가에 효과적인 평가 기준을 선정 가능성을 제시한다.

주제어: 한국어교육, 작문 자동 평가,

1. 서론

자연언어처리 기술을 이용해 사람의 작문 결과를 평가하고자 하는 시도는 종종 진행되어 왔다. 그러나 자연언어처리 기술의 정확성이 사람의 언어 능력에 비해 큰 차이가 있어 널리 적용되지는 못했다. 영어권에서는 미국의 ETS(Educational Test Service)는 오랜 기간 작문 자동 평가(Automatic Essay Evaluation) 기술을 개발해 인간 평가 전문가 평가결과와의 유사도를 70% 수준까지 달성하였다.[1] 이러한 성과는 ETS가 보유한 방대한 언어 자원과 평가 인프라 뿐만 아니라, 다양한 평가 요소에 대한 자동 평가 가능성과 방법에 대한 시도로부터 얻어진 결과이다. 그 목적은 평가 주체가 누구(인간 또는 컴퓨터)인가의 문제가 아니라, 일관되고 투명한 평가 체계를 공개함으로써 신뢰성을 제고하는 데 있다.

국내에서도 [2],[3]에서 한국어 모국어 사용자의 작문 평가를 위한 시스템이 개발된 바 있다. 그러나, 이는 제한된 문항과 형식을 전제로 한 대규모 자료처리를 목적으로 한다. 반면에 한국어 교육 분야에서 작문의 평가는 외국어로서의 한국어 능력을 평가하는 것으로 한국어 교육의 확대와 함께 평가 항목과 기준에 있어서도 다양한 방법이 제시되고 있다. 최근 20 여년 동안 한국어에 대한 세계인의 관심이 높아짐에 따라 TOPIK 등의 평가 체계와 객관성, 신뢰성에 제고에 대한 논의도 확대되고 있다.[4]

본 연구에서는 한국어 학습자 작문의 오류와 한국어 교육에서 널리 통용되는 평가범주와 오류 유형을 한국어 교사에게 제공하고, 각각의 오류에 대해 적합한 범주와 유형을 선택하도록 하여 그 자료 분포를 분석한다. 이러한 평가범주 및 오류 유형 레이블 데이터는 기계학습을 이용한 자동 평가 시스템 개발 뿐만 아니라, 한국어 학습자의 오류에 대한 적절한 평가 기준을 설정하는 데 기여할 수 있다.

2. 관련 연구

ETS는 Grammar, Usage, Mechanics, Style의 4개 영역, 34개 세부항목에 대해 평가를 수행한다.[1] 세부항목은 Subject-Verb Agreement 등과 같은 영어 특유의 문법 기준 뿐만 아니라, 마침표, 쉼표, 물음표 누락과 같은 기초 정서법을 포함하고 있다.

반면에 [3]에서는 학습 데이터 구축을 배제하고, 고득점자 답안으로부터 기능어를 제외한 어휘집합의 군집화를 통해 개념답안을 생성하여 채점자질(평가기준)으로 활용하였다. 이러한 접근은 전문가에 의한 학습 데이터 구축이 최소화되는 데 반해, 평가 결과에서 대한 설명적 근거 제시가 어렵다는 단점이 있다.

3. 연구 방법

한국어 학습자의 작문을 자동 평가하는 과정은 크게 2개의 기능으로 구분할 수 있다. 하나는 학습자의 오류를 식별하는 과정이고, 다른 하나는 식별된 오류의 범주가 무엇인지 판정하는 하는 과정이다. 전자는 입력 어절이 오류인지 아닌지를 추정하는 방법이고, 후자는 현재 어절이 오류일 때 그 범주가 어디에 속하는지를 결정하는 분류의 방법이다.

본 연구는 학습자 작문의 오류를 다수의 한국어 교사에게 제공하고, 그 범주를 결정하도록 한 데이터를 수집, 분석한다. 두 가지 과정을 분리함으로써 학습자의 평가자의 한국어 능력에 내재된 평가기준에 의한 편향을

성을 줄이고, 제시된 오류에 대한 평가(판정)의 기준(명목)이 무엇인지에 집중하도록 한다. 이를 위해 다음과 같은 도구를 제공함으로써 수집과 분석의 효율을 높인다.

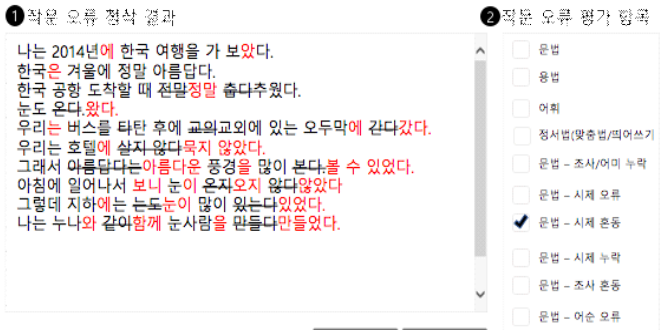


그림 1 작문 오류에 대한 평가 범주 데이터 수집도구

그림 1의 도구는 ①의 화면에 이미 수집된 학습자 작문의 첨삭 결과 자료를 출력한다. 한국어 교사는 ①의 출력 결과중 붉은 색으로 표현된 항목을 클릭하고, ②에서 클릭된 항목에 대한 오류의 평가 기준을 선택한다. ②의 목록에 평가 기준이 없는 경우, 기타를 선택하고 평가 기준명을 작성하여 등록할 수 있다. ①의 화면에 미리 수집된 작문 첨삭 결과를 제공함으로써 한국어 교사는 작문의 오류 여부가 아니라 해당 오류의 평가 기준에 어떤 범주가 적용되어야 하는지만 판단한다.

4. 수집 및 결과

데이터 구축은 그림 1의 도구를 이용하여 한국어 교사 10명에게 각각 100개의 작문 첨삭 결과를 제공하여 구축하였다. 구축된 데이터는 다음과 같다.

표 1 작문 오류 평가 기준 학습데이터 구축 결과

어절 번호	오류어절	교정어절	평가 범주1 (철자법)	평가 범주2 (정서법)	평가 범주...	평가 범주48 (문법-시제오류)
1	간다	갔습니다	0	8	...	792
2	가리킸니다	가르칩니다	38	51	...	0
3	갓수를	가수를	91	157	...	0
4	계절이	계절이	108	22	...	0
...	...	...	...	...	...	...

표 1은 구축 결과에 대해 오류어절과 교정어절이 동일한 항목별로 한국어 교사가 선정한 평가기준을 집계하여 빈도를 산출하였다. 제시된 평가 기준 범주는 철자, 용법, 표현, 문법의 4개 영역, 30개 세부항목이었으나, 최종 수집된 평가 기준 범주는 총 48개 항목으로 증가하였다. 구축된 결과는 Word2Vec 기법을 이용하여 각 평가 기준 범주에 대해 Softmax로 평가하도록 하였다. 오류 어절은 형태분석을 적용하기 어려우나, 교정어절을 보편적인 한국어 형태 분석이 가능하므로 교정어절의 형태분석 결과를 문맥정보로 제공하였다. skip-gram의 윈도우 크기는 교정어절 앞뒤 어절의 1까지도 동적할당 되도록 한 경우의 성능(학습효율)이 가장 좋았다. 또한, 학습된 자동평가 모델에서 평가 범주의 변별력은 표 2와 같다.

표 3 평가범주별 오류 범주 분류 변별력

평가범주	Precision	Recall	F1
철자법	0.67	0.82	0.737
담화-문/구어 구분	0.631	0.76	0.689
담화-담화표지	0.643	0.712	0.675
문법-조사 용법	0.6	0.52	0.557
문법-존대	0.42	0.63	0.504
….	…	…	…
용법-문맥의미	0.16	0.48	0.24
문법-어미활용	0.14	0.27	0.184
어휘 철자법	0.13	0.25	0.171
내용-주제 완성도	0.14	0.12	0.129

철자법, 문법-시제, 문법-존대, 조사 용법, 조사 누락 등 범주는 자동 평가의 가능성이 높았다. 이러한 범주들은 구축결과에서 평가범주가 3개 이하로 나타난 것으로 총 21개 범주가 해당된다.

5. 결론 및 향후 과제

본 연구 결과 첨삭 결과 자료를 이용한 자동 평가의 구현이 어느 정도 가능함을 확인하였다. 그러나, 자동 평가 평가 범주의 정교화, 최적 평가 모델과 특성 정보의 조정 등 보다 많은 데이터 구축과 실험이 필요하다.

그러나, 학습자 작문의 오류 식별과 교정, 평가를 구분함으로써 각 모듈의 자동처리 성능대비 한국어 교사에 의한 사후 검토/교정 양이 감소량이 효과적으로 감소한다는 점에서 본 연구 결과의 의의가 있다.

참고문헌

[1] Charles A. MacArthur, Steve Graham, and Jill Fitzgerald, Handbook of Writing Research Second Edition, The Guilford Press, 2016.

[2] 노은희, 성경희, 임은영, "한국어 문장 수준 서답형 문항 자동채점 적용 가능성 탐색", 교육평가연구, 제28권, 제2호, pp. 523-551, 2015.

[3] 이경호, 이공주, "기계학습을 이용한 중등 수준의 단문형 영어 작문 자동 채점 시스템 구현", 정보과학회논문지 제41권 11호, pp.911-920, 2014.

[4] 이인혜, “한국어 쓰기 평가의 채점 방식에 따른 채점자 신뢰도 연구 : 종합적 채점 및 분석적 채점을 중심으로”, 고려대학교 대학원, 2012.

텍스트 기반 상담시스템의 효율성 제고를 위한 합성곱신경망을 이용한 자동답변추천 시스템

나훈엽○, 서상현, 윤지상, 정창훈, 전용진, 김준태
동국대학교, 컴퓨터 공학과
hoonyeob@dongguk.edu, shseo@dongguk.edu, js_yun@dongguk.edu,
gravity7508@dongguk.edu, yongjin117@dongguk.edu, jkim@dongguk.edu

Automated Answer Recommendation System Using Convolutional Neural Networks For Efficient Customer Service Based on Text

Hunyeob Na, Sanghyun Seo, Jisang Yun, Changhoon Jung, Yongjin Jeon, Juntae Kim
Dongguk University, Department of Computer Science

요 약

대면 서비스보다 비대면 서비스를 선호하는 소비자들의 증가로 인해 기업의 고객 응대의 형태도 변해가고 있다. 기존의 전화 상담보다는 인터넷에 글을 쓰는 형식으로 문의를 하는 고객이 증가하고 있으며, 관련 기업에서는 이와 같은 변화에 효율적으로 대처하기 위해, 텍스트 기반의 상담시스템에 대한 다양한 연구 및 투자를 하고 있다. 특히, 입력된 질의에 대해서 자동 답변하는 챗봇(ChatBot)이 주목받고 있으나, 낮은 답변 정확도로 인해 실제 응용에는 어려움을 겪고 있다. 이에 본 논문에서는 상담원이 중심이 되는 텍스트 기반의 상담시스템에서 상담원이 보다 쉽게 답변을 수행할 수 있도록 자동으로 답변을 추천해주는 자동답변추천 시스템을 제안한다. 실험에서는 기존 질의응답 시스템 구축에 주로 사용되는 문장유사도 알고리즘과 더불어 합성곱신경망을 이용한 자동답변추천 기법의 답변추천 성능을 비교한다. 실험 결과, 문장유사도 기반의 답변추천 기법보다 본 논문에서 제안한 합성곱신경망(Convolutional Neural Networks) 기반의 답변추천시스템이 더 뛰어난 답변추천 성능을 나타냄을 보였다.

주제어: 상담시스템, 자동답변추천, 합성곱신경망, 문장 분류

1. 서론

질의응답 시스템(Question Answering System)이란 사용자로부터 자연어로 구성된 질문을 입력받아 사용자가 원하는 답변을 자동으로 제공해주는 시스템이다. 최근에는 인공지능을 활용한 챗봇(Chat Bot)이 대두됨으로써 상담센터의 업무를 상당부분 대체할 수 있을 것으로 기대되고 있다. 실제로 은행이나 관공서를 방문하기보다는 전화, 혹은 인터넷을 통한 비대면 소통 채널을 선호하는 성향의 고객들이 늘어남에 따라, 국내외 다수 업체에서 챗봇 관련 서비스를 연구 개발하고 출시하여 서비스에 돌입하고 있다.

최근 많은 상담 센터에서는 상담원이 고객들의 다양한 상담을 전화 상담과 채팅을 통해 처리하고 있다. 채팅 상담의 경우 상담원 한명이 한 번에 처리할 수 있는 대화의 수가 제한적이기 때문에 시간, 비용적인 부분에서 효율적으로 처리하기 어렵다. 때문에 챗봇 서비스의 도입의 필요성이 대두되고 있다.

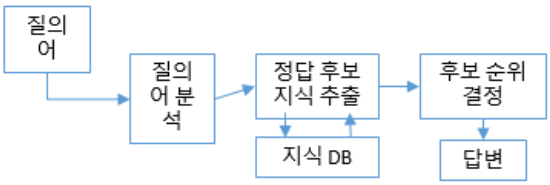
하지만 텍스트 생성모델의 제한과 규칙기반 챗봇의 한계점과 같은 문제로 인해 현실적으로 챗봇을 바로 실제 업무에 투입하기에는 어려움이 있다. 본 연구에서는 외식 주문 상담 업체에서 고객의 질의에 대한 자동 답변을 추천해주는 기능에 초점을 맞춰 주문 상담 센터의 업무 효율을 올릴 수 있는 질의응답 시스템을 설계하고 합성곱신경망을 활용하여 효율성을 높이는 방법을 제안한다.

본 논문의 2장에서는 질의응답 시스템의 구조와 자연어 처리 분야에서 사용되어온 문장 유사도 알고리즘에 대한 관련 연구, 그리고 Word2Vec과 합성곱신경망에 대하여 살펴보고, 3장에서는 기존 연구들의 단점과 본 연구에서 제시하는 모델에 대해서 설명한다. 4장에서는 사용될 실험 데이터를 소개하고, 기존의 알고리즘들과 제시한 모델을 사용한 실험 결과를 비교 분석한다. 마지막으로 5장에서 결론과 향후 연구 방향에 대해서 논한다.

2. 관련 연구

2.1 질의응답 시스템

과거에는 문장 유사도 기반의 다양한 알고리즘을 이용하여 질의응답 시스템을 구축하였다. [그림1]은 기존 연구 중 코사인 유사도를 사용한 질의응답 시스템의 모형이다. 우선 필요한 지식데이터를 구축한 뒤에, 질의어가 들어오면 해당 질의어를 분석하고 질의에 대한 정답 후보가 주어져 있을 때, 구축한 지식데이터를 이용하여 정답후보에 대한 지식을 추출하고 이를 통해 후보 순위를 결정하여 정답을 결정하는 방식이다.[1][2]



[그림1] 자연어처리와 정보검색을 이용한 질의응답 시스템

2.2 문장 유사도 알고리즘

기존의 질의응답 시스템에서는 피어슨 상관계수 (Pearson Correlation Coefficient), 자카드 (Jaccard), 코사인 (Cosine), 타니모토 (Tanimoto), 레벤슈타인 (Levenshtein) 등 문장 유사도를 비교함으로써 질문에 대한 답변을 제시하는 방법들이 많이 사용되었다.[3][4][5] 하지만, 이러한 방법들은 단어가 본질적으로 다른 단어와 어떤 관련성을 가지는지 이해 할 수 없다는 단점이 존재한다.

우선, 피어슨 상관계수는 두 변수간의 관련성을 구하기 위해 사용되는 개념이다. X와 Y가 함께 변하는 정도를 X와 Y가 따로 변하는 정도로 나누어 계산하며 공식은 다음과 같다.

$$r = \frac{\Sigma(x-\bar{x})(y-\bar{y})}{\sqrt{\Sigma(x-\bar{x})^2 \times \Sigma(y-\bar{y})^2}}$$

r값의 범위는 $-1 \le r \le 1$이며, r=1일 경우, x와 y가 완전한 상관관계에 있는 경우를 나타낸다. r값이 0일 경우, 두 변수는 완전한 독립관계이다. 일반적으로 r값이 0.1과 0.3 사이라면, 약한 선형관계, 0.3과 0.7 사이라면 뚜렷한 선형관계, 0.7과 1.0 사이라면 강한 선형관계이다.

자카드 유사도는 두 집합 A와 B를 비교할 때, 교집합의 원소를 전체 원소로 나눈 것으로 계산한다.

$$r = \frac{|A \cap B|}{|A \cup B|}$$

두 종류의 개체가 가지고 있지 않은 것을 제외하고, 동일한 특성이 많을수록 r값이 증가한다.

코사인 유사도는 내적 공간의 두 벡터의 유사도를 측정한다. 두 벡터간 각도의 코사인으로 측정하며, 두 벡터가 동일한 방향을 향하고 있는지 여부를 측정한다. 코사인 유사도는 문서를 비교하는데 사용되기도 한다. 코사인 유사도의 계산식은 다음과 같다.

$$\mathrm{sim}(x,y) = \frac{x \cdot y}{\|x\| \, \|y\|}$$

타니모토 계수는 정보수집에서 자주 사용되는 방법으로 계산식은 다음과 같다.

$$T = \frac{A \cap B}{A \cup B - A \cap B}$$

레벤슈타인 거리는 두 개의 문자열이 얼마나 유사한지 알아내는 알고리즘으로 문자열 A와 B가 존재할 때, A가 B와 같아지기 위해서 몇 번의 연산이 필요한지 계산하는 개념이다. 본 논문에서는 레벤슈타인 거리의 개념에 단어 기반으로 한 one-hot encoding 방식을 적용시킨다면 문장 유사도를 측정이 가능할 것이라는 가설을 세우고 실험을 수행하였다.

2.3 유사도기반 답변추천 시스템의 문제점

앞에서 살펴본 기존의 문장 유사도를 기반으로 하는 예측은 다음과 같은 한계점을 갖고 있다.

첫째, 두 사용자 프로필 간의 상관관계는 두 사용자가 평가 한 항목을 기반으로만 계산할 수 있다. 즉, 상관공식의 합계 및 평균은 두 사용자가 평가한 항목에 대해서만 계산된다. 사용자가 수천 개의 항목 중에서 선택하여 평가할 수 있는 경우, 두 사용자가 평가한 항목의 중복이 많은 경우가 거의 없다. 따라서 계산 된 상관 계수 중 상당수는 적은 수의 관측치를 기반으로 하기 때문에, 계산 된 상관관계를 신뢰할 수 있는 유사도 척도로 간주할 수 없다.

둘째, 상관관계 접근법은 등급의 클래스에 대한 별도의 모델이 아닌, 사용자 간의 유사성에 대한 하나의 글로벌 모델을 유도한다. 이러한 접근 방식은 두 개의 사용자 프로필이 양의 상관관계인지, 상관관계가 없는지, 또는 음의 상관관계인지를 측정한다. 그러나 한 사용자가 제공한 평점은 두 사용자 프로필이 상관관계가 없더라도 다른 사용자의 평점에 대한 좋은 예측 요소가 될 수 있다. 예를 들어, 사용자 A의 긍정적 인 등급이 사용자 B의 부정적인 등급에 대한 완벽한 예측자인 경우가 있을 때, 사용자 A의 부정 등급은 사용자 B의 긍정적 인 등급을 의미하지는 않는다. 즉, 2개의 프로파일 사이의 상관관계가 0에 가까울 수도 있고, 잠재적으로 유용한 정보가 손실된다.

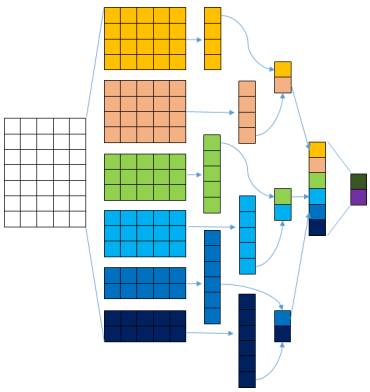
셋째, 두 명의 사용자가 등급이 서로 겹치면 유사 할 수 있다는 점이다. 즉 사용자가 공통 항목을 평가하지 않으면 해당 사용자 프로필은 상관관계를 가질 수 없다. 많은 도메인에서 평가할 수 있는 항목이 매우 많기 때문에 특히 시동 단계에서 많은 필터링 서비스에 방해가 될 수 있다. 그러나 사용자가 동일한 항목을 평가하지 않았다는 것을 발견하더라도 반드시 비슷한 생각을 하지는 않는다는 것은 아니다. 예를 들어, 사용자 A와 B는 높은 상관관계가 있고, B와 C도 서로 높은 상관관계가 있다. 이러한 관계는 사용자 A와 C 간의 유사성에 대한 정보도 제공한다. 하지만 사용자 A와 C가 공통 항목을 평가하지 않은 경우, 상관관계 기반 유사성 측정은 두 사용자 간의 관계를 발견하지 못한다. 이런 종류의 전이 유사성 관계를 발견 할 수 없다면 잠재적으로 유용한 정보가 손실된다.

2.4 Word2Vec

본 논문에서는 단어 임베딩 방법론 중 최근 많은 인기를 끌고 있는 Word2Vec을 활용한다. 단어 임베딩(Word Embedding)이란 고차원의 데이터를 그보다 낮은 차원으로 변환하면서 모든 데이터간의 관계가 성립되도록 처리하는 과정이다. 간단하게 정리하면 자연어 처리과정에서 단어를 벡터로 삽입하는 것이다. 다시 말해, 문자로 이루어진 단어를 숫자로 변환하는 것이다. 단어 자체를 아스키코드나 유니코드로 처리를 하여 사용해왔지만, 이것만으론 의미를 추론하기가 힘들었다. 단어 임베딩 기법에는 여러 모델이 있다. 초기모델인 NNLM모델과 RNNLM이 있고, 가장 최근에 발표되었고 현재 많은 인기와 동시에 많이 사용하는 Word2Vec모델이 있다. Word2Vec 모델을 사용하여 워드 임베딩을 진행하면 단어를 벡터화할 때 단어의 문맥적 의미를 보존하고 벡터로 바뀐 단어들은 코사인유사도와 같은 방식들로 그 거리를 잴 수 있고 단어사이의 거리가 가까울 경우 의미가 비슷한 단어끼리 벡터공간상에 맵핑되기 때문에 본 논문에서는 Word2Vec 모델을 활용하여 효과적으로 단어임베딩을 진행한다.[6]

2.5 합성곱신경망(Convolutional Neural Networks)

합성곱신경망은 이미지 분류와 컴퓨터 비전 시스템분야에서 활용이 되어왔지만, 최근 들어 자연어처리에 적용되기 시작하여 좋은 결과가 나오고 있다. 자연어 처리 분야에서는 이미지 픽셀대신, 행렬로 표현된 문장을 입력 값으로 받으며, 행렬의 각 열은 토큰, 일반적으로 단어에 대응된다. 각 행은 단어를 표현하는 벡터이다.



[그림2] 텍스트 처리를 위한 합성곱 신경망.

본 논문에서는 이러한 벡터를 Word2Vec을 활용하여 임베딩을 진행하여 입력하는데, 이것이 합성곱신경망에서의 이미지가 된다. 비전에서 필터는 이미지의 지역 조각을 슬라이딩하지만 자연어처리에서는 일반적으로 행렬의 전체 행을 슬라이딩 한다. 그러므로 필터의 너비는 보통 입력 행렬의 너비와 같고 높이 또는 지역의 크기는 변한다. 하지만 슬라이드 윈도우는 2~5단어가 일반적이다. 그림으로 설명하면 [그림2]와 같다.

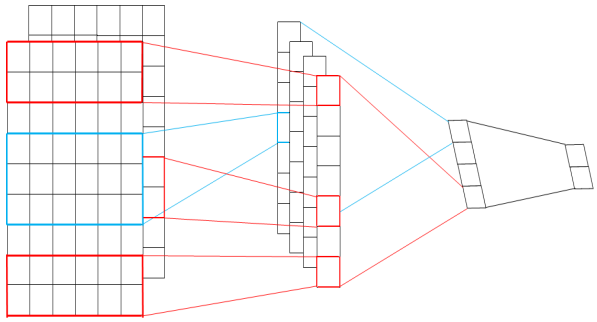
세 개의 필터 지역 크기는 2,3,4이다. 그리고 각각 두 개의 필터를 가지고 있다. 모든 필터는 문장 행렬에 대해서 합성곱을 수행하고 피쳐맵을 생성한다. 그 후 1-max-pooling이 전체 맵에 대해서 수행되어진다. 각 피쳐맵에서 가장 큰 수가 기록되게 된다. 그러므로 피쳐 벡터들은 모든 6개의 맵에서부터 만들어지고, 이 6개 피쳐 벡터들은 특성 벡터를 형성하기 위해 이어붙여진다. 마지막 소프트맥스 층은 이 특성 벡터를 입력으로 받아서 문장을 분류하는데 사용하게 된다.[7]

3. 합성곱신경망 기반 답변 추천 시스템

3장에서는 기존에 진행되었던 유사도기반 질의응답 시스템의 한계점, 그리고 이를 보완할 본 연구에서 사용할 Word2Vec과 합성곱신경망을 활용한 모델에 대해서 살펴본다.

3.1 Word2Vec과 합성곱신경망을 활용한 답변 추천

본 논문에서는 Yoon Kim이 제안한 Word2Vec과 합성곱신경망을 활용한 문장 구분 방법[8]을 활용하여 답변 추천 시스템을 만들어 사용한다. Yoon Kim이 제안한 모델의 구조는 [그림3]과 같다.



[그림3] Word2Vec + CNN 모델

위 모델은 우선 문장을 n*k 크기의 이미지로 변환을 한 뒤에 합성곱을 통해 피쳐맵을 만든다. 이렇게 만들어진 피쳐맵을 하나로 합친 뒤에, 완전 연결 층을 통해 결과 값을 만들어 낸다.

이 모델이 사용된 연구에서는 긍정과 부정으로 분류하는데 사용이 되었지만, 본 연구에서는 이렇게 나온 결과 값을 특성벡터로 활용하여 문장을 10개의 클래스로 분류하여 입력된 문장이 어떤 종류의 문의인지 구분하는데 사용한다. 이렇게 문의의 종류가 분류가 되면 해당 문의에 적합한 답변을 추천해줌으로써 상담원이 고객의 문의를 처리하는데 도움을 준다.

4. 실험

이 장에서는 앞서 소개한 5개의 유사도 기반 알고리즘을 활용한 답변 추천 시스템과 Word2Vec과 합성곱신경망을 사용한 답변 추천 시스템을 실험하여 그 결과를 비교한다.

4.1 실험 방법

데이터 셋은 [표1]처럼 400개의 데이터가 10개의 클래스로 각각 40개씩 나누어져있다. 데이터는 외식주문업체의 웹사이트에 등록되어 있는 FAQ 항목을 참고하고, 집단지성을 활용하여 유사 문장을 자체적으로 구축하였다. 이 데이터 셋을 3:1 비율로 학습에 사용될 트레이닝 데이터와 실험에 사용될 테스트 데이터로 나누었다.

No.	Input Data	Class
1	온라인 주문은 몇 시부터 가능하나요?	1
2	온라인 주문은 아침부터 가능하나요?	1
...	...	...
40	기프티콘 주문방법을 알려주세요.	2
41	제가 기프티콘을 가지고 있는데 어떻게 주문해야 되나요?	2
...	...	...
400	다른 매장으로 주문을 하고 싶어요.	10

[표1] : 실험에 사용된 데이터 셋

이러한 데이터 셋 앞에서 소개한 5개의 문장 유사도 기반 알고리즘과 이 논문에서 제시하는 Word2Vec과 CNN을 사용한 답변 추천 시스템을 사용해 실험을 진행하여 정확도를 계산한다. 정확도(Accuracy)는 다음과 같은 방법으로 산출하였다.

$$Accuracy = \frac{\text{클래스를 맞게 분류한 문장의 갯수}}{\text{전체 테스트 문장의 갯수}}$$

4.2 실험 결과 및 분석

Model	Accuracy
Pearson Coefficient	61%
Cosine Similarity Coefficient	63%
Tanimoto Coefficient	64%
Jaccard Coefficient	64%
Levenshtein Distance	57%
Word2Vec & CNN	84%

[표2]: 문장 유사도 알고리즘들과 Word2Vec & CNN 모델의 정확도 비교

위 표는 기존 문장 유사도 알고리즘을 사용한 모델과 본 논문에서 제시하는 모델의 정확도를 비교하여 보여준다. 실험결과, 유사도 기반의 알고리즘의 정확도는 약 60%를 살짝 넘어서는 수준이었으며, 레벤슈타인 거리는 57%로 가장 정확도가 낮았고, 연산에 걸린 시간도 약 6시간가량 소요되었다. 본 논문에서 제시하는 Word2Vec & CNN을 활용한 모델은 84%의 정확도를 보이면서, 다른 알고리즘들에 비해 약 20% 정도 향상된 정확도를 보여주고 있다.

5. 결론

본 논문에서는 Word2Vec과 합성곱신경망을 사용하여 답변 추천 시스템을 구현하여 기존의 문장유사도 기반 모델들과 비교하는 실험을 진행하였다.

실험을 통해, Word2Vec과 합성곱신경망을 사용한 답변 추천 시스템이 기존의 문장유사도 기반 모델에 비하여 20%정도의 정확도 향상이라는 결과물을 얻을 수 있었다.

본 연구에서는 Word2Vec을 사용함으로써 각 단어별 벡터를 학습시키는 것이 가능해짐으로써 수치화를 통한 단어의 개념적 구분이 가능해졌다. 여기에 합성곱신경망을 통해 좀 더 높은 정확도를 얻을 수 있었다.

향후 연구에서는 순환신경망을 사용한 질의응답 시스템을 연구하여 합성곱신경망을 사용한 방법과 비교 연구하는 것이 필요하다.

감사의 글

본 연구는 과학기술정보통신부 및 정보통신기술진흥센터의 SW중심대학지원사업의 연구결과로 수행되었음 (2016-0-00017)

참고문헌

[1] 박세영, 윤희근, 김다영, 김동건, 김민정, 신우석 “자연어처리와 정보검색을 이용한 질의응답 시스템,” 2015년 한국컴퓨터종합학술대회 논문집, 2015

[2] 이승우, 이근배 “유한패턴매칭을 이용한 자연어 질의응답 시스템”, 정보과학회지 22(4), 2004.4

[3] W.H.Gomaa and A.A.Fahmy “A Survey of Text Similarity Approaches,” International Journal of Computer Applications Volume 68-No.13, 2013

[4] V.U.Thompson, C.Panchev, M. Oakes “Performance Evaluation of Similarity Measures on Similar and Dissimilar Text Retrieval,” Knowledge Discovery, Knowledge Engineering and Knowledge Management (IC3K), 2015 7th International Joint Conference on, 2015

[5] S.Minmin, Q. Dongmei “The Application of Levenshtein Algorithm in the examination of the Question Bank Similarity,” 2016 International Conference on Robots & Intelligent System, 2016

[6] T. Mikolov, K. Chen, G. Corrado, J. Dean “Efficient Estimation of Word Representations in Vector Space,” arXiv:1301.3781, 2013

[7] Y.Zhang and B.C.Wallace “A Sensitivity Analysis of (and Practitioners' Guide to) Convolutional Neural Networks for Sentence Classification” arXiv:1510.03820, 2016.

[8] Yoon Kim “Convolutional Neural Networks for Sentence Classification,” 2014 Conference of Emirical Methods in Natural Language Processing(EMNLP), 2014.

트리 유사도: 상호운용성 평가도구

정성훈○, 배재학*
울산대학교 IT융합학부
gomgom9kr@outlook.kr, jhjbae@ulsan.ac.kr

Tree Similarity: Interoperability Evaluation Tool

Seonghoon Jeong○, Jae-Hak J. Bae*
School of IT Convergence, University of Ulsan

요 약

현대사회에 존재하는 다양한 시스템들이 병합될 때는 병합을 위해서 여러 가지 방법을 사용해 볼 수 있다. 이때 시스템의 성격에 따라 더 적절한 병합 방법론이 존재할 수 있지만, 어떤 방법이 해당 시스템을 통합하는데 더 적절한지를 판단하기는 쉽지 않다. 본 논문에서는 서로 다른 시스템을 통합할 때, 그 상호운용성을 평가하기 위한 수단으로 트리의 유사도를 측정하는 방안을 제시한다. 이렇게 측정된 유사도는 0 이상 1이하의 값을 가지며, 정확한 수치로 제시되기 때문에 서로 다른 통합 방법론을 평가하기 위한 계량적 근거로 사용될 수 있다. 다만 트리 구조로 나타낼 수 없는 일부 시스템들에 대해서는 적용할 수 없는 한계를 가진다.

주제어: 상호운용성, 트리, 유사도, 온톨로지

1. 서론

1.1. 연구배경

현대사회에서는 다양한 조직이 존재하고, 그 조직들은 고유의 시스템을 가지고 있다. 이러한 조직들은 필요에 따라서 조직들간의 병합이나 분리과정을 거치게 된다.

조직들이 병합할 때에는 그 조직들이 가지고 있는 고유한 시스템들 또한 같은 과정을 거치는데, 이러한 통합을 위한 방법론은 여러 가지가 존재한다. 온톨로지 분야를 예로들면 언어적인 방법(Lexical), 구조적인 방법(Structural), 인스턴스에 의한 방법(Instance-Based), 간접적으로 정렬하는 방법(Mediated), 의미론적 유사성에 의존하는 방법(Semantic Similarity)등의 방법이 존재하며[1], 이를 위한 소프트웨어도 다수 존재한다[2]. 흔히 사용되는 관계형 데이터베이스 또한 기본적인 join 연산을 통한 병합 외에도 객체지향적인 개념의 통합 설계방법론이 존재한다[3].

이렇듯 통합방법론이 한가지로 수렴하지 않고 여러 가지가 존재하는 것은, 각 방법론마다 고유의 강점과 약점이 존재하기 때문이다. 하지만 실제로 개별 시스템에 어떤 방법이 더 적합한지를 판단하는 것은 매우 어려운 일이다.

1.2. 연구의 필요성

개별 시스템에 더 적절한 방법론을 판단하는 것은 어려운 일이지만, 또한 매우 중요한 일이기 때문에 정부와 기업들 모두 해당 주제를 긴 시간동안 활발히 연구해왔다[4]. LISI, i-Score등의 평가도구들은 모두 이러한 노력의 결과로 고안된 것 들이다[4].

하지만 이러한 도구들은 고도의 수학적 지식이 필요할 뿐 아니라 복잡한 연산과정이 필요하기 때문에 비교적 많은 비용을 요구한다.

1.3. 연구목표

본 연구에서는 시스템 사이의 통합의 정도를 평가하는 방법으로 트리 유사도를 소개한다. 트리로 표현된 통합 시스템과 기존의 시스템간의 유사도를 절대적인 수치로 표현할 수 있다면, 이는 상호운용성을 평가하는 계량적인 근거로 사용될 수 있을 것이다. 또한 트리의 순회와 정수 연산과 같은 비교적 단순한 연산기법을 사용하여 비교적 적은 자원으로 평가에 필요한 연산을 가능하게 한다.

2. 관련 연구

2.1. 유물 분류 시스템 통합

이 논문은 서로 다른 박물관에서 사용하는 온톨로지로 표현된 유물 분류체계를 통합하는 방법[5]에 대해서 설명한다. 해당 논문에서는 확실히 동일한 것으로 간주할 수 있는 실제 유물에 대한 분류계(Classification)를 채널(Channel)로 두고 기존의 시스템들을 각 레벨별로 서로에 대한 제약식을 구하는 과정을 반복하는 것으로 두 박물관 분류체계간의 상호운용성을 확보하는 과정을 거친다.

* 교신저자

2.2. 온톨로지 통합 방법에 관한 연구

온톨로지 정렬(Alignment)[2]에는 언어적인 방법, 구조적인 방법, 인스턴스에 의한 방법, 간접적으로 정렬하는 방법, 의미론적 유사성에 의존하는 방법 등 복수의 온톨로지를 정렬하는 여러 방법론이 개발되어 있다. 각 방법은 고유의 장단점을 가지고 있다.

2.3. 상호운용성 평가에 관한 메타분석

상호운용성(Interoperability)은 사용되는 맥락에 따라서 여러 가지로 정의할 수 있다[4]. 이를 바탕으로 상호운용성의 종류와 평가를 위한 여러 접근법을 고찰해 볼 수 있다. 본 연구에서는 상호운용성을 "둘 이상의 시스템이 이질적인 네트워크에서 정보를 교환하고 사용하는 능력" [4][6]으로 정의한다.

3. 제약식과 온톨로지구조 표현

시스템을 병합할 때는 Information flow 이론[7]에 기초를 두면 유리한데, 이는 간단한 수학적 구조로 나타낼 수 있을 뿐 아니라 구조가 유연해서 큰 수정 없이 여러 경우에 일반적으로 적용할 수 있기 때문이다. Information flow 이론에서는 직접적으로 트리를 나타내기 보다는 제약식(Constraint)의 형식을 사용하여 논리적 구조를 표현한다. 표1에서는 제약식의 몇 가지 예시를 보여준다.

표 1 : 제약식의 특별한 경우 [7]

제약식	의미
α ⊢β	α가 논리적으로 β를 수반한다.
⊢α	α는 항상 참이다.
α ⊢	어떠한 토큰(token)도 α에 속하지 않는다.
⊢α,β	모든 토큰이 α또는β에 속한다.
α,β ⊢	α와 β는 상호배타적이다.

비록 이러한 제약식은 원래 논리관계를 설명하기 위한 도구이지만, 트리로 표현된 시스템 또한 묘사할 수 있다. 예를 들어 "⊢α"는 트리에서 α가 최상위에 존재하는 노드라는 것을 나타내는데 사용될 수 있고, "α ⊢β"는 α가 β의 자식노드를 나타내는 것으로 이해할 수 있다. 또한 "α,β ⊢"는 α와 β가 서로 다른 부모 노드에 포함되었다고 이해할 수 있을 것이다.

온톨로지 구조 또한 이와 유사하게 트리 구조로 표현이 가능하다. α의 입장에서 "α ⊢β"를 "subclass of β"로, "α,β ⊢"를 "disjoint with β"로 표현한 것으로 이해할 수 있다.

표 2는 2.1.논문에서 제시된 통합된 시스템을 나타내는 제약식들이고, 그림 1은 이를 바탕으로 구성한 트리의 모습이다.

표 2 : 통합된 박물관 시스템에서의 제약식 [5]

FCV ⊢BGV AV ⊢BGV MPV ⊢BGV BGV, WV ⊢	ADV ⊢DV ACV ⊢DV ADV ⊢BGV ACV ⊢BGV
CV = FCV MPV = MV AV = DV	J ⊢AV Z ⊢AV J = ADV Z = ACV

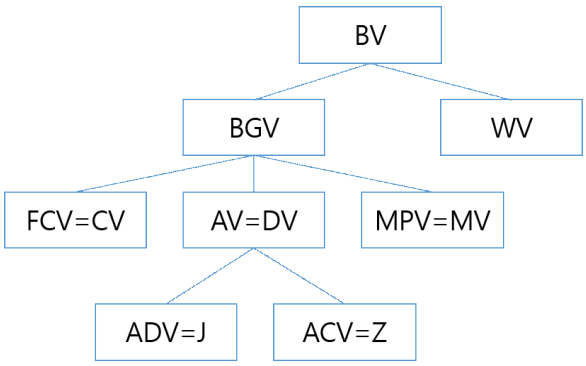


그림 1 : 통합된 청동기 유물 분류계

4. 구조적 유사도 평가

본 장에서는 2.1.논문에서의 트리를 사용하여 실제로 트리의 유사도를 구하고 상호운용성을 평가해본다. 아래의 그림 2와 그림 3은 통합되기 전의 트리를 나타낸다.

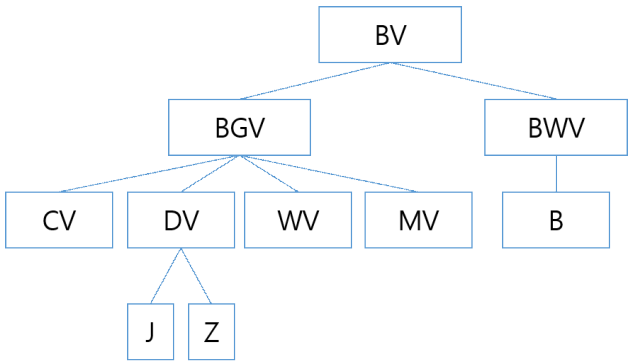


그림 2 : 청동기에 대한 유물 분류계 A [5]

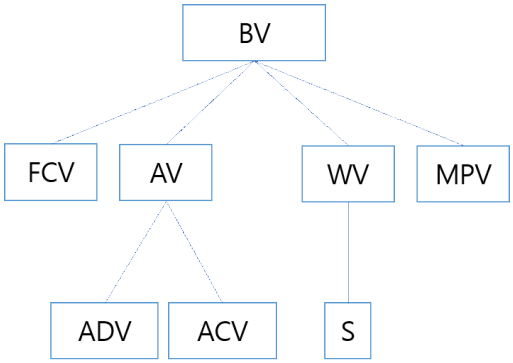


그림 3 : 청동기에 대한 유물 분류계 B [5]

이하의 과정에서 위 분류계 A와 통합 분류계, 또 분류계 B와 통합 분류계 간의 유사도를 구할 것이다. 계산과정은 최종적으로 유사도가 0이상 1이하의 값을 가지도록 고안되었으며, 두 트리가 조금도 유사하지 않을 경우 0, 완전히 동일한 경우 1의 값을 가지게 된다.

실제 계산은 영향을 주는 요인에 대해서 점수를 할당하고 얻은 수치의 합을 구하고, 획득할 수 있는 최대 점수를 나누는 것으로 이루어진다.

4.1. 동일한 노드의 존재유무 확인

트리의 유사도를 계산할 때 가장 먼저 고려해야 할 것은 비교하는 트리 사이에 같은 노드가 얼마나 존재하는가 하는 점이다.

이때 점수는 각 노드에 대해서, 비교하는 트리에 동일한 노드가 존재한다면 1점, 존재하지 않는다면 0점을 부여하고, 획득할 수 있는 최대점수는 비교하는 두 트리의 노드의 개수의 합이 될 것이다.

4.2. 동일한 부모 노드 보유 여부

비교대상 트리에 동일한 노드가 존재하는 것을 확인했을 때 양쪽 노드의 부모 노드를 확인하여 부모가 동일한 노드인지 체크하여 점수를 부여할 수 있다. 이는 개별 노드의 입장에서 보면 단순히 부모를 체크하는 작업이지만, 전체적으로 보면 트리 내에서 노드의 상대위치를 파악하는 효과를 얻을 수 있다.

상대위치를 파악하는 것은 자식을 비교하는 것으로도 가능하지만, 일반적으로 부모 노드는 최상위 노드를 제외하고 전부 보유하고 있으며, 특별한 경우가 아니면 자식노드는 다수가 존재할 수 있으나, 부모 노드는 오직 하나만 존재하기 때문에 계산하기에 용의하다.

점수는 비교트리에 동일한 노드가 존재하고 그 노드의 부모 노드까지 동일하면 1점, 노드가 존재하지 않거나 존재하지만 부모 노드가 다르다면 0점을 부여하고, 획득할 수 있는 최대점수는 위의 과정과 마찬가지로 비교하는 두 트리의 노드의 개수의 합이 될 것이다.

4.3. 유사도 계산

편의상 위 두 과정을 별개의 과정으로 서술하였지만, 구현의 난이도나 효율성 측면에서 보면 두 과정을 동시에 두고 보는 것이 더 합리적이다. 따라서 트리를 탐색하면서 두 점수를 동시에 파악하고, (두 트리의 노드의 개수 합 * 2)를 나누는 식으로 계산을 한다.

이 때 4.1.과 4.2.에서 모두 동일한 노드의 존재유무를 체크하기 때문에 해당항목이 과대평가 되는 것으로 보일 수 있다. 그러나 동일한 노드가 없다면 총 2점 중 한쪽 트리에서만 0점을 받게 되고, 동일한 노드가 존재하지만 부모만 다르다면 양쪽 트리에서 동일한 노드에 대해 각각 1점씩만 부여하게 된다. 전체적으로 보면 두 경우 모두 2점을 감점하는 것이 되고, 따라서 둘에 같은 가중치를 주는 것으로 볼 수 있다.

트리를 탐색하는 방법은 중복 없이 모든 노드를 탐색하는 알고리즘이라면 무엇이든 상관없다. 그림4 에서는 BFS(너비 우선 탐색)[8]알고리즘을 사용하는 계산과정을 개략적으로 보여준다.

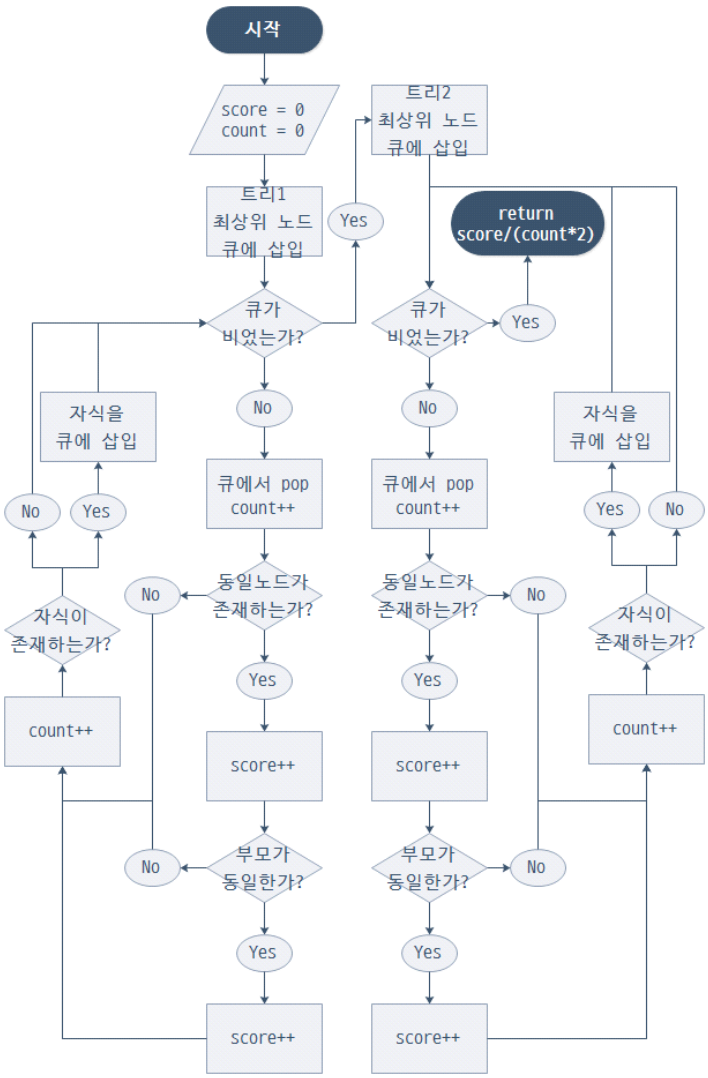


그림 4 : 유사도 계산과정

4.4. 유사도 평가

지금까지 보인 계산과정을 토대로, 본문에서 언급된 통합된 박물관 트리와 기존의 청동기 유물 분류 트리의 유사도를 비교해보면 표3 과 같다.

표 3 : 통합 트리와 기존 트리의 유사도 비교 결과

트리 A / 통합 트리	트리 B / 통합 트리
0.8889	0.6875

비교 결과 트리 A는 약 89%, 트리 B는 약 68%정도의 유사도를 가지므로 위 예시에서 병합된 트리는 기존 트리들의 상호운용성을 비교적 잘 반영하고 있다고 할 수 있다. 만약 2.1. 논문에서 제안한 것과 다른 병합 방법을 사용하고자 한다면, 위의 결과보다 더 높은 유사도를 보이는 방법을 찾아야 할 것이다.

5. 의미론적 유사도 평가

앞서 살펴본 방법은 구조상의 유사도를 평가할 수 있지만, 만약 트리 구조 자체와 상관없는 기능적인 역할이 시스템에 존재했을 경우 오직 구조만을 평가하고 의미론적인 유사도는 평가에 반영하기 힘들게 된다. 이러한 경우의 예시로는 온톨로지의 속성(Property), 객체지향 프로그램에서의 함수(Method)등을 들 수 있다. 이러한 것들은 트리의 구조와는 별개로 평가해줄 필요가 있다. 이러한 기능들은 기본적으로 함수의 꼴을 취할 수 있기 때문에 동일한 입력(Parameter)에 대해서 유사한 출력(Return)을 가지는 정도를 측정하는 것으로 의미론적인 유사도의 부재를 보완할 수 있다.

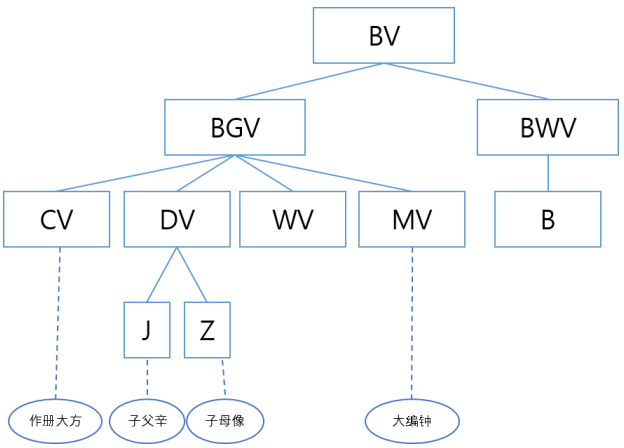


그림 5 : 인스턴스를 포함한 유물 분류계 A [5]

위에서 예로 들었던 유물 분류 시스템에서 이러한 예를 찾는다면 유물의 검색기능을 예로 들 수 있다. 이 기능을 함수 꼴로 생각한다면, 입력은 실재 존재하는 유물 인스턴스(Instance)가 되고, 출력은 그 유물이 속해있는 클래스(Class)가 될 것이다.

표 4 분류계A에서의 getClassificiation()

호출 형태	결과 값
getClassification(作册大方)	CV
getClassification(子父辛)	J
getClassification(子母像)	Z
getClassification(大編鐘)	MV

표 5 통합된 분류계에서의 getClassification()

호출 형태	결과 값
getClassification(作册大方)	FCV = **CV**
getClassification(子父辛)	ADV = **J**
getClassification(子母像)	ACV = **Z**
getClassification(大編鐘)	MPV = **MV**

표4와 표5의 결과를 비교하면 위의 예시는 의미론적으로 유의미한 차이가 없는 것을 볼 수 있다. 이 예시 에서는 분류계A 에 대해서만 비교를 해봤지만 분류계B에 대해서도 동일한 결과가 나오는데, 이는 예시가 의미론적 유사성에 기반을 두고 병합을 했기 때문이다.

만약 함수의 실행결과에 차이가 있는 경우라면, 구조적 유사도를 구할 때와 유사한 방법으로 그 차이를 전체 정의역과 치역의 가짓수로 나누어서 0에서 1사이의 값으로 나타낼 수 있을 것이다.

6. 결론

본 연구에서는 통합된 시스템과 기존의 시스템들의 유사도를 측정하는 것으로 두 트리간의 상호운용성을 수치적으로 평가하는 방법을 제시한다.

본문에서는 동일 노드의 존재유무와 트리 내에서의 상대적 위치(동일한 부모 노드)만을 고려하였고, 이 두 요소가 같은 중요도를 가진다고 가정했지만, 실제로 어떤 요소에 더 가중치를 둬야할지, 또한 어떤 변수를 고려해야 할지 여부는 시스템의 성격에 따라 달라질 수 있다. 예를 들어, 군대조직과 같이 실제 수행하는 업무의 내용보다 계급의 높낮이가 더 중요한 조직을 병합할 때는 부모 노드의 비교를 통해 상대위치를 이용하기 보다는 차수(Degree)를 비교해서 절대위치를 비교하는 것이 더 합리적일 것이다.

본 연구에서는 트리의 유사도를 평가도구로 사용하기 때문에 일부 트리 구조로 나타낼 수 없는 시스템을 평가할 때는 적용할 수 없으며, 결국 평가를 위한 개별 항목의 가중치는 일정부분 직관에 의존할 필요가 있다는 한계를 가진다. 이러한 한계들은 후속연구에서 보충되어야 할 것이다.

참고문헌

[1] Dargie, Waltenegus, ed. "Context-aware computing and self-managing systems, CRC Press", 2009.
[2] Natalya F. Noy, Mark A. Musen,"The PROMPT Suite: Interactive Tools For Ontology Merging And Mapping,"International Journal of Human-Computer Studies, Vol. 59, No. 6, pp. 983-1024, December 2003.
[3] 주경수, 조도형, "관계형 데이터베이스 응용시스템을 위한 통합 설계방법론 개발" 한국컴퓨터정보학회 논문지 , 16(11), 25-34, 2011.
[4] Ford, Thomas C., et al. “Survey on Interoperability Measurement”, AIR FORCE INST OF TECH WRIGHT-PATTERSON AFB OH, 2007.
[5] Hongzhe Liu, Hong Bao and Junkang Feng, “IF Based Semantic Interoperability for Distributed Digital Museums”, COMPUTING AND INFORMATION SYSTEMS, 10, 1, 2006.
[6] Morris, E., et al., “System of Systems Interoperability (SoSI): Final Report,” Carnegie-Mellon University-Software Engineering Institute, Pittsburgh, PA, Tech. Rep. CMU/SEI-2004-TR-004, Apr. 2004.
[7] Barwise J and Seligman J, Information Flow: the Logic of Distributed Systems, Cambridge University Press, Vol. 44, 1997.
[8] E. F. Moore, "Shortest path through a maze" in Annals of the Computation Laboratory of Harvard University, Mass., Cambridge:Harvard University Press, vol. 30, pp. 285-292, 1959.

L2 영어 학습자들의 연어 사용 능숙도와 텍스트 질 사이의 수치화

권준혁0, 김재준, 김유래, 박명관, 송상헌

동국대학교, 인천대학교

Kyunjh1272@gmail.com

Quantifying L2ers' phraseological competence and text quality in L2 English writing

Junhyeok Kwon0, Jaejun Kim, Yoolae Kim, Myung-Kwan Park (Dongguk University)

Sanghoun Song (Incheon National University)

Abstract

On the basis of studies that show multi-word combinations, that is the field of phraseology, this study aims to examine relationship between the quality of text and phraseological competence in L2 English writing, following Yves Bestegen et al. (2014). Using two different association scores, t-score and Mutual Information(MI), which are opposite ways of measuring phraseological competence, in terms of scoring frequency and infrequency, bigrams from L2 writers' text scored based on a reference corpus, GloWbE (Corpus of Global Web based English). On a cross-sectional approach, we propose that the quality of the essays and the mean MI score of the bigram extracted from YELC, Yonsei English Learner Corpus, correlated to each other. The negative scores of bigrams are also correlated with the quality of the essays in the way that these bigrams are absent from the reference corpus, that is mostly ungrammatical. It indicates that increase in the proportion of the negative scored bigrams debases the quality of essays. The conclusion shows the quality of the essays scored by MI and t-score on cross-sectional approach, and application to teaching method and assessment for second language writing proficiency.

Key words: phraseological competence, corpus, n-grams, Collgrams, quality of essay

1. L2ers' academic writing based on phraseology

Traditionally, the focus in second language acquisition has been on how L2 learners acquire morphological and grammatical knowledge rather than other levels of language. Although the focus on grammatical knowledge for l2 leaners is pervasive, recently corpus linguistic research has taken lexis to a central role in second language acquisition. Since corpus linguistics is mainly lexical, which is based on corpora from real world text, it is easy to deal with lexical items and a sequence of lexico-grammatical patterns.

Lewis' (1993) idea that "language consists of grammaticalized lexis, not lexicalized grammar" has shed light on lexicalization in language teaching. The notion of lexis is basically phraseological, which is on the premise that lexis is not just the study of single word, but phraseology is "the study of the structure, meaning and use of word combinations" (Cowie, 1994). Language production, including writing and speaking, is largely relies on pre-patterned phrases which are prefabricated in L1; therefore, the role of phraseology is quite essential for L2 learners.

L2 writers generally tend to use only limited repertoire of collocations that they mastered, but very little amount of native-like lexical bundles. "Failure to use native-like formulaic sequences is on factor in making their writing feel nonnative" (Li and Schmitt, 2009). As pointed out above, recently the focus on formulaic sequences in SLA has been issues. Formulaic sequences are contiguous word sequences which are stored and retrieved as one unit from memory, and they are rather than being generated by single words and their structures, considered as whole sequence at the time of use. Coxhead and Byrd (2007) enumerate three reasons for a focus on formulaic sequences in L2 academic writing: (1) for students, using ready-made formulaic sequences is easy; (2) formulaic sequences define standards of fluent academic writing; (3) formulaic sequences are easier to detect on the basis of corpus data. Besides these three reasons, using corpus data makes it possible to quantify its data and help search significant formulaic sequences for academic writing, and also those significant formulaic sequences are reliable in terms of 'usage based learning'.

2. N-gram method and problems with using frequent n-grams

N-grams are continuous word sequences of n items from a given text. A size of n-grams increases by one, and each of n-grams are referred to uni-grams, bi-grams, tri-grams, four-grams, and so on. Extracting n-grams from corpus data is one of the effective ways to detect formulaic sequences. From an extracted data, counting the same word combinations of n-grams shows how some kinds of n-grams are used in a text. However, focuses on the frequency of those n-grams only pay attention to absolute frequency, not the degree of relation within the given n-grams; for instance, bigrams 'of the' or 'it is' could be one of the most frequent bigrams from any texts in the real world. Both 'of' and 'the' or 'it' and 'is' are very frequent words themselves, but this high frequency does not present evidence for relevant phraseological status.

3. Mutual Information, t-score, and CollGrams

Since high frequency of n-grams only shows the absolute frequency of n-grams, frequency does not detect significant n-grams. To figure out the association within each n-grams, all of the n-grams extracted from a given text are assigned each MI (Mutual Information) and t-score, and each of the n-grams formulate CollGrams, which form the basis of evidence for phraseological association within the n-grams.

MI is a measurement for strength of association, which originated from information theory. The higher MI score each n-gram gets, the more infrequent words sequences appear within the n-grams. MI accords with the log transformed ratio between observed frequency of n-grams, including not only single words and word sequences, and its expected frequency. Following is the formula of MI from Church & Hanks, 1990:

$$Mutual\ Information = log2\frac{Freq\ xy}{(Freq\ x\ \times Freq\ y)/N}$$

(Church & Hanks, 1990)

On the other hand, t-score represents certainty of n-grams. The higher t-score each n-grams gets, the more frequent words sequences appear within the n-grams. t-score is the expected frequency of the square root of the observed frequency, including frequency of single words and word sequences. Following formula is from Church et al., 1991:

$$t \approx \frac{Freq(xy) - \frac{1}{N}Freq(x)Freq(y)}{\sqrt{Freq(xy)}}$$

(Church et al., 1991)

Each of the n-grams assigned MI and t-score forms CollGrams, which show their collocational status. This CollGram technique solve the problem of using frequent n-grams.

4. Previous study: Yves Bestegen and Sylviane Granger (2014)

In their study, they aim to assess L2 text quality in terms of phraseological competence. Using Collgrams, they calculated each bigram from the Michigan State University corpus of English based on the reference corpus, COCA (Corpus of Contemporary American English). The students who participated for making MSU corpus of English wrote essays three times from the beginning of a semester to the end of the semester, which made this study possible to search on both longitudinal and cross-sectional approach.

The results of the longitudinal study showed a decrease in mean t-score, which explains that L2 learners acquire more complex collocations and idioms instead of low-level binary chunks.

On the other hand, the results of the cross-sectional study presented that only MI and absent category from the reference corpus are statistically significant in terms of correlation between the quality of the texts and the mean scores. They assumed that bigrams consist of low frequency words might be more noticeable to raters who judged the texts, and positively influence their judgement.

5. Methodological approach and data

The technique, CollGram, aims to assign to each bigram association scores computed on the basis of a reference corpus. If a bigram extracted from a learner corpus does not exist in a reference corpus, it is classified in an absent category. CollGrams consist of three measures: the mean MI score, the mean t-score, and the proportion of absent bigrams from the reference corpus. Steps for profiling CollGrams are followed: (1) both the learner and the reference corpus are tokenized except for punctuation marks; (2) all bigrams extracted from the learner corpus; (3) each bigram extracted from the learner corpus are looked up in the reference corpus; (4) the bigrams existed in the reference corpus are assigned their MI and t-score; (5) if the bigrams do not exist in the reference corpus, they are

classified in the absent category.

In this study, Yonsei English Learner Corpus (YELC) is used for the learner corpus. YLEC is collected from January 2011 to February 2011. YELC where 3,286 freshmen at Yonsei University participated contains 1,081,280 words.

Details of YELC learner corpus	Beginner	Intermediate	Advanced
Number of texts	910	2256	120
Total number of words	233661	798907	48712
Number of words per proficiency (mean)	265.77	354.12	405.93
Rating	A1/A1+/A2	B1/B1+/B2	B2+/C1/C2

(*Table 1. details of YELC*)

All the texts from YELC are rated from A1 to C2, and for the study we classified the nine different ranks into three proficiencies: beginner, intermediate, and advanced. Total number of words from each proficiency shows that intermediate-level contains the largest words because more than the half of the texts rated from B1 to B2, but average words per each proficiency present that texts rated advanced-level have the largest words; 405.93 words per a text.

As to the reference corpus, we opted for Global Web based English Corpus (GloWbE), which has more than 1.9 billion words. About 60% of the corpus are from blogs, which means it is very informal (one of the reasons we opted for the corpus as a reference corpus). From the varieties of English in the corpus, we chose U.S. English since most of the Korean students are exposed to American English in Korea.

6. Results: highest-scoring bigrams and the absent category

Table below lists the top 20 highest-scoring bigrams in YELC learner corpus, advanced-level, beginner-level, respectively. The left-hand side of the table are sorted in decreasing order of the MI score. Many of the bigrams with high MI scores consist of either infrequent verbs and definite article 'the', or infrequent verbs and infinitive to (or prepositions). On the other hand, the right-hand side of the table shows the bigrams with high t-score which are composed of frequent prepositions and definite article 'the', or pronouns and verbs.

Top-scoring bigrams	MI	Top-scoring bigrams	t	Top-scoring bigrams	MI	Top-scoring Bigrams	t
circulates_the	40.4014	of_the	128.5095	refer_to	39.3586	i_think	62.4293
outweigh_the	40.0795	on_the	121.813	ought_to	39.3285	i_am	62.2972
throughout_the	39.7033	to_be	92.3940	disposing_of	39.2940	this_is	61.7925
violates_the	39.6424	on_the	81.7483	consequences_of	39.2940	at_the	61.2471
outweighs_the	39.5941	it_is	73.8763	opposed_to	39.2940	will_be	60.9732
according_to	39.5681	if_you	73.8233	tries_to	39.2688	i_have	60.6789
tends_to	39.5496	is_a	64.9545	refuse_to	39.2576	want_to	59.2859
able_to	39.5468	it_was	64.5843	due_to	39.2533	the_same	59.2609
subjeted_to	39.5289	for_the	63.5135	intend_to	39.2523	as_a	58.9083
tend_to	39.5020	i_do	62.75	distorting_the	39.2467	going_to	58.5038

(*Table2. top 20 highest-scoring bigrams in advanced-level*)

Top-scoring bigrams	MI	Top-scoring bigrams	t	Top-scoring bigrams	MI	Top-scoring bigrams	t
distorts_the	40.4014	of_the	128.5095	coffees_and	39.4517	i_think	62.4293
embodies_the	39.9420	in_the	121.813	embarrased_and	39.4517	i_am	62.2972
displace_the	39.8164	to_be	92.3940	drugging_and	39.4517	this_is	61.7925
pollute_the	39.8164	on_the	81.7483	willing_to	39.4437	at_the	61.2471
according_to	39.5681	it_is	73.8763	accustomed_to	39.4357	will_be	60.9732
able_to	39.5468	if_you	73.8233	supposed_to	39.4219	i_have	60.6789
unable_to	39.5200	is_a	64.9545	contaminate_the	39.4014	want_to	59.2859
tend_to	39.5020	it_was	64.5843	diminishes_the	39.4014	the_same	59.2609
lessen_the	39.4628	for_the	63.5135	unify_the	39.4014	as_a	58.9083
relating_to	39.4565	i_do	62.75	trying_to	39.3967	going_to	58.5038

(*Table3. top 20 highest-scoring bigrams in advanced-level*)

The other category that needs to be analyzed closely in terms of pedagogy is the absent category in the reference corpus. Since the bigrams in the learner corpus are absent in the reference corpus, this shows what types of erroneous bigrams are generated from L2 writers. The proportion of the absent category also represents quality of the given texts in accordance with CollGrams. In many cases, L2 writers made mistakes using two determiners at one time, and also, they made many erroneous bigrams with inappropriate prepositions. Following table shows randomly selected erroneous bigrams which are absent in the reference corpus:

W1W2	sentences	Error
a this	No I disagree a this opinion because that is a important thing	article
a students	Physical punishment can be a good medicine for a students who don't have a strong will	article
went Japan	I went Japan with my 4 best friends .	preposition
is need	I think physical punishment is need for them	past participle
animals	I think animals be used in medical experiments is need.	be verb
not use	so I want drivers not use their phone while driving.	to infinitive
doesn't free	Yet my country Korea doesn't free from fright that includes North Korea's threat.	negation

(*Table4. randomly selected erroneous bigrams in the absent category*)

7. discussion

Mean score	Beginner	Advanced
MI(Mean)	25.8309	26.4923
MI(SD)	6.5826	6.7566
t(Mean)	0.4526	2.6722
t(SD)	14.3550	12.4856
Number of absent category	42459	8657

(*Table5. Difference between Beginner and Advanced learners*)

The increase in the mean MI and the mean t-score indicates that the higher score L2ers get, the higher quality of the texts L2ers write. The decrease in the proportion of the absent category also presents that advanced-level L2ers made less erroneous formulaic sequences. These three indices constitute CollGrams, and it shows that both of infrequent and frequent formulaic sequences get higher score on each measurement.

The way to calculate CollGrams of the learner corpus based on the large reference corpus is a text-external measure. The text-external measure leads to operationalize formulaicity of L2 writing, which is helpful to make L2 writers more native-like and also for assessment.

For developing this study, we need to adopt statistical inferences to analyze the given results of CollGrams, and also the 'analysis of variance' (ANOVA) to measure the correlation between given results and rated text quality.

Reference

1. Bestgen, Yves, and Sylviane Granger. "Quantifying the development of phraseological competence in L2 English writing: An automated approach." *Journal of Second Language Writing* 26 (2014): 28-41.

2. Church, Kenneth Ward, and Patrick Hanks. "Word association norms, mutual information, and lexicography." *Computational linguistics* 16.1 (1990): 22-29.

3. Church, Kenneth, et al. "Using statistics in lexical analysis." *Lexical acquisition: exploiting on-line resources to build a lexicon* 115 (1991): 164.

4. Cowie, A. P. "Applied linguistics: lexicology." *Encyclopedia of Language and Linguistics. Pergamon, Oxford* (1994): 177-180.

5. Coxhead, Averil, and Pat Byrd. "Preparing writing teachers to teach the vocabulary and grammar of academic prose." *Journal of second language writing* 16.3 (2007): 129-147.

6. Lewis, Michael, and Cherry Gough. *Implementing the lexical approach: Putting theory into practice*. Vol. 3. No. 1. Hove: Language Teaching Publications, 1997.

7. Li, Jie, and Norbert Schmitt. "The acquisition of lexical phrases in academic writing: A longitudinal case study." *Journal of Second Language Writing* 18.2 (2009): 85-102.

8. Rhee, S., and C. Jung. "Yonsei English learner corpus (YELC)." *Proceedings of the First Yonsei English Corpus Symposium*. 2012.

MTRNN을 이용한 한국어 대화 모델 생성

신창욱○, 차정원
창원대학교
{papower1, jcha}@changwon.ac.kr

Korean Dialogue Modeling using MTRNN

Chang-Uk Shin○, Jeong-Won Cha
Changwon National University

요 약

본 논문에서는 Multi-layer sequence-to-sequence 구조를 이용해 한국어 대화 시스템을 개발하였다. sequence-to-sequence는 RNN 혹은 그 변형 네트워크에 데이터를 입력하고, 입력이 완료된 후의 은닉층의 embedding에 기반해 출력열을 생성한다. 우리는 sequence-to-sequence로 입력된 발화에 대해 출력 발화를 내어주는 대화 모델을 학습하였고, 그 성능을 측정하였다. RNN에 대해서는 약 80만 발화를, MTRNN에 대해서는 5만 발화를 학습하고 평가하였다. 모델의 결과로 나타난 발화들을 정리하고 분석하였다.

주제어: sequence-to-sequence, 대화 모델, LSTM, MTRNN

1. 서론

대화 시스템은 대화의 기록을 유지하며, 입력된 사용자의 발화에 대해 적절한 응답을 내어주는 시스템이다.

대화 시스템에서 가장 중요한 모듈은 주어진 대화 기록과 입력된 사용자의 발화에 대하여 시스템의 출력 발화를 결정하는 모듈이라고 볼 수 있다. 우리는 그것을 대화 모델이라 부른다.

sequence-to-sequence 등의 end-to-end 구조를 이용하여 자연언어처리의 문제를 해결하려는 시도가 종종 있어왔다. 대화 시스템에서는 사용자의 발화 처리, 대화 기록 관리, 시스템 발화 생성을 하나의 모델로 수행하는 방식이 이에 해당한다. 이러한 end-to-end 시스템은 기존에 연구된 다단계 시스템에 비해 연구자의 노력과 시간이 적게 소요됨에도 불구하고 높은 성능을 보여주고 있어 여러 분야에서 시도되고 있다.

우리는 sequence-to-sequence 구조로 한국어 대화 모델을 학습하고 그 결과를 분석하였다. 특히, Recurrent unit으로 LSTM과 MTRNN을 비교하여 분석하였다.

2. 관련 연구

여러 연구자들이 end-to-end 구조로 자연어처리 문제들을 해결하기 위해 연구를 진행한 바 있다. sequence-to-sequence 구조는 순서를 갖는 데이터를 입력하여 입력열 전체에 대한 표현을 획득하고, 그것에 기반하여 출력열을 생성하는 구조를 취하고 있다. 이 방식이 기존 자연어처리 분야의 여러 문제에 적용하기 적합하여, 형태소 분석 등 여러 분야에 적용되었다.

[1, 2, 3]에서는 sequence-to-sequence 구조를 이용해 형태소 분석 및 품사 태깅을 시도한 바 있다. 음절 단위, 또는 어절 단위로 인코딩된 입력을 sequence-to-sequence 모델에 입력하고, 그 출력을 형태소 분석 및 품사 부착의 결과물로써 사용한다.

[4]에서는 sequence-to-sequence 구조로 구구조 구문분석을 수행하였다. 형태소 분석된 입력 문장을 음절 또는 형태소 단위로 입력받아, 구구조 구문분석 결과를 토큰 단위로 출력하도록 학습하였다.

대화 시스템과 모델링에서 관련된 이전 연구로, 많은 양의 발화 입-출력 쌍을 수집하고 입력되는 발화에 매치된 출력 발화는 내어주는 방식의 연구가 진행되었다. [5]에서는 TF-IDF와 단어 임베딩을 대화 매치에 사용해 MRR 93.9%를 달성하였다.

3. 제안 방법

우리는 Multi-layer sequence-to-sequence 구조를 이용해 대화 모델을 학습하였다. Recurrent Unit으로 LSTM[6]과 MTRNN[7]의 변형을 학습하였고, 그 결과를 비교하였다.

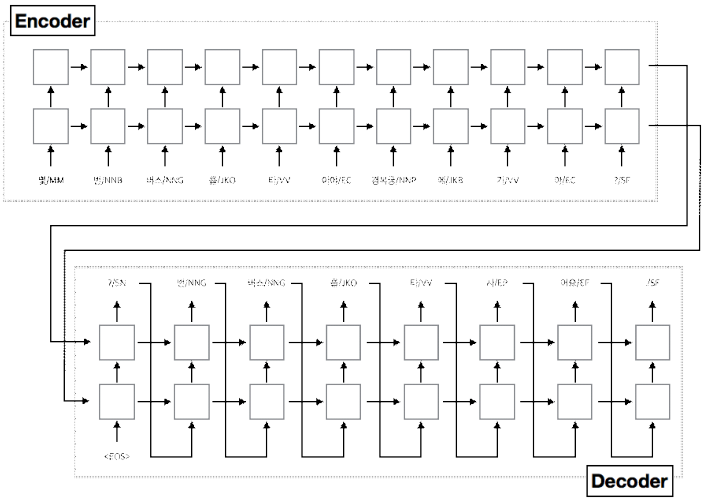
[그림 1]은 sequence-to-sequence의 구조를 설명한다. 그림에서 입력열을 분산 표현으로 생성하는 단계를 encode, 그것을 수행하는 네트워크를 encoder라 칭한다. encoder는 매 입력이 주어질 때마다 다음 식으로 RNN variant의 hidden state를 업데이트 한다.

$$h_t = f(h_{t-1}, x_t) \quad (1)$$

식1에서 f는 non-linear activation 함수이다. 마지막 토큰까지 입력이 완료되었을 때의 hidden state를 sequence 전체에 대한 분산 표현으로 간주한다. 그리고 해당 분산 표현을 최초의 state로 하는 decoder를 동작시켜 출력 시퀀스를 생성한다.

$$h_t = f(h_t, y_{t-1}) \quad (2)$$

위 식2는 decoder에서의 매 타임 스탭 t에서 이전 스탭의 hidden state h_t와 출력 y_{t-1}로 이번 타임 스탭의 hidden state를 생성함을 이야기한다. [그림 1]에서도



[그림 1] multi-layer sequence-to-sequence

이전의 출력 토큰이 다음 스텝의 입력으로 취해지는 것을 확인할 수 있다.

sequence-to-sequence는 여러 RNN variant로 구성할 수 있는데, 본 논문에서는 그 중 LSTM과 MTRNN으로 실험을 진행하였다.

LSTM은 RNN에서 발생하는 그래디언트 소실(vanishing gradient) 문제를 해결한 변형으로 여러 문제에서 RNN 대신에 주로 사용되고 있다. RNN에서는 하나의 state를 관리하는 반면, LSTM은 세 개의 gate를 이용해 두 개의 state를 관리하고 있다. 그 첫 번째는 아래 식의 c이고 cell state라 불리운다. 두 번째 state는 아래의 h이고 hidden state이다. 식의 $\odot$는 Hadamard product이다.

$$f_t = \sigma(W_{xh\_f}x_t + W_{hh\_f}h_{t-1} + b_{h\_f}) \quad (3)$$

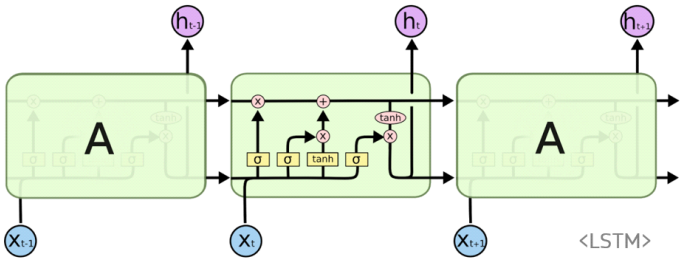
$$i_t = \sigma(W_{xh\_i}x_t + W_{hh\_i}h_{t-1} + b_{h\_i}) \quad (4)$$

$$o_t = \sigma(W_{xh\_o}x_t + W_{hh\_o}h_{t-1} + b_{h\_o}) \quad (5)$$

$$g_t = \tanh(W_{xh\_g}x_t + W_{hh\_g}h_{t-1} + b_{h\_g}) \quad (6)$$

$$c_t = f_t \odot c_{t-1} + i_t \odot g_t \quad (7)$$

$$h_t = \tanh(c_t) \odot o_t \quad (8)$$



[그림 2] LSTM 구조

MTRNN은 Recurrent Node의 상태 업데이트 식을 새로이 정의하고, 그것을 파라미터 τ로 조절할 수 있도록 하였다. 같은 층의 노드를 그룹화하고, 그룹마다 τ를 달리 설정하여 특정 그룹은 빠르게, 특정 그룹은 느리게 업데이트되도록 설정하였다.

우리는 MTRNN τ를 층 내에 그룹에 설정하지 않고, 2개의 층에 각각 다른 τ를 설정하여 변형을 시도하였다. 이렇게 하면 층마다 다른 state representation을 갖게 될 것이다.

$$u_{i,t+1} = (1 - \frac{1}{\tau_i})u_{i,t} + \frac{1}{\tau_i}\left[\sum_{k \in N} w_{jk}x_{k,t}\right] \quad (9)$$

위 식9에서 τ를 layer마다 달리 설정해 줌으로써 multi-layer에서 multiple time-scale을 작성할 수 있다. 식에서 w는 weight matrix, x는 입력 혹은 이전 layer의 state, N은 입력 차원 혹은 이전 layer의 노드의 수, $u_{i,t}$는 시간 t에서 i번째 layer의 state이다.

4. 실험

4. 1. 실험 설정

sequence-to-sequence로 대화 모델링을 수행한다. 어떤 사용자의 입력에 대하여, 그것의 응답에 해당하는 발화 출력을 목표로 설정한다. 평가로는 평가 데이터셋의 입력 발화를 시스템에 입력하고, 그 출력물을 평가 데이터셋의 출력 발화와 비교하여 성능을 도출한다.

학습에 사용한 코퍼스는 직접 작성하였고, 일상 대화 도메인의 코퍼스이다. 코퍼스에 대한 정보는 [표 1]에 정리하였다. 하나의 입력 발화에 대해 여러 출력 발화가 부착되어 있는 형태이다.

[표 1] 학습 코퍼스의 통계 정보

구분	수량	단위
입력 발화 수	146,276	발화
출력 발화 수	1,123,902	발화
입력 발화 내 발화당 형태소 수	4.50	형태소/발화
출력 발화 내 발화당 형태소 수	9.42	형태소/발화

코퍼스가 하나의 입력 발화에 대해 여러 출력 발화가 부착된 코퍼스이므로, 전처리 과정이 다소 필요하다. 우리는 먼저 여러 출력 발화를 갖는 입력 발화에 대하여, 각 하나씩의 출력 발화를 갖도록 분리하였다. 그리고 이렇게 생성된 총 112만 발화쌍을 80%의 학습 코퍼스와 20%의 평가 코퍼스로 나누었다.

평가는 번역 등의 연구에서 주로 사용되는 BLEU 스코어[8]를 사용하였다. 정답으로 주어진 출력 발화들을 번역에서의 정답과 같다고 보고, 모델의 결과를 평가하는 방식이다.

실험은 가장 적합한 파라미터를 알아내기 위하여 learning rate 등을 수정하며 진행하였고, 그 결과를 4. 2. 실험 결과 및 분석에 기술한다.

4. 2. 실험 결과 및 분석

모든 실험에서 hidden unit는 512, embedding size는 256을 사용하였고, layer는 2로 설정, dropout과 attention을 적용하였다. optimizer로 adam을 사용하였다.

첫 번째 실험은 LSTM으로 진행하였다. learning rate는 0.00001로 설정하였고, 학습 발화 89만 9천여 발화로 학습, 22만여 발화로 평가를 수행하였다.

두 번째 실험은 sequence-to-sequence의 Recurrent Node를 LSTM 대신 위에서 설명한 MTRNN의 변형으로 설정한 실험이다. learning rate는 0.0001로 설정하였고, τ는 첫 번째 레이어에 2, 두 번째 레이어에 3을 설정하였다. 5만 개의 발화로 학습, 5만 개의 발화로 평가를 수행하였다. 첫 번째 실험과 학습 및 평가 발화 수에 차이가 있다.

세 번째 실험은 두 번째 실험에서 τ를 3, 4로 수정한 실험이다. 두 번째 실험과 τ 설정에만 차이가 있고 다른 설정은 같다.

[표 2] 대화 모델의 실험별 성능

실험	cell type	BLEU1	BLEU2	BLEU3	BLEU4
1	LSTM	0.452	0.278	0.212	0.189
2	MTRNN varient	0.464	0.328	0.248	0.205
3	MTRNN varient	0.479	0.341	0.263	0.220

우리의 설정에서 하나의 입력 발화가 여러 출력 발화에 매치되는 경우가 있다. 우리는 이에 대한 분석을 수행하고자, 평가 코퍼스에서 몇 개의 샘플을 추출해 [표 6]에 정리하였다. 실제 학습은 형태소 품사가 부착된 형태소 단위이지만, 편의를 위해 원문을 복원하여 기술하였다.

[표 3] 대화 모델의 입력 발화에 대한 출력 샘플

입력발화	그래 미안한걸
정답 출력발화 (총 40발화)	내가 더 미안해
	미안하다면 다야? 마음 상했다고~
	나도 미안해~
모델 출력 발화	나도 미안하지~
입력발화	너 사귀는 사람 있어?
정답 출력 발화 (총 35발화)	난 화려한 싱글이야!!
	당연히 있지~
	놀리는 것이오?
모델 출력 발화	난 화려한 싱글이야?
입력발화	철수야 식구중에 언니 있니?
정답 출력 발화 (총 28발화)	없어... 하지만 누나 한 명 있었으면 좋겠다...
	누나 없당... 쏠로지~
	아니 없어. 혼자야
모델 출력 발화	당연한 A형이야?

위 두 예제는 적절히 잘 모델링 된 것으로 볼 수 있다. 첫 번째 예제 '그래 미안한걸'은 학습 코퍼스에서 유사한 발화 '진짜 미안했어', '그래 미안해요' 등이 학습되어서 효과를 발휘한 것으로 판단된다. 두 번째 예제는 더욱 확실하게 중첩되는 학습 데이터가 있었다. '철수야 너 사귀는 사람 있어?', '철수야 사귀는 사람 있어?' 등이다.

마지막 예제는 모델이 적절한 답변을 내어주지 못한 경우이다. 혈액형을 묻는 예제 발화쌍으로 '철수야 혈액형 뭐야?' / '자상한 A형이야~'가 있었다. 그리고 언니가 있는지 묻는 발화에 대해서는 '당연한'으로 시작하는 발화가 없었다. 우리의 실험에서 '당연/XR'과 '자상/XR'의 임베딩이 유사하고, 이 중에서 잘못 선택한 것이 뒤까지 영향을 미쳐 이러한 결과가 발생한 것이다.

5. 결론

높은 성능과 응용력을 갖춘 대화 모델링 기법을 목표로 많은 연구가 진행되고 있다. 우리는 sequence-to-sequence 구조에 Recurrent unit으로 LSTM과 MTRNN을 비교실험하고, 그 결과를 기술하였다. 우리의 실험에서는 학습 발화와 평가 발화에 수의 차이가 있지만 MTRNN이 LSTM에 비해 높은 성능을 보였다.

대화를 모델링하기 위한 연구가 계속 진행되고 있다. 우리의 이번 연구는 대화 모델링을 end-to-end로 수행할 수 있음을 보였다. 학습 후 측정한 성능은 BLEU4 0.161 ~ 0.220의 성능을 보였다. Attention 메커니즘과 MTRNN을 적용하였음에도 낮은 성능을 보여, 대화 모델링을 end-to-end로 수행하기 위해서는 심도 깊은 연구가 진행되어야 할 것으로 보인다.

사 사

이 논문은 2017년도 정부(교육부)의 재원으로 한국연구재단의 지원을 받아 수행된 기초연구사업임 (No. 2017R1D1A1B03033534).

참 고 문 헌

[1] 정의석, 박전규, seq2seq 주의집중 모델을 이용한 형태소 분석 및 품사 태깅, 한글 및 한국어 정보처리 학술대회 논문집, 2016.

[2] 이건일, 이의현, 이종혁, Sequence-to-sequence 기반 한국어 형태소 분석 및 품사 태깅, 정보과학회논문지, 44권 1호, 57-62, 2017.

[3] 박건우, 이현구, 김학수, Sequence-to-Sequence 기반 다중 발화 후보를 이용한 형태소 분석기, 한국컴퓨터종합학술대회 논문집, 648-650, 2017.

[4] 황현선, 이창기, Sequence-to-sequence 모델을 이용한 한국어 구구조 구문 분석, 한글 및 한국어 정보처리 학술대회 논문집, 2016.

[5] 이호경, 배경만, 고영중, 격틀과 워드 임베딩을 활용한 유사도 기반 대화 모델링, 한글 및 한국어 정보처리 학술대회 논문집, 2016.

[6] Sepp Hochreiter, Jurgen Schmidhuber, LONG SHORT-TERM MEMORY, Neural Computation, 9(8), 1735-1780, 1997.

[7] Yuichi Yamashita, Jun Tani, Emergence of Functional Hierarchy in a Multiple Timescale Neural Network Model: A Humanoid Robot Experiment, PLoS Computational Biology, 2007.

[8] Kishore Papineni et al, BLEU: a Method for Automatic Evaluation of Machine Translation, Association for Computational Linguistics, 311-318, 2002.

한국어 튜터링 챗봇을 위한 말뭉치 구축

김한샘○, 최경호, 한지윤, 정해영, 곽용진
연세대학교, ㈜이르테크, 연세대학교, ㈜이르테크, ㈜이르테크
khss@yonsei.ac.kr, khchoi@iirtech.co.kr, hanjiyoon01@gmail.com, hyjung@iirtech.co.kr, silhuett@iirtech.co.kr

Building a Corpus for Korean Tutoring Chatbot

Hansaem Kim○, Kyung-Ho Choi, Ji-Yoon Han, Hae-Young Jung, Yong-Jin Kwak
Yonsei University, IIR Tech Inc., Yonsei University, IIR Tech Inc., IIR Tech Inc.

요 약

교수-학습 발화는 발화 턴 간에 규칙화된 인과관계가 강하고 자연 발화에서의 출현율이 낮다. 일반적으로 어휘부, 표현 제시부, 대화부로 구성되며 커리큘럼과 화제에 따라 구축된 언어자원이 필요하다. 기존의 말뭉치는 이러한 교수-학습 발화의 특징을 반영하지 않았기 때문에 한국어 교육용 튜터링 챗봇을 개발하는 데에 활용도가 떨어진다. 이에 따라 이 논문에서는 자연스러운 언어 사용 수집, 도구 기반의 수집, 주제별 수집 및 분류, 점진적 구축 절차의 원칙에 따라 교수-학습의 실제 상황을 반영하는 준구어 말뭉치를 구축한다. 교실에서 발생하는 언어학습 상황을 시나리오로 구성하여 대화 흐름을 제어하고 채팅용 메신저와 유사한 형태의 도구를 통해 말뭉치를 구축한다. 이 연구는 한국어 튜터링 챗봇을 개발하기 위해 말뭉치 구축용 챗봇과 한국어 학습자, 한국어 교수자가 시나리오를 기반으로 발화문을 생성한 준구어 말뭉치를 최초로 구축한다는 데에 의의가 있다.

주제어: 한국어교육(Korean Language Education), 튜터링(tutoring), 챗봇(chatbot), 말뭉치(corpus)

1. 서론

이 연구의 목적은 한국어 튜터링 챗봇 개발에 필요한 말뭉치의 구축이다. '한국어 튜터링 챗봇'은 물리적 거리나 비용의 한계를 극복하고 챗봇과의 대화를 통해서 자연스럽게 한국어 구사 능력과 언어 지식을 습득할 수 있도록 하는 대화 시스템을 의미한다. 이러한 시스템을 개발하기 위해서는 한국어 교수-학습이라는 특수한 영역에 최적화된 말뭉치가 필요하고 이를 구축할 방법론과 도구에 대해 논의해야 한다. 자연언어 기반 대화형 인터페이스는 한국어를 모국어로 하는 화자의 일반 대화 자료를 주로 사용해 왔다. 그러나 자연스러운 대화 전개를 위해서는 해당 도메인과 채팅 환경 등 실제 시스템이 작동하는 환경에서 생성된 언어자원이 필요하다. 이 연구에서 구축하여 활용하고자 하는 말뭉치는 실제 한국어 교수-학습 상황을 반영하기 때문에 챗봇이 튜터링 프로세스를 진행하는 데에 직접적인 도움을 줄 수 있다.

1960년대에 세계 최초로 Brown 말뭉치가 구축된 이후 지금까지 수많은 다양한 말뭉치가 구축되어 왔는데, 이들은 자연언어처리 분야에서의 활용이라는 관점에서 크게 세 가지 유형으로 나눌 수 있다. 첫 번째는 Brown 말뭉치로부터 영국 국가 말뭉치(BNC: British National Corpus)로 이어진 전통적인 말뭉치로 인간의 언어를 있는 그대로의 모습으로 관찰하고자 하는 데 그 특징이 있다. 전통적인 말뭉치는 형태소 분석, 구문 분석, 의미 분석과 같은 기초적인 언어 처리에 주로 사용된다. 두 번째로 게임, 영화, 소설과 같이 내러티브가 있는 콘텐츠 제작, 텍스트의 요약 및 생성 등에 사용되는 말뭉치가 있다. 이들 말뭉치는 이야기 모티브와 전개 정보, 주제에 대한 정보를 중점적으로 담고 있어서 컴퓨터가 화제의 전개와 커뮤니케이션, 창의성 등을 모방하는 데 사용된다. 마지막으로 말뭉치에 포함된 광범위한 정보를 제한하여 목표 시스템 또는 서비스 환경에서의 언어 사용을 충실하게 담아내는 말뭉치들이다. Standford의 QnA를 위한 SQUAD나 대화형 에이전트 개발에 자주 사용되는 Ubuntu 채팅 말뭉치, 트위터나 페이스북 데이터가 대표적이다. 이들 말뭉치는 목표 시스템이나 서비스의 환경 하에서 생성된 인간의 언어 사용을 담고 있어서 보편적인 언어 사용 정보로 인한 기계학습 효과의 발산을 막는다. 이러한 말뭉치들은 구축 단계부터 특정 시스템이 결합됨으로써 전산적 처리가 용이하고 일관성이 높아 챗봇, 자동 QA, 감성 분석 등과 같은 목표 지향적 시스템 및 서비스 개발에 효과적이다.

이 연구에서 구축하는 말뭉치는 마지막 유형에 해당하는 말뭉치로 한국어 튜터링용 챗봇이라는 구체적인 시스템의 개발을 위해 기계에 학습시킬 데이터로서, 말뭉치 구축용 챗봇과 한국어 학습자, 한국어 교수자 등의 화자가 시나리오를 기반으로 발화문을 생성하여 구축하는 최초의 말뭉치이다.

2. 관련 연구

Facebook 등에서는 다양한 챗봇용 언어자원을 수집하기 위한 플랫폼을 개발하여 공개하고 있다[1]. 챗봇을 이용한 언어자원 수집은 챗봇과 인간 사용자간의 대화뿐만 아니라, 챗봇과 챗봇, 챗봇들과 인간이 복합된 환경 등 다양한 상황, 테마, 발화 주체를 구성하여 진행되고 있다. 챗봇을 위한 트레이닝 데이터를 확보하기 위해 실제 대화를 전사하여 구축한 스크립트를 가공하거나, 특정 서비스를 이용하기 위한 대화를 전사하여 텍스트화하는 방식, 웹에서 수집한 대화를 가공하는 방식으로 데

이터를 구축한다[10]. 교육, 학습 분야에서도 컴퓨터를 활용한 튜터링 기술을 주목하면서 CALL이나 튜터링용 챗봇 개발에 대한 연구가 수행되어 왔다[2]. 언어 튜터링 챗봇이라는 언어 교육용 시스템 개발을 위해 특화된 말뭉치를 따로 구축한 경우는 흔하지 않다. 교육용 대화 시스템의 논리 전개와 추론을 위해 오류를 포함하는 영어 학습자 말뭉치를 통사의미적으로 분석하여 활용한 사례가 있다[11]. 국내에서도 학습자의 중간 언어 사용 양상을 구축한 한국어 학습자 말뭉치가 국가 차원과 대학 연구소 차원에서 구축된 바 있다[12], [13]. 그러나 발화자와 주제가 통제되지 않은 학습자 말뭉치는 바로 언어 처리에 활용하기 힘들고 특히 한국어 학습자 말뭉치는 문어 중심으로 구축되어 있어 대화를 전제로 하는 챗봇 시스템에는 활용도가 떨어진다. 학습자 구어 말뭉치를 대신해 활용할 만한 말뭉치로 한국어 교재 말뭉치가 있다. 연세대 언어정보연구원에서 구축하여 서비스하고 있는 한국어 교재 말뭉치는 교육 항목 전달을 위해 인위적으로 만든 정제된 대화로 구성되어 있어 오류를 포함하는 학습자 말뭉치보다 활용도가 높다[4]. 다만 발화의 길이와 수준이 학습자 등급에 따라 통제되어 있어, 대화 참여자의 발화가 담당한 기능을 화제 도입, 화제 전개, 상대방 발화 내용 확인, 발화 보충, 발화 수정, 대답 발화, 대화 지속 반응, 발화 지연, 의식적 표현 등으로 분류할 때[14], 기본적인 기능을 수행하는 발화에 치우쳐 있어 자연스러운 의사소통을 산출하기 위한 자료로는 부족하다. 이 연구와 같이 교육용 챗봇 시스템에 활용하기 위해 실제 교수-학습 현장에서의 교수자와 학습자의 상호 작용을 반영한 말뭉치를 구축하기 위한 논의는 진행된 바가 없다.

3. 교수-학습 말뭉치 구축 방법론

3.1. 교수-학습 말뭉치의 필요성

한국어 교재 말뭉치 등에서 볼 수 있는 언어 학습의 대화 상황 예시는 주로 다음과 같이 이루어져 있다.

정우: 투이 씨, 로라 씨가 이번에 승진을 했어요.
이야기 들었어요? 정말 잘 됐어요.
투이: 어제 로라씨가 저에게도 전화했어요. 정말 잘됐어요.
정우: 그래서 내일 저녁에 친구들이 모여서 로라 씨 승진을 축하해 주려고요.
투이: 좋아요. 갈게요. 저도 뭘 좀 도와드릴게요.

예시1 세종 한국어3(p36.대화2) 국립국어원

로라: 저건 색깔이 너무 화려할까요?
직원: 어머니께 드릴 선물이라면 이 색깔이 더 좋을 것 같습니다.
로라: 음, 그럼 이걸로 포장해 주세요. 그런데 혹시 어머니가 보시고 색깔을 마음에 들어하지 않으시면 교환할 수 있어요?
직원: 네, 일주일 이내에 오시면 교환이 가능합니다. 교환하러 오실 때는 반드시 영수증을 가지고 오셔야 합니다.

예시2 세종 한국어5(p70.대화2) 국립국어원

위의 예는 실제 대화를 전사한 구어 말뭉치가 아니라 대화를 연습하기 위한 스크립트로서 준구어 말뭉치의 일종이다. 이러한 대화는 실제 발화 상황을 그대로 반영할 수 없기 때문에 챗봇 시스템에 활용하는 데에 있어 여러 가지 한계점을 가지고 있다. 특정 상황에 대한 암묵적 가정 하에 대화를 진행하므로 도입부, 마무리 없이 배워야하는 교육 항목에 해당하는 4~5 턴의 짧은 대화만 주어져 있다. 대화를 시작하거나 마무리할 때에 필요한 의식적 발화가 생략되어 있다는 것이다. 교육 항목에 해당하는 어휘와 표현을 포함시키기 위해 구성한 대화이므로 교실수업에서 교사-학생의 대면을 전제로 함에도 불구하고 학습자의 감성 화행이 간과되어 관계 중심적 학습 보다는 과제 중심적 학습 지문의 흐름이 연출되어 있다[9].

이러한 한계를 극복하기 위하여 챗봇 등의 컴퓨터 대화 기반 튜터링에 필요한 교수-학습 말뭉치를 구축할 때에 도입부터 마무리까지 1:1 '챗봇:학습자' 대화 형태로 구성되어야 하며, 어휘나 표현은 학습자의 수준과 상황에 따라 달라지므로, 교과서상 제시된 맞춰진 본문과는 달리 예측 어려운 학습자 발화에 대한 대응까지 고려하는 어려움이 있다. 따라서 대화쌍도 교사인 챗봇 발화를 포함시켜야 하며, 최소 7~8 턴 이상으로 구성해야 한다. 기존의 말뭉치를 활용하기 힘들고 챗봇을 위한 말뭉치를 따로 구축해야 할 필요성이 여기에 있다.[5][6]

교수-학습 말뭉치는 지식의 전달과 숙련을 위한 구조화된 흐름 속에서 제한된 언어 사용이 이루어진다. 세종 말뭉치나 BNC와 같은 전통적 말뭉치에도 이러한 교수-학습 상황의 언어사용 자료가 포함되어 있으나, 일정한 교육적 목표와 일관된 다양한 학습 주제를 획득하기에는 어려움이 있다. 교수-학습 발화, 특히 언어학습에 대한 교수-학습 발화에는 다음과 같은 특징이 있다.

1) 발화 턴 간에 규칙화된 인과관계가 강함
2) 자연 발화에서의 출현율이 낮음
3) 일반적으로 어휘부, 표현 제시부, 대화부로 구성
4) 커리큘럼과 화제에 따른 언어자원 구성 필요

교사-학생간의 발화는 교사의 질문이나 학습 지시에 대한 학생의 반응, 질문과 수행으로 이루어지는 경우가 많다. 이러한 특성은 보편적인 언어현실을 포착하기 위한 전통적 말뭉치에서는 잘 드러나지 않는다. 대부분의 발화에도 발화 턴간에 유의미한 인과관계가 발생하나, 교수-학습과 같은 특수한 환경에서는 교사의 질문, 지시가 발화의 흐름을 주도한다.

또한 교수-학습 발화는 교육하고자 하는 내용(어휘, 표현, 문법)과 이를 이해시키기 위한 설명적 재료(화제, 문화, 경험)가 일련의 연관성을 지니고 구조화되어 나타난다. 예를 들어 [날씨]를 [묻는] [표현]을 가르치기 위해 [최근 장마철 날씨]를 주제로 한 대화 상황을 조성하여 교육을 진행한다. 그러므로, 교수-학습 발화 말뭉치는 이러한 교육적 구성과 화제에 따라 언어자원을 수집하여 분류해야 한다.

3.2. 교수-학습 말뭉치 구축의 원칙

이 연구에서는 한국어 학습을 위한 말뭉치를 구축함에 있어서 아래 4가지의 원칙을 견지한다.

1) 자연스러운 언어 사용 수집
한국어 교재에 등장하는 대화문과 같이 인위적으로 교사-학습자 발화를 작성하지 않는다. 기능을 중심으로 분류할 때 발화의 규형이 골고루 적재적소에 배치될 수 있도록 실제 학습자의 발화문과 교수자의 발화를 통한 교육적 접근을 수집한다.

2) 도구 기반의 수집
교사와 학습자의 대화가 일정한 교육 흐름에 따라 유도록 되도록 대화형 인터페이스를 제공하고, 특히 교사의 발화는 교육적 흐름에 대한 발화 의도가 자연스럽게 주석될 수 있도록 교육 흐름의 정보를 제공한다. 교수-학습 말뭉치 구축용 도구는 학습자의 반응과 교육 흐름에 따른 다양한 교수 발화를 수집하는 교수 발화 수집 모드, 학습자의 다양한 반응을 수집하는 학습자 발화 수집 모드, 수집된 발화 또는 챗봇간 발화의 문제점을 교정하는 관찰 모드가 제공된다. 또한 수집된 말뭉치로부터 발화 턴간 상관 관계 분석이 용이하도록, 형태, 구문, 발화 의도를 주석할 수 있는 자동/반자동 주석 도구를 제공한다.

3) 주제별 수집 및 분류
앞서 언급 바와 같이, 동일한 교수 흐름이라도 학습자의 흥미, 수준, 반복 정도에 따라 서로 다른 주제와 교육 내용이 필요하다. 예를 들어 '시제'에 대한 문법적 교육을 위해 사용될 수 있는 주제는 '날씨'일 수도 있고, '여행' 계획에 대한 주제일 수도 있다. 또한 각각의 주제와 상황으로부터 교육 주제로 이끌어 가는 방법 또한 다양하게 나타난다. 그러므로 교사-학습자의 발화쌍은 주제별로 분류될 것을 고려해야 한다.

4) 점진적 구축 절차

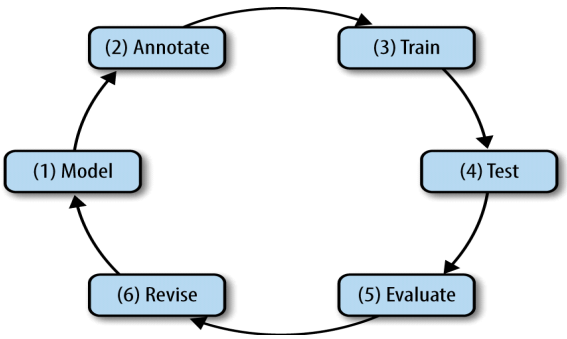


그림 1 MATTER Cycle(Pustejovsky & Stubbs 2012)

말뭉치의 구축이 점진적으로 수행되어야 함은 오래전부터 강조되어 왔다[7],[8]. 특히, 보편적인 언어현실을 반영하기 위해 자연스러움이 강조되는 전통적 말뭉치와 달리 기계학습을 전제로 하는 말뭉치는 말뭉치가 기계학습의 대상인 동시에 결과물이므로 구축 목적의 지향성과 부합하는지를 지속적으로 확인하는 과정이 필요하다. 그림 1은 Pustejovsky & Stubbs이 기계학습을 위한 말뭉치와 주석 개발을 위해 제시한 MATTER 프로세스이다. 지식 모델과 지침을 기반으로 구축한 말뭉치를 기계학습을 통한 훈련, 테스트, 평가를 거쳐 지식 모델과 지침을 개선한 뒤 데이터를 확대하는 방식으로 언어자원의 양적 희소성과 소규모 데이터를 학습 모델의 수렴에 용이하게 하는 장점이 있다. 대규모 자료 수집이 쉽지 않고 유사한 기존 말뭉치가 존재하지 않는 교수-학습 말뭉치의 구축에 적합하다.

3.3. 교수-학습 말뭉치의 설계

한국어 교육용 챗봇 개발에 필요한 준구어 말뭉치 구축은 우선 교수-학습 상황을 배경으로 한 예상 대화 시나리오를 기본으로 한다. 이 대화 시나리오는 교수해야 하는 어휘와 표현, 주제 등을 고려하여 설계된다.

대화는 교사의 발화와, 이에 대한 학습자의 예상 발화, 그에 따른 교사의 피드백이 하나의 단위로 구성되며, 이 단위들이 복수 개로 모여 일정한 흐름을 지닌 것이 대화 시나리오가 된다. 한국어 교육용 챗봇에 사용될 대화 말뭉치 구축 도구는 실제 교사와 학습자에게 각각의 발화를 수집하기 위해 이러한 학습 시나리오에 따른 흐름을 교사와 학습자에 제공하도록 설계되었다.

교사와 학습자에게 기대하는 역할이 다르기 때문에 교사와 학습자는 서로 다른 환경에서 구축에 참여하게 된다. 교사는 발화의 적절성을 판단하면서 동시에 발화를 생성한다. 발화를 생성하는 동시에 제공받은 대화 단위의 구성과 흐름이 적절한 지 판단하는 것이다. 이러한 과정을 통해 실제 교수-학습 상황과 유사한 발화를 수집하여, 실제 챗봇이 학습자에게 다양한 발화를 제공할 수 있게 된다. 또한 시나리오를 벗어나는 학습자의 발화에 대해 유연한 대응이 가능해진다. 학습자의 경우, 제공된 발화에 대한 응답을 작성한다. 이 때 수집한 학습자의 발화는 실제 대화형 인터페이스에 입력될 학습자의 발화 예측을 돕는다. 말뭉치 안에 다양한 유형의 학습자 발화가 포함될수록 대화형 인터페이스의 품질이 높아진다.

교사와 학습자의 상호작용이 반영된 말뭉치를 구축하기 위한 세부 계획은 다음과 같다. 한국어 급수 3-4급에 해당하는 중급 실력의 학습자 20명과, 경력 5년 이상의 한국어 교사 5명을 실험대상으로 한다. 총 구축 단계는 6차례로 10개 주제에 대한 서로 다른 시나리오에 대한 학습자와 교사의 발화를 수집한다. 1차와 2차는 본 수집에 앞선 파일럿 작업으로 우선 1개 주제에 대한 데이터를 수집한다. 1차에서는 예상된 시나리오에 대한 학습자의 발화를 수집한다. 2차 구축 단계에서는 1차에서 수집된 학습자의 발화를 교사에게 제공하여 기존 시나리오의 수정 발화와 학습자 발화에 대한 응답 발화를 수집한다. 1차와 2차의 발화를 토대로 보완점을 개선한 뒤, 주제를 확장하여 학습자와 교사의 발화를 교대로 수집하는 방식으로 3-6차 수집을 완료한다. 예상 수집 발화량은 10만

어절로, 각 7-8 단위로 구성된 10개 주제 시나리오에 대하여 각각 100개의 변이형, 총 1000개 내외의 변주된 시나리오를 수집하게 된다.

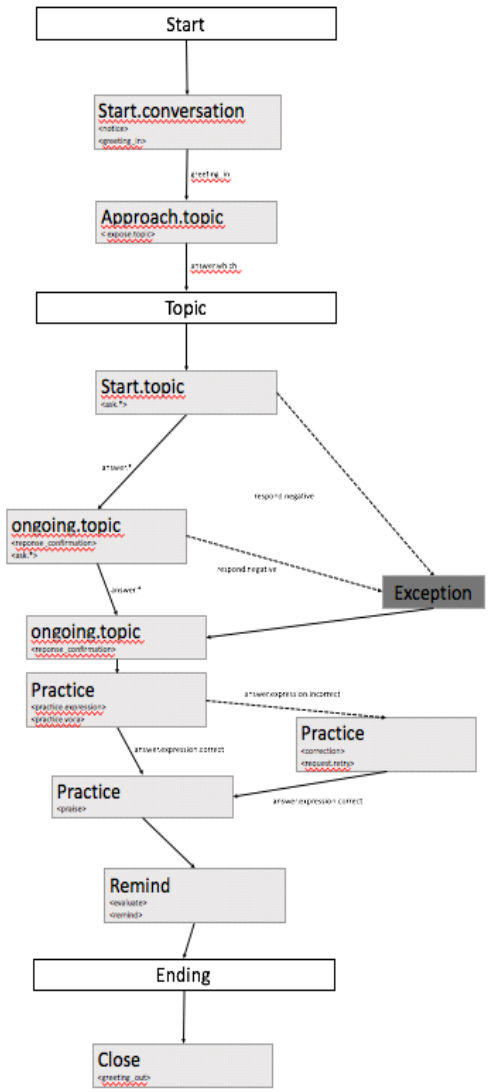


그림 2 교수-학습 말뭉치 구축도구의 교수 시나리오

4. 교수-학습 말뭉치 구축 도구

교수-학습 말뭉치는 교사-학습자간의 발화를 수집하기 위해 기본적으로 채팅용 메신저와 유사한 형태를 갖는다. 또한 언어학습 상황을 충실히 재현하도록 교실에서 발생하는 언어학습 상황을 시나리오로 구성하여 대화 흐름을 제어한다.

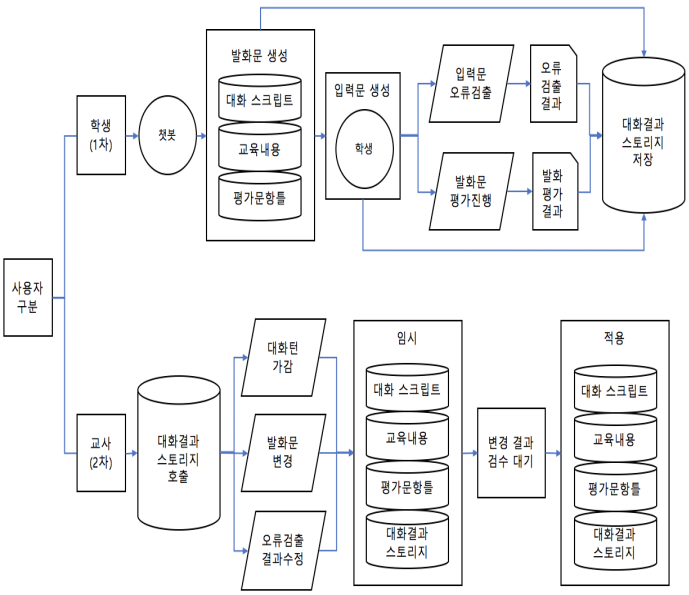


그림 3 말뭉치 구축 도구 시스템 구성도

학습자 대화 말뭉치 구축 프로세스는 크게 두 단계로 구분된다. 1차 작업의 목표는 첫째로 챗봇 발화에 대한 학습자의 응답문 데이터를 획득하기. 둘째로 챗봇 발화의 적절성에 대한 학습자의 피드백 정보를 얻기이다. 따라서 1단계에서는 기 제작된 시나리오에 의거하여 챗봇과 학습자가 대화를 진행하면서 학습자가 입력한 문장을 기록하고, 학습자가 챗봇의 발화문에 대해 추가 정보를 요청할 수 있게 하여 최종적으로 획득한 조작기록을 분석해 챗봇 발화문의 적절성에 대해 간접적으로 평가할 수 있도록 한다. 동시에 시스템에 내장된 오류검출 기능을 통해 오류라고 인식된 부분들도 함께 대화 기록 파일에 저장한다. 2차로 학습자와 챗봇의 대화를 통해 생성된 대화 기록을 기반으로 한국어 교사가 챗봇 시나리오의 적절성 및 시스템의 오류검출 결과의 타당성에 대해 검수하고 수정하는 작업을 진행한다. 검수작업 결과는 개인 작업자 별로 보관되었다가 시스템 적용 검토과정을 거쳐 최종적으로 챗봇 시스템에 적용하게 된다.

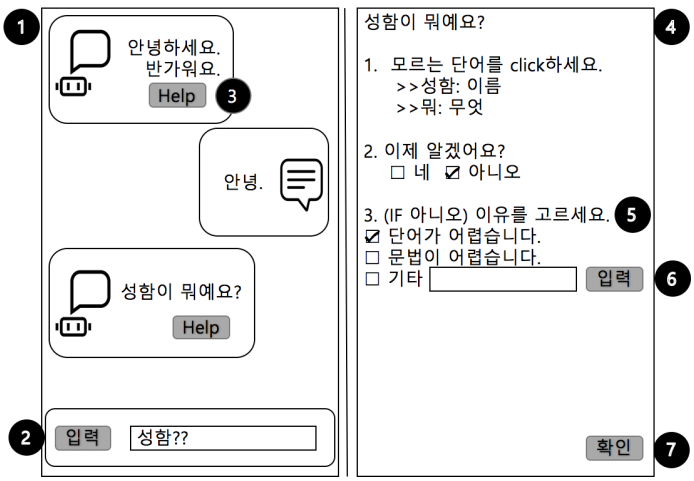


그림 4 학습자 사용화면

1차 구축작업에서 사용되는 학습자 사용화면을 살펴보면, ①영역에서 실제 챗봇과 학습자의 사이에 진행되는 대화를 볼 수 있다. 학습자는 챗봇의 발화문을 이해한 뒤 ②영역에 문장을 입력하는 방식으로 응답할 수 있다. 만약 챗봇의 발화를 이해하지 못하여 도움이 필요한 경우 ③Help 버튼을 클릭하여 ④영역에 나타난 발화문 정보요청 페이지를 통해 부족한 정보를 획득하는 작업을 진행한다. ⑤부분은 만약 학습자가 시스템에서 제공하는 정보를 획득하고도 충분하지 못하다고 생각하는 경우 발화문 적절성을 저해하는 요소가 무엇인지 파악할 수 있도록 추가된 문항이며 선택사항이 없을 경우에는 ⑥을 통해 서술형으로 응답이 가능하다. 사용자가 ⑦확인 버튼을 클릭하면 다시 ②의 입력창이 활성화 되어 대화를 진행 할 수 있다.

2차 구축작업에서 사용되는 교사 사용화면을 살펴보면 오류 검출 결과와 시나리오의 적절성을 검수하는 기능이 함께 제공된다. 다만 그림 4와 그림 5는 기능 설명을 위해 별도로 분리하였으며 실제로는 작업화면 좌측(그림 4, 5의 ①번 영역)의 대화 기록창에 제공된 기능 버튼을 클릭하는 것에 따라 우측(그림 4의 ③번, 그림 5의 ⑥번 영역)에 Toggle형태로 출력되는 작업화면이 달라지게 된다.

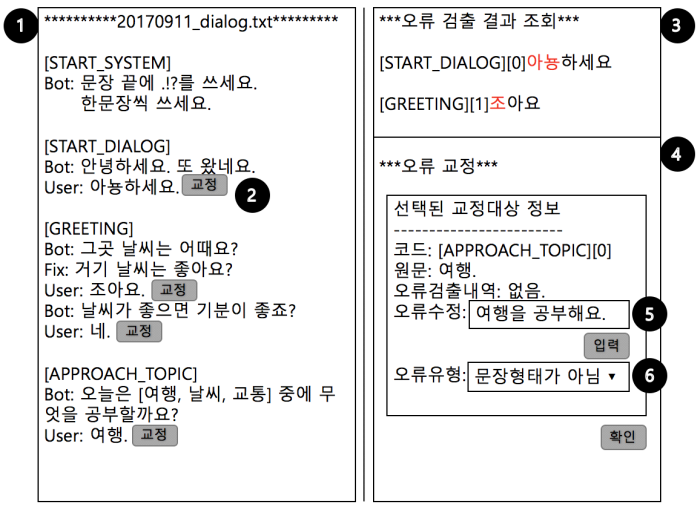


그림 5 교사 사용화면 - 오류 검출 결과 검수

그림 5는 시스템에서 자동으로 검출한 학습자 입력문의 오류내역을 한국어 교사가 검수할 수 있도록 특정 대화기록에 대한 오류검출내역을 ③번 영역에 출력해주고 교사는 검출내역에서 누락되거나 부적절한 교정을 발견하였을 경우에 ②번의 교정버튼을 클릭하여 나타난 ④번 영역의 교정작업 창에서 검수작업을 진행하게 된다. ⑤번에 적절하다고 판단되는 오류교정 내용을 서술형으로 입력하고 ⑥에서 오류의 유형을 선택한 뒤 ④번 영역 하단의 확인버튼을 클릭하여 작업결과를 저장한다.

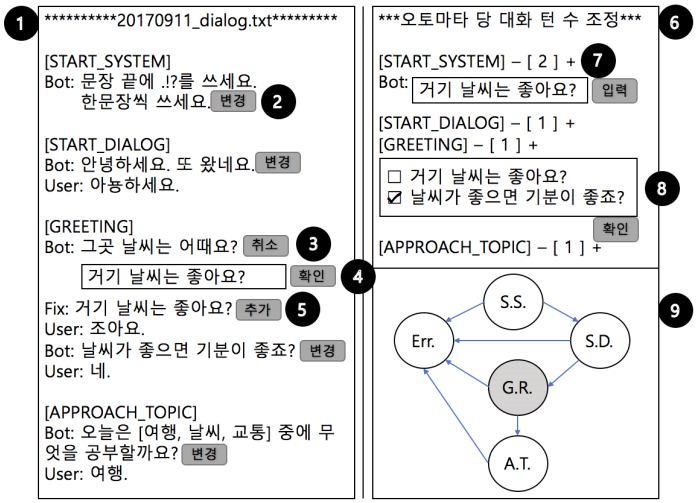


그림 6 교사 사용화면 - 시나리오 적절성 검수

그림 6은 챗봇의 발화와 학습자의 응답 기록을 한국어 교사가 살펴보고 기록된 학습자의 응답 양상과 그림 3에서 획득한 학습자의 챗봇 발화에 대한 Feedback 정보를 토대로 챗봇의 발화문을 수정하는 검수기능이 제공되는 화면이다. 교사는 챗봇 발화가 부적절하다고 판단되면 ②번의 변경버튼을 클릭한다. ④번 영역에 수정할 문정을 입력하고 확인버튼을 눌러 저장하면 ⑤번에 [Fix]라는 표지가 붙은 수정문장이 신규로 추가되어 교사가 수정한 문장을 시각적으로 확인이 가능하다. 또한 문장 끝의 추가버튼을 클릭하면 해당 단계에서 적절하다고 생각되는 수정문장을 추가로 입력이 가능하기 때문에 각 대화 시나리오 단계의 특정 대화 턴에 복수의 수정문을 입력 가능하다. ⑥번 영역에서는 각 대화 시나리오 단계의 대화 턴이 부족하거나 많다고 생각되는 경우 (+)(-)버튼을 클릭하여 턴 수를 조정할 수 있다. 대화 턴을 추가하는 경우에는 ⑦번 영역처럼 챗봇의 발화문을 교사가 직접 입력하여 추가가 가능하고, 대화 턴을 삭제하는 경우에는 ⑧번 영역처럼 기 제작된 발화문이 제공되어 그 중 한 가지를 선택해 제거가 가능하다. ⑨번은 현재 작업중인 대화 시나리오 단계를 시각적으로 강조하여 교사의 검수작업에 도움을 주고자 하였다.

5. 결론 및 향후 과제

이 연구는 한국어 튜터링 챗봇을 개발하기 위해 말뭉치 구축용 챗봇과 한국어 학습자, 한국어 교수자의 세 주체가 대화 흐름에 통제를 가한 시나리오를 기반으로 발화문을 생성한 준구어 말뭉치를 구축하는 것을 목적으로 하였다. 학습자의 감성 화행을 간과한 과제 중심적 대화문으로 구성된 짧은 대화 자료의 한계를 극복하기 위해 학습 시나리오에 따른 흐름을 교사와 학습자에 제공하도록 설계된 말뭉치 구축용 챗봇과 학습자의 대화, 이에 대한 교수자의 검증 및 응답 발화 생성 등의 단계를 거쳐 1000개 내외의 변주된 시나리오를 10만 어절 내외로 구축한다.

구축된 한국어 교수-학습 말뭉치는 교육용 목적에 최적화된 준구어 말뭉치로서 튜터링 챗봇 개발에 직접적으로 활용할 수 있는 데이터이다. 자연 발화의 출현율이 높아 말뭉치 전체가 학습 데이터로 사용될 수 있으며, 학습자의 경험을 토대로 한 다양한 발화를 포함하므로 챗봇 인터페이스의 품질을 향상시킬 수 있다. 구체적으로 제안한 말뭉치 수집 방법과 구축 도구는 챗봇용 학습 콘텐츠를 개발하고 챗봇의 대응 알고리즘을 구성하는 데에 기여할 것으로 기대한다. 말뭉치 수집이 진행되면서 데이터가 구축 목표에 부합한지 확인하는 절차를 통해 점진적으로 말뭉치의 완성도가 높아지는 과정에 대한 논의와 실제 챗봇 시스템에 활용한 결과에 대한 분석이 향후 과제이다.

감사의 글

이 논문은 2017년도 정부(미래창조과학부)의 재원으로 정보통신기술진흥센터의지원을 받아 수행된 연구임 (No.2017-01217,말하기/쓰기 평가와 챗봇을 이용한 1:1 언어학습 튜터링 기술 개발)

참고문헌

[1] Alexander H. Miller, Will Feng, Adam Fisch, Jiasen Lu, Dhruv Batra, Antoine Bordes, Devi Parikh, Jason Weston, ParlAI: A Dialog Research Software Platform, eprint arXiv:1705.06476, 2017.

[2] Jiyou Jia, “CSIEC: A computer assisted Englishlearning chatbot based on textual knowledge and reasoning”, Knowledge-BasedSystems 22(4), p.p 249-255, Elsevier B.V. 2009.

[3] Jiwei Li, Michel Galley, Chris Brockett, Georgios P. Spithourakis, Jianfeng Gao, Bill Dolan, "A Persona-Based Neural Conversation Model", Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, pp.994-1003, Berlin, Germany, 2016.

[4] 언어정보연구원, 한국어 교재 말뭉치 https://ilis.yonsei.ac.kr/corpus/koreantext3

[5] 박범준(2010), “콘텐츠 로봇의 감성적 반응을 위한 지능형 메신저 개발”, 한국콘텐츠학회논문지 ‘10Vol.No.9.pp.13~14

[6] 조윤주 외(2009), 모바일 환경에서의 대화형 에이전트와 대화 내용에 관한 연구

[7] Biber, D. Finegan, E. “On the exploitation of computerized corpora in variation studies”, in Aijmer,K. & Altenberg, B. (ed.), 1991.

[8] James Pustejovsky, Amber Stubbs, “Natural Language Annotation for Machine Learning”, O’Reilly Media, 2012

[9] 박창균, “대화분석을 통한 말하기 교수-학습 방법 연구” 인천교대 교육대학원 초등국어 교육전공 석사학위논문,pp.23-25,1998

[10] Kadlec, R., Schmid, M., & Kleindienst, J. (2015). Im-proved deep learning baselines for ubuntu corpus dialogs. arXiv preprint arXiv:1510.03753

[11] Jia, J. (2009). CSIEC: A computer assisted English learning chatbot based on textual knowledge and reasoning. Knowledge-Based Systems 22(4), 249-255.

[12] 강현화, “학습자 말뭉치의 구축과 활용연구 ‘, 소통, 2017

[13] 서상규, 유현경, 남윤진, 한국어 학습자 말뭉치와 한국어 교육, 한국어 교육, 제13권 제1호, pp. 127-156, 2002

[14] 강현주, 말하기 능력 평가에서 대화 과제 도입의 필요성, 어문논집, 71, pp.353-376, 2014

텍스트 마이닝을 이용한 기사 내 부적합 문단 검출 시스템

김규완[0], 신현주, 김선진, 이현아
금오공과대학교, 컴퓨터소프트웨어공학과
rla9826@naver.com, dotcomehe@naver.com, junnis0123@naver.com, halee@kumoh.ac.kr

Detecting Improper Sentences in a News Article Using Text Mining

[Kyu-Wan Kim[O], Hyun-Ju Sin, Seon-Jin Kim. Hyun Ah Lee]
Dept. of Computer Software Engineering, Kumoh National Institute of Technology

요 약

SNS와 스마트기기의 발전으로 온라인을 통한 뉴스 배포가 용이해지면서 악의적으로 조작된 뉴스가 급속도로 생성되어 확산되고 있다. 뉴스 조작은 다양한 형태로 이루어지는데, 이 중에서 정상적인 기사 내에 광고나 낚시성 내용을 포함시켜 독자가 의도하지 않은 정보에 노출되게 하는 형태는 독자가 해당 내용을 진짜 뉴스로 받아들이기 쉽다. 본 논문에서는 뉴스 기사 내에 포함된 문단 중에서 부적합한 문단이 포함되었는지를 판정하기 위한 방법을 제안한다. 제안하는 방식에서는 자연어 처리에 유용한 Convolutional Neural Network(CNN)모델 중 Word2Vec과 tf-idf 알고리즘, 로지스틱 회귀를 함께 이용하여 뉴스 부적합 문단을 검출한다. 본 시스템에서는 로지스틱 회귀를 이용하여 문단의 카테고리를 분류하여 본문의 카테고리 분포도를 계산하고 Word2Vec을 이용하여 문단간의 유사도를 계산한 결과에 가중치를 부여하여 부적합 문단을 검출한다.

주제어: deep learning, text embedding, Word2Vec, Doc2Vec, logistic regression, 부적합 문단 검출

1. 서론

인터넷의 발전으로 인해 뉴스 콘텐츠는 TV보도와 종이 신문을 대신하여 인터넷 뉴스가 주류를 이루는 형태로 변화했다. 이로 인하여 뉴스 독자는 개인의 필요와 관심에 맞는 뉴스 기사를 선택하여 읽을 수 있게 되었다. 하지만 사람들이 흥미 위주로 뉴스 기사를 열람하는 성향이 강해지면서 이를 악용하여 가짜 뉴스를 배포하는 개인 및 기업이 등장했다. 특히 SNS와 개인 미디어를 통하여 확산되는 가짜 뉴스는 사람들을 속이기 위한 목적으로 만들어진 만큼 그 파급력이 크다[1]. 이러한 가짜 뉴스가 점차 늘어나면서 사람들의 혼란이 초래되고 매체에 대한 불신이 높아지고 있다. 하지만 수많은 매체가 쏟아내는 방대한 양의 뉴스 데이터를 사람의 손으로 분류해내는 데에는 비용과 시간의 한계가 있다. 때문에 가짜 뉴스를 자동으로 분류해내기 위한 알고리즘에 대한 연구는 필수 불가결이다. 하지만 가짜 뉴스가 무엇인지 명확하게 정의하기 어렵고, 사람조차 구분해내기 힘든 사실에 대한 진실과 거짓을 컴퓨터가 분류하는 것은 쉽지 않은 문제이다.

이러한 가짜 뉴스의 한 유형에는 화제성 높은 주제를 채택하고 본문 내부에 다른 분야의 문단을 섞어 넣거나 광고성 문구를 포함하는 것이다. 본 논문에서는 이러한 가짜 뉴스 유형을 본문에 해당되는 카테고리와 다른 카테고리의 문단이 포함된 뉴스로 보고 이러한 문단을 검출하기 위한 방법을 제안한다. 뉴스 문단의 카테고리의 자동 분류에서는 자연어 처리에 대해 이미 성능이 검증된 Word2Vec 방식을 적용하고, tf-idf 가중치 알고리즘에 로지스틱 회귀를 적용한 뉴스 부적합 문단 검출 방법을 제안한다.

2. 관련 연구

최근 들어 Convolutional Neural Network(CNN)가 자연어처리에 적용되기 시작하면서 놀라운 결과를 얻고 있다[2]. CNN모델 중 하나인 Word2Vec은 입력한 말뭉치의 문단에 있는 단어와 인접 단어의 관계를 이용하여 단어의 의미를 학습한다. Word2Vec의 학습 방법은 CBOW, Skip-gram의 두 종류가 있다. CBOW(Continuous Bag Of Words)방식은 주변 단어가 만드는 맥락을 이용해 타겟 단어를 예측하고, Skip-gram은 한 단어를 기준으로 주변에 올 수 있는 단어를 예측한다.

Word2Vec[3]의 학습 과정은 큰 틀에서 일반적인 인공신경망의 학습과 비슷하다. 한 단어에 이미 할당된 벡터, 즉 단어 임베딩(word embedding)이 있다고 가정하고 이 값을 이용해 주변 문맥을 얼마나 정확하게 예측하는

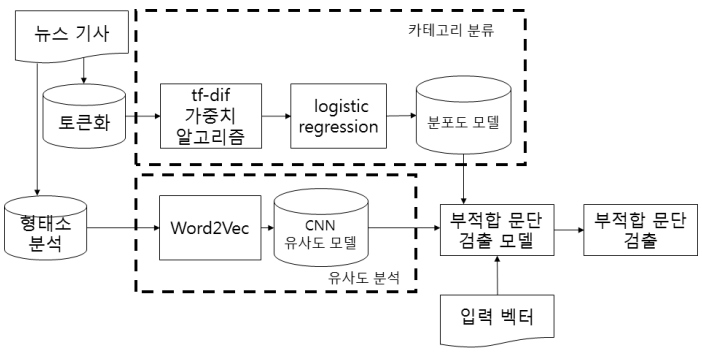
지 계산한다. 학습 과정에서 한 단어를 기준으로 단어 주변의 문맥을 참고하여 현재 임베딩 된 벡터가 얼마나 정확한지, 오차의 값은 어느 정도인지를 알아낸다. 만일 어떤 두 단어가 비슷한 문맥에서 꾸준하게 사용될 경우 두 단어는 비슷한 벡터 값은 갖는다. 학습이 잘 완료되었다면 고차원 공간에서 비슷한 단어가 근처에 위치하게 되는데 이 관계를 이용하면 유사성을 알아낼 수 있다.

Doc2Vec[4]은 Word2Vec을 문단, 단락 또는 전체 문서와 같이 더 큰 텍스트 블록에 대한 연속 표현을 학습하도록 수정한 모델이다. Doc2Vec은 단어와 레이블에 대한 표현을 동시에 학습한다. 본 논문에서는 카테고리 분류와 유사도 모델을 혼합하여 부적합 문단을 추출하고자 하는데, 카테고리 분류에 대해서는 Doc2Vec과 Word2Vec을 혼합하여 분류율을 비교한 김도우[4]의 연구가 있다. 또한, Word2Vec을 이용하여 문서간 유사도를 비교한 사례가 있다[5].

3. 제안 시스템

본 논문에서는 Word2Vec을 통한 word Embedding을 통한 유사도 모델과, 공백으로 토큰화 한 후 tf-dif 가중치 알고리즘과 Logistic regression를 적용한 분포도 모델을 통하여 부적합 문단을 검출하는 모델을 제안한다.

제안 모델의 구조는 아래 [그림 1]과 같으며, 시스템은 크게 카테고리 분류 부분과 유사도 분석 부분으로 나뉜다.



[그림 1] 시스템 구조도

문서에 비해 단어의 수가 상대적으로 부족한 문단 단위의 분류를 무리 없게 수행하기 위해서는 충분한 데이터 셋의 구축이 필요하다. 본 실험에서는 조선일보에서 2017년 8월 ~ 2007년 3월 일자까지 수집한 뉴스 데이터를 사용했다. 학습 데이터 셋은 각 카테고리 별 5만 건의 뉴스 기사로 이루어진 30만 건의 뉴스 데이터로 구축하였다.

분포도 모델의 학습에 사용 될 레이블은 뉴스 카테고리를 [0]스포츠, [1]정치, [2]경제, [3]사회, [4]연예/방송, [5]오피니언/칼럼/사설로 나누어 0~5의 식별자를 부여했다. 학습된 분포도 모델에 입력 벡터를 입력하면 각 문단에 대한 레이블을 반환받을 수 있다. 여기서 입력 벡터는 기사 제목, 본문으로 구성된다. 제안 모델에서는 카테고리 분류를 위하여 각 문단에서 반환된 레이블과 문단의 수를 이용한 분포도를 계산한다. 예를 들어, 분포도 모델에서 입력 벡터를 분석한 결과가 아래 [그림 2]와 같다고 가정하자. 부적합 문단을 검출하기 위해서는 본문의 레이블 분포 파악이 필요하다. 우선 시스템에서는 각 레이블에 포함되는 문단의 개수를 전체 문단의 개수로 나누어 레이블별 분포율을 구한다. [그림 2]의 예제에서 총 문단 수는 4개이고, 그 중 레이블이 0인 문단이 3개, 레이블이 4인 문단이 1개이다. 그러므로 [0]스포츠에 대한 분포율은 0.75, [4]연예/방송에 분포율은 0.25가 되고, 문단 A와 B, C는 0.75, 문단 D는 0.25의 분포율을 가진다.

손흥민(토트넘)이 유럽축구연맹(UEFA)챔피언스리그에서 시즌 첫 골을 폭발시켰다. [문단A:레이블0]
손흥민은 14일(한국시간) 오전 영국 런던 웸블리 스타디움에서 열린 2017-2018시 UEFA 챔피언스리그 조별리그 1차전 도르트문트(독일)와 홈 경기에서 득점포를 가동했다. [문단B:레이블 0]
이날 선발 출격한 손흥민은 0-0이던 전반 4분 하프라인 아래에서 해리 케인의 패스를 받은 뒤 도르트문트의 왼쪽 진영을 뚫었다. [문단C:레이블 0]
'군함도'는 영화 '베테랑'을 만든 류승완 감독 작품이다. 송중기, 황정민, 소지섭, 이정현 씨 등 인지도 높은 배우들이 출연해 개봉 전부터 주목받았다. [문단D:레이블 4]

[그림 2] 가짜 뉴스의 예시

분포도 모델은 토큰화된 뉴스 데이터를 tf-dif 가중치 알고리즘을 적용하여 임베딩된 벡터와 레이블을 Scikit-learn library[6]에서 제공하는 로지스틱회귀분석(logistic regression) 모델을 통해 학습을 하였다. 김도우의 실험을 참고하면 Doc2Vec 모델만을 이용했을 때, 학습되는 data와 vector의 차원을 조절하여도 카테고리 분류율이 고정되는 문제점이 있다. 이에 본 연구에서는 선형 분류 문제를 해결하기 위해 단순하면서도 보다 강력한 분류 알고리즘인 로지스틱회귀분석을 함께 사용하였다. 우선, 토큰화 과정에서 Doc2Vec은 twitter 형태소 분석기로 추출한 명사만을 학습하였다. Doc2Vec을 이용하여 임베딩 된 Vector와 레이블을 로지스틱회귀분석 모델로 학습시켜 얻어낸 분류율은 83.6%로 나타났다. 그러나 뉴스 기사의 경우 명사 외의 형태소도 카테고리 분류에 영향을 미칠 수 있다. 그에 따라 공백으로 토큰화하여 얻은 모든 형태소를 tf-dif 가중치[7] 알고리즘으로 추출한 Vector와 레이블을 로지스틱회귀분석 모델로 학습시켜 87.0%의 카테고리 분류율을 얻을 수 있었다.

유사도 모델은 형태소 분석된 뉴스 기사 본문 각 문단

에 존재하는 명사 벡터 사이의 유사성을 기준으로 문단의 유사도를 계산한다. 입력받은 각 n개의 문단에 대하여 대상 문단을 제외한 다른 문단과의 유사도를 구하고, 얻어진 유사도들의 평균값을 각 문단의 유사도로 부여한다. 예를 들어 [그림 1]에서 문단 1과 문단2의 유사도를 sim(1,2), 문단1과 문단3의 유사도를 sim(1,3), 문단1과 문단4의 유사도를 sim(1,4)라 하면, 문단 1의 평균 유사도는 sim(1,2)+sim(1,3)+sim(1,4)/3이 된다.

제안하는 시스템에서는 분포도 모델과 유사도 모델을 통해 나온 문단별 분포도와 유사도를 결합하여 적합도를 계산한다. 적합도 = 분포도×유사도이며, 본문에서 적합도가 가장 낮은 문단을 부적합 문단으로 판단하기로 한다.

예를 들어, [그림 2]의 예제를 이용하여 유사도 모델을 적용한 결과 값은 아래 [그림 3]과 같다.

	문단A	문단B	문단C	문단D
문단A	1	0.12	0.11	0.06
문단B	0.12	1	0.14	0.07
문단C	0.11	0.14	1	0.06
문단D	0.06	0.07	0.06	1

[그림 3] [그림 2]예제에 대한 유사도 실험 결과

분포도 모델에서는 각 문단에 대하여 0, 0, 0, 3의 레이블을 반환받았으며, 각 모델에서 얻은 분포도와 유사도로 도출한 적합도는 문단A의 경우 0.72, 문단B의 경우 0.825, 문단C의 경우 0.77, 문단D의경우 0.15이다.

위 실험 결과를 통해 제안 모델에서는 부적합한 문단의 적합도가 상대적으로 낮게 나오는 것을 알 수 있다. 그러나 적합도가 낮다고 해서 모두 부적합 문단으로 간주하기는 어렵다. 그러므로 본 논문에서는 문단별 부적합도를 측정하여 결과값을 출력한다.

4. 실험 및 결과

4.1 실험 데이터 구성

테스트 데이터 셋은 각 카테고리에 대해 조선일보에서 2017년 9월 뉴스를 수집한 뒤 부적합 문단이 포함되지 않은 1만 건의 데이터와 임의로 타 카테고리 기사의 한 문단을 포함시킨 1만 건의 데이터로 구축하였다. 부적합 문단이 포함되지 않은 뉴스는 43218개의 문단, 부적합 문단이 포함 된 문단은 43214개로 이루어져 있다.

4.2 학습 결과

유사도 모델의 경우 각 문단에 포함된 모든 명사간의 유사도를 조사한 평균값을 문단 간의 유사도로 설정하였다. 86432개의 테스트 셋에 대하여 평균 72%의 정확도를 나타냈다. tf-dif 가중치 알고리즘을 적용하여 임베딩된 벡터에 대해 scikit-learn 지도학습을 활용한 분포도 모델의 분류율은 문단에 대한 레이블 반환 값에 대한 단순 비교로 측정하였으며, 86432개의 테스트 셋에 대하여 평균 87.0%의 정확도를 나타냈다. 반환 레이블에 대해서는 모든 레이블과 비교하여 가장 높은 확률을 기록한 레이블을 채택하였다.

4.3 평가 결과 분석

본 시스템의 정확도를 평가하기 위해 검출 분야에서 사용되고 있는 평가방식으로 정확성(Accuracy)을 이용하여 제안된 알고리즘의 성능을 평가한다. 본 시스템의 실험 결과는 다음 [표 1]와 같다.

[표 1] 테스트 셋에 대한 실험 결과

		label	
		True	False
Prediction	True	TP : 38896건	FP : 4422건
	False	FN : 4322 건	TN :38792건

본 시스템의 Accuracy(정확성)는 $\frac{tp+tn}{tp+tn+fp+fn}$ 수식을 통하여 89.88%의 결과를 볼 수 있다.

5. 결론

본 논문에서는 가짜 뉴스에서 부적합한 문단 검출을 위해 CNN모델의 Word2Vec과 공백을 이용한 토큰화, tf-dif 알고리즘을 적용하여 로지스틱 회귀를 이용한 새로운 방법을 시도했다. Doc2Vec을 통해 부적합 문단을 검출하는 방식만으로는 높은 정확도를 기대할 수 없었던 반면, 토큰화와 tf-dif 알고리즘과 로지스틱 회귀를 사용함으로써 기사의 불특정한 부분에서 나타나게 되는 부적합 문단 검출에 높은 성능을 기대 할 수 있게 되었다. 로지스틱 회귀를 이용한 문단별 카테고리 분류와 Word2Vec을 이용한 문단 간 유사도에 가중치를 부여하여 얻는 적합도 점수를 통한 부적합 문단 검출은 각 모델을 단독 사용했을 때보다 시스템 성능이 좋게 나타났다. 그리고 학습을 계속 진행할수록 결과가 좋게 나오는 것을 볼 수 있었다. 또한, 제안 모델은 본 논문에서 주제로 삼은 가짜 뉴스 부적합 문단 검출뿐만 아니라 자기소개서나 논설문 등 부적합한 문단이 포함될 수 있는 글 또는 시스템 전반에 적용 가능할 것으로 보인다.

향후 연구로는 Word2Vec의 성능 향상을 위한 모델링 개발과 RNN 연구를 함께 진행할 것이다.

참고문헌

[1] 권만우, 전용우, 임하진, "가짜뉴스(Fake News) 현황분석을 통해 본 디지털매체 시대의 쟁점과 뉴스콘텐츠 제작 가이드라인", 멀티미디어학회 논문지, 제18권, 제11호, 1419-1426, 2015

[2] Ye Zhang, Byron C. Wallace, "A Sensitivity Analysis of (and Practitioners' Guide to) Convolutional Neural Networks for Sentence Classification", arXiv:1510.03820, 2015.

[3] Yoon Kim, "Convolutional Neural Network for Sentence Classification", Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing(EMNLP), 2014.

[4] 김도우, "Doc2Vec을 활용한 CNN 기반 한국어 신문기사 분류에 관한 연구", 제28회 한글 및 한국어 정보처리 학술대회 논문집(2016년), 2016

[5] 황명권, 공현장, 황광수, 김판구. "문서의 계층화를 이용한 문서비교 방법." 한국정보과학회 학술발표논문집, 33.2B (2006.10): 143-147.

[6] http://scikit-learn.org/stable

[7] https://ko.wikipedia.org/wiki/TF-IDF

품사 부착 실험을 통한 Bags-of-Features 방법의 정량적 평가

이찬희○, 이설화, 임희석

고려대학교 정보대학 컴퓨터학과
chanhee0222@korea.ac.kr, whiteldark@korea.ac.kr, limhseok@korea.ac.kr

Quantitative Evaluation of Bags-of-Features Method Using Part-of-Speech Tagging

Chanhee Lee○, Seolhwa Lee, Heuiseok Lim
Department of Computer Science and Engineering, College of Informatics, Korea University

요 약

본 논문에서는 단순하지만 효과적인 단어 표현 방법인 Bags of Features에 대한 비교 실험을 수행한다. Bags of Features는 어휘집의 크기에 제한이 없으며, 문자 단위의 정보를 반영하고, 벡터화 과정에서 신경망 구조에 의존하지 않는 단어 표현 방법이다. 영어 품사 부착 실험을 사용하여 실험한 결과, one-hot 인코딩을 사용한 모델과 대비하여 학습 데이터에 존재하지 않는 단어의 경우 49.68%, 전체 부착 정확도는 0.96% 향상이 관찰되었다. 또한, Bags of Features를 사용한 모델은 기존의 영어 품사 부착 분야의 최첨단 모델들 중 학습 데이터 외의 추가적인 데이터를 활용하지 않는 모델들과 비견할 만한 성능을 보였다.

주제어: 자연어처리, 품사 부착

1. 서론

현재 단어를 입력으로 사용하는 많은 자연어처리 시스템(품사 부착, 단어 임베딩, 개체명 인식 등)은 단어를 독립적인 단위로 취급하고 one-hot 인코딩이나 lookup table을 이용하여 단어를 벡터로 변환한다[1,2]. 그러나 이러한 방법은 고정된 크기의 어휘집을 사전에 정의해야 하며, 모델의 파라미터 수가 어휘집의 크기에 따라 선형적으로 증가한다. 어휘집에 포함되지 못한 (Out-Of-Vocabulary, OOV) 단어도 추가적인 문제를 발생시킨다. 회귀 신경망이나 합성곱 신경망을 이용하여 문자 수준에서 단어를 처리함으로써 이러한 문제를 극복하는 방법도 있지만, 이는 추가적인 신경망 구조가 필요하므로 모델의 복잡도를 증가시킨다.

이찬희(2017)[3]의 연구에서는 bag of characters를 응용하여 어휘집에 제한이 없으며 신경망 구조에 의존하지 않고 문자 단위의 정보를 반영하는 단어 표현 방법인 Bags of Features(BOF)를 제안하였다. 본 연구에서는 BOF 방법을 이용하여 영어 품사 부착 실험을 수행한다.

2. 모델

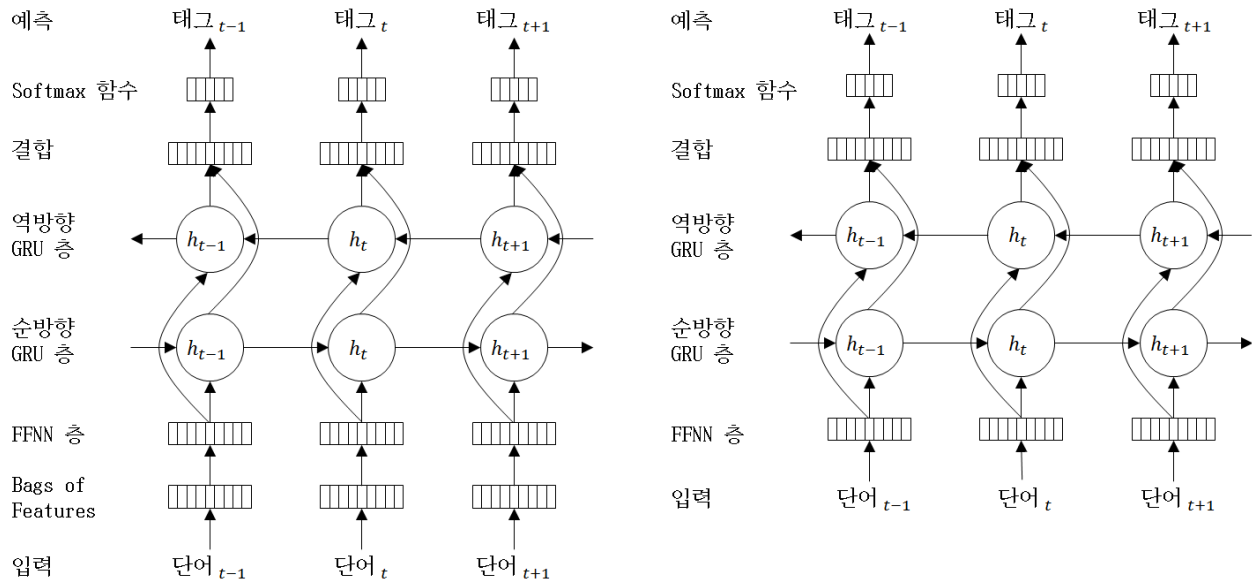
2.1. Bags of Features

이찬희(2017)[3]는 한국어를 기준으로 단어 표현 방법을 제안하였다. 한국어는 교착어이므로 품사 부착 이전에 형태소 분석이 우선적으로 수행되어야 하지만, 영어는 교착어에 속하지 않으므로 이러한 추가적인 과정이 필요하지 않다. 따라서 본 연구에서는 사전 실험으로써 영어를 대상으로 BOF 방법의 성능을 평가한다.

이찬희(2017)[3]의 연구에서 제안된 방법에 추가적으로, 영어의 특성을 반영한 대소문자 정보를 포함시켜 실험을 수행하였다. Collobert(2011)[4]에서 제안된 방법에 따라, 단어를 [*모두 대문자, 모두 소문자, 첫 글자 대문자, 기타*] 중 하나로 분류하여 이를 one-hot 인코딩을 사용하여 벡터로 변환 후 BOF 벡터에 결합하는 방법으로 이를 구현하였다.

2.2. 양방향 회귀 신경망

품사 부착과 같은 순열 분류 작업에서는 중심 단어의 앞 뒤 단어 정보에 접근 가능하다. 양방향 회귀 신경망[5]은 특정 입력에 대하여 이전 및 이후 단어 정보를 효과적으로 활용할 수 있다. 순수한 회귀 신경망은 장거리 의존성 문제에 취약하므로, 본 연구에서는 이 문제를 개선시킨 Gated Recurrent Unit(GRU)[6]을 사용한다. 모델의 뼈대는 GRU를 이용한 다층 회귀 신경망이며, 각 층 사이에는 Dropout[7]을 적용하였다. 양방향 회귀 신경망의 출력에 Feed-Forward Neural Network(FFNN)을 적용한 후, softmax 함수를 이용하여 품사들에 대한 확률 분포를 얻는다. 입력과 회귀 신경망 사이에도 FFNN 층들을 추가하여 더 높은 차원의 자질이 추출될 수 있도록 하였으며, 이 FFNN 층에는 Dropout을 적용하지 않는다. 이러한 모델의 구조는 [그림 1](a)에 나타나 있다.



(a) Bags of Features 사용 모델

(b) One-hot 인코딩 사용 모델

그림 1: 품사 부착 실험에 사용된 모델의 구조도. 점선은 Dropout의 적용을 나타냄. 도식의 단순화를 위하여 GRU 및 FFNN 층들은 다수의 층을 하나로 축약하여 표시함. (a) Bags of Features를 이용한 품사 부착 모델. (b) One-hot 인코딩을 이용한 품사 부착 모델.

2.3. One-hot 인코딩

BOF를 이용한 모델과의 성능 비교를 위하여 단어의 벡터화에 one-hot 인코딩을 사용하는 모델을 제작하였다. 이 모델의 학습 시에는 학습 데이터에 등장한 모든 어휘가 어휘집에 포함되도록 하였으며, 개발 또는 평가 데이터에만 등장하는 단어는 OOV를 나타내는 특수 단어로 치환하였다. 이 모델의 구조는 [그림 1](b)에 나타나 있다.

3. 실험 설계

3.1. Penn TreeBank 말뭉치

본 실험에서는 영어 품사 부착 실험에서 가장 널리 쓰이는 말뭉치인 Penn TreeBank(PTB)의 Wall Street Journal(WSJ)부분을 사용하였으며, 표준에 따라 0-18항은 학습, 19-21항은 개발, 22-24항은 평가 데이터로 활용하였다. 해당 말뭉치는 총 45 종류의 품사 중 하나로 단어들이 구분되어 있다.

3.2. 실험 방법

실험에 사용된 모델의 학습에는 경사 하강법을 바탕으로 한 최적화 알고리즘인 Adam[8]을 사용하였으며, 이를 이용하여 cross-entropy 손실 함수를 최소화 시켰다. 모든 모델의 구현에는 TensorFlow 라이브러리[9]를 사용하였다.

모든 FFNN 층 및 GRU 층에는 512개의 은닉 노드를 사용하였다. BOF를 사용한 모델에는 입력과 회귀 신경망 사이에 3개의 FFNN 층을 추가하였다. One-hot 인코딩을 사용한 모델의 단어 임베딩은 무작위로 초기화된 512차원 벡터가 사용되었다. 모든 모델은 20 에포크 동안 학습되었으며, 배치 크기는 64로 실험하였다.

4. 실험 결과

[표 1]은 BOF를 사용한 모델과 one-hot 인코딩을 사용한 모델의 품사 부착 정확도를 정리한 것이다(PTB WSJ 말뭉치의 평가 데이터 기준). One-hot 인코딩을 이용한 모델은 OOV 단어에 대해 매우 낮은 성능을 보인다. 이는 모델이 OOV 단어 자체에 대한 정보 없이 주변 단어를 기반으로 품사를 추정해야 함에 따른 것으로 사료된다. 반면 BOF를 이용한 모델은 단어를 구성하는 문자 정보를 활용할 수 있으며, 이는 정량적 실험 결과로도 확인할 수 있다. BOF를 이용한 모델은 one-hot 인코딩 이용 모델과 비교하여 OOV 단어는 49.68%, 전체 단어는 0.96% 향상된 성능을 기록하였다.

또 한가지 주목할 점은 모델에 사용된 파라미터의 수이다. BOF를 이용한 모델은 성능이 우수할 뿐만 아니라 사용된 파라미터의 수도 one-hot 인코딩 이용 모델의 32.96%에 불과하다. 추가적으로, one-hot 인코딩을 사용하면 어휘집의 크기와 비례하여 파라미터의 수가 증가하지만 BOF를 사용할 경우 어휘집의 크기와 무관하게 파라미터의 수가 동일한 수준으로 유지된다.

[표 1] PTB WSJ 말뭉치의 평가 데이터를 기준으로 한 품사 부착 정확도. 전체: 모든 단어에 대한 정확도. OOV: OOV 단어에 대한 정확도. 파라미터: 모델의 파라미터 수.

모델	전체	OOV	파라미터
One-hot 인코딩	96.16	57.71	32,119K
Bags of Features	**97.08**	**86.38**	10,585K

[표 2]는 본 연구의 실험 결과와 기존의 영어 품사 부착 연구들의 결과 비교이다. '추가' 열은 해당 연구에서 PTB WSJ 학습 데이터 외의 추가적인 데이터를 사용했

[표 2] 기존 연구들의 품사 부착 정확도(PTB WSJ 말뭉치의 평가 데이터 기준). 전체: 모든 단어에 대한 정확도. OOV: OOV 단어에 대한 정확도. 추가: 모델의 학습에 PTB WSJ 말뭉치 외의 데이터를 활용했는지 여부.

모델	전체	OOV	추가
Manning (2011)	97.32	90.79	Yes
Shen (2007)	97.33	89.61	No
Sun (2014)	97.36	-	No
Moore (2015)	97.36	91.09	Yes
Hajič (2009)	97.44	-	Yes
Søgaard (2011)	97.50	-	Yes
Tsuboi (2014)	97.51	91.64	Yes
Huang (2015)	97.55	-	Yes
Choi (2016)	**97.64**	**92.03**	Yes
본 연구	97.08	86.38	No

는지 여부를 나타내는데, 최근 최첨단 자연어처리 시스템들에서 널리 사용되는 단어 임베딩이 추가 데이터 활용의 대표적인 예이다. 본 연구에서의 실험 결과는 모델의 파라미터 학습에 추가적인 데이터를 활용하지 않는 연구 방법들과 비견할 만한 성능을 보였다.

Collobert(2011)[4]에 따르면, 추가 데이터를 활용한 단어 임베딩의 적용은 모델의 성능에 큰 향상을 가져온다. 본 연구에서 제안된 모델에도 마찬가지로 단어 임베딩을 적용하여 성능을 향상시킬 수 있을 것으로 기대되며, 이는 향후 추가 연구로 수행할 수 있을 것이다.

5. 결론

본 연구에서는 단어를 고정 길이 벡터로 변환하는 단순하면서도 효과적인 방법인 Bag of Features를 이용하여 영어 품사 부착 모델을 구현하고 성능을 정량적으로 비교 평가하였다. BOF 방법은 문자 수준에서 동작하므로 사전에 정의된 한정적인 어휘집이 필요하지 않다. 또한, 문자 단위로 단어를 처리하는 기존의 방법과 달리 합성곱 신경망 혹은 회귀 신경망과 같은 추가적인 구조가 요구되지 않는다. 영어 품사 부착 실험 결과, one-hot 인코딩을 사용한 비교 모델과 대비하여 OOV 단어에 대한 품사 부착 정확도에서 49.68%의 성능 향상을 보였으며, 모든 단어에 대한 품사 부착 정확도 또한 0.96% 상승함을 관찰하였다. 기존의 영어 품사 부착 연구들과의 비교에서도 추가 데이터를 활용하지 않는 모델들과 비견할 만한 성능을 나타냄을 확인할 수 있었다.

Acknowledgement

본 연구는 문화체육관광부 및 한국콘텐츠 진흥원의 2017년도 문화기술 연구개발 지원 사업으로 수행되었음. [2017. 스마트 시니어세대의 문화향유를 위한 인지반응 맞춤형 UI /UX기술 개발]

참고문헌

[1] T. Mikolov, I. Sutskever, K. Chen, G. S Corrado, and J. Dean. Distributed representations of words and phrases and their compositionality. In Advances in neural information processing systems. pages 3111-3119, 2013.

[2] Z. Huang, W. Xu, and K. Yu. Bidirectional lstm-crf models for sequence tagging. arXiv preprint arXiv:1508.01991, 2015.

[3] 이찬희, 이설화, 임희석. "Bag of Characters를 응용한 단어의 벡터 표현 생성 방법", 한국컴퓨터교육학회 하계학술대회 학술발표 논문집, 제21권, 제2호, pp. 47-49, 2017.

[4] R. Collobert, J. Weston, L. Bottou, M. Karlen, K. Kavukcuoglu, and P. Kuksa. Natural language processing (almost) from scratch. Journal of Machine Learning Research 12(Aug):2493-2537, 2011.

[5] A. Graves and J. Schmidhuber. Frame-wise phoneme classification with bidirectional lstm and other neural network architectures. Neural Networks 18(5):602-610, 2005.

[6] K. Cho, B. V. Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio. Learning phrase representations using rnn encoder-decoder for statistical machine translation. arXiv preprint arXiv:1406.1078, 2014.

[7] N. Srivastava, G. E Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. Journal of Machine Learning Research 15(1):1929-1958, 2014.

[8] D. Kingma and J. Ba. Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980, 2014.

[9] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S Corrado, A. Davis, J. Dean, M. Devin. Tensorflow: Large-scale machine learning on heterogeneous distributed systems. arXiv preprint arXiv:1603.04467, 2016.

한국어 상대시간관계 추출을 위한 LSTM 기반 모델 설계

임채균[1], 정영섭[2], 이영준[1], 오교중[1], 최호진[1]

[1]한국과학기술원 전산학부
{rayote, yj2961, aomaru, hojinc}@kaist.ac.kr

[2]순천향대학교 빅데이터공학과

bytecell@sch.ac.kr

Design of LSTM-based Model for Extracting Relative Temporal Relations for Korean Texts

Chae-Gyun Lim[1], Young-Seob Jeong[2], Young Jun Lee[1], Kyo-Joong Oh[1], Ho-Jin Choi[1]
[1]School of Computing, KAIST
[2]Department of BigData Engineering, Soonchunhyang University

요 약

시간정보추출 연구는 자연어 문장으로부터 대화의 문맥과 상황을 파악하고 사용자의 의도에 적합한 서비스를 제공하는데 중요한 역할을 하지만, 한국어의 고유한 언어적 특성으로 인해 한국어 텍스트에서는 개체간의 시간관계를 정확하게 인식하기 어려운 경향이 있다. 특히, 시간표현이나 사건에 대한 상대적인 시간관계는 시간 문맥을 체계적으로 파악하기 위해 중요한 개념이다. 본 논문에서는 한국어 자연어 문장에서 상대적인 시간표현과 사건 간의 관계를 추출하기 위한 LSTM(long short-term memory) 기반의 상대시간관계 추출 모델을 제안한다. 시간정보추출 연구에는 TIMEX3, EVENT, TLINK 추출의 세 가지 과정이 포함되지만, 본 논문에서는 특정 문장에 대해서 이미 추출된 TIMEX3 및 EVENT 개체를 제공하고 상대시간관계 TLINK를 추출하는 것만을 목표로 한다. 또한, 사람이 직접 태깅한 한국어 시간정보 주석 말뭉치를 대상으로 LSTM 기반 제안모델들의 상대적 시간관계 추출 성능을 비교한다.

주제어: 시간정보추출, 상대적 시간관계, 시간 표현, LSTM

1. 서론

시간정보추출에 관한 연구는 자연어 입력데이터로부터 시간이나 사건 표현을 추출할 뿐만 아니라 그 표현들 사이의 시간적 관계를 발견하는 것을 포함한다. 시간에 관한 문맥 정보는 자연어 표현에 암시된 의미적 특징을 포착하기 위해 이용될 수 있기 때문에, 시간정보를 추출하는 연구는 질의응답시스템 또는 구조화되지 않은 텍스트에 대한 처리를 하는 응용 등에서 중요한 비중을 차지한다. 이러한 연구가 전세계적으로 진행되면서 SemEval-2013의 shared task인 TempEval-3 [1]이 널리 알려져 있다. TempEval-3는 다량의 영어 문서에 대한 시간 및 사건 표현, 시간관계 추출에 대한 모델을 제시하고 성능을 평가한다. 최근에는 임상 도메인에서 시간정보를 추출하는 데 중점을 둔 SemEval의 다른 shared task인 Clinical TempEval [2]이 지속적으로 추진되고 있다.

그러나, TempEval과 같은 시간정보를 분석하는 연구는 대부분 영어를 대상으로 하며, 한국어 문서에 대한 시간정보추출 연구는 많이 이루어지지 않았다. 또한, 한국어에서 시간정보를 추출하기 위해 의존구문분석 트리(dependency parse tree)나 POS 태그와 같은 언어학적 정보가 자주 활용되지만, 한국어에 대한 언어분석 성능은 아직 충분하지 않다. 시간(TIMEX3)이나 사건(EVENT) 추출에 비해서 시간적 관계(TLINK)를 찾을 때 문장 내 의미론적 특성이 더욱 중요할 뿐만 아니라, 상대적인 시간관계를 이해하려면 시간 문맥도 파악할 필요가 있다. 언어분석 결과의 오류가 파급된다면 시간정보추출의 전반적인 성능 저하에 영향을 미친다.

본 논문에서는 LSTM(long short-term memory)을 적용한 신경망을 기반으로 한국어 문장에서 상대적인 시간관계를 나타내는 TLINK를 추출하는 딥러닝 모델을 제안한다. 시간정보추출 작업 중에서 TLINK 추출만을 대상으로 하는 모델을 설계하고, 시간관계 형성에 필요한 TIMEX3 및 EVENT 개체가 이미 추출되었다고 가정하여 모델의 입력으로 제공한다. 또한, LSTM 기반 모델의 입력 벡터를 생성하기 위해 전체 한국어 시간정보 주석 말뭉치를 분석하여 워드임베딩 모델을 구성하며, 이를 사용하여 TIMEX3 및 EVENT 개체를 임베딩 벡터로 변환한다. 이러한 TIMEX3/EVENT 개체의 임베딩 벡터들은 서로 다른 구조를 지닌 3가지 상대시간관계 추출 모델들을 학습할 때 활용된다. 3개의 모델은 서로 다른 개수의 LSTM 레이어로 이루어져 있으며, 실험을 통해 제안모델들의 상대시간관계 추출 성능의 차이를 비교한다.

본 논문은 다음과 같이 구성된다. 2장에서는 말뭉치로부터 시간관계를 추출하는 기존 연구를 소개하고, 3장에서는 제안하는 LSTM 기반 상대시간관계 추출 모델에 대

해 자세히 설명하며, 4장에서는 제안모델의 성능에 대한 실험결과를 나타내며, 5장에서 결론을 맺는다.

2. 관련 연구

시간정보추출 분야에서 딥러닝 모델을 활용하여 시간관계를 추출하려는 여러 연구들이 있다. Peng Zhou(2016)는 주어진 입력 문장 내에서 중요한 정보들이 아무 곳에나 위치하는 문제를 해결하기 위해 Attention-Based Bidirectional LSTM Networks 모델을 제안하였으며 F1-score 84%의 성능을 보였다[3]. Few Cheng(2017)은 관계 추출에 뛰어난 결과를 나타내는 DP(dependency path) 기반의 신경망 모델을 시간관계 분류에 적용하였고 F1-score 54%의 성능을 확인하였다[4]. Pengda Qin(2017)은 관계 분류에서 여러 개체들 간의 semantic knowledge가 완전히 활용되지 않는 문제를 해결하기 위해 Entity-pair-based Attention Mechanism 를 제안하였고 F1-score 84.7%의 결과를 산출하였다[5]. Julien Tourille(2017) 은 narrative container identification 을 위한 신경망 모델을 제안하였다. 이 모델은 최근에 입력되는 데이터에 편향되는 경향이 있는 LSTM의 특성에 대응하기 위해 문장을 역순으로 읽는 LSTM을 추가로 활용한다. 두 개의 LSTM을 통해서 주어진 입력에 대한 많은 정보를 학습할 수 있으며, 이 모델은 F1-score 61.3%의 성능을 보였다[6]. Yuanliang Men(2017)은 개체들 간의 시간관계를 복원하기 위해 shortest DP(dependency path)를 이용한 LSTM 기반 구조를 제안하였고 F1-score 47%의 결과를 나타냈다[7].

본 논문에서는 LSTM 기반 딥러닝 모델을 활용하여 TIMEX3과 EVENT 개체들 사이의 상대적인 시간관계를 학습한다. 기존의 연구는 주로 영어 코퍼스를 대상으로 시간관계를 추출하는 목적을 가지고 있지만, 본 논문에서는 한국어 코퍼스를 집중적으로 분석하고 한국어 특성을 반영하여 시간 및 사건 표현 간의 관계를 추출하는 것이 목표이다.

3. 제안 기법

본 논문에서는 한국어 문장에서 시간 표현과 사건 간의 상대적인 시간관계를 추출하기 위해 TLINK에서 잠재성을 파악할 수 있는 LSTM 기반의 딥러닝 모델을 제안한다. 본 논문에서 제안하는 상대적 TLINK 추출 모델의 전체 디자인은 그림 1과 같다. 일반적으로, 시간정보추출은 주어진 말뭉치의 자연어 문장으로부터 TIMEX3 및 EVENT개체들을 추출하고, 그 개체들 사이의 시간적 관계를 발견하여 TLINK 개체로 추출하는 과정으로 이루어진다. 그러나, 본 논문에서는 딥러닝 모델에 기반한 TLINK 추출이라는 목적에 집중하기 위해서 TIMEX3 및 EVENT 개체들을 이미 추출한 상태라고 가정하고 제안모델의 입력으로써 제공한다. 먼저, 한국어 시간정보 주석 말뭉치에 포함된 자연어 문장으로부터 워드임베딩 모델을 학습한다. 각각의 문장은 단일 문자 단위로 토큰화되어 워드임베딩 모델 학습에 사용되며, 임베딩 벡터는 100차원 공간으로 구성된다. 그 후, 워드임베딩 모델을 사용하여 TIMEX3 또는 EVENT 태그의 범위 내에 포함된 텍스트를 벡터화한다.

TLINK 태그는 관계유형 중 하나를 의미하는 relType 속성과 시간관계를 갖는 2개의 다른 개체들에 대한 참조로 구성된다. 따라서, 임베딩 벡터는 시간관계를 형성하는 2개의 개체에 대해 개별적으로 생성해야 하며, 이들

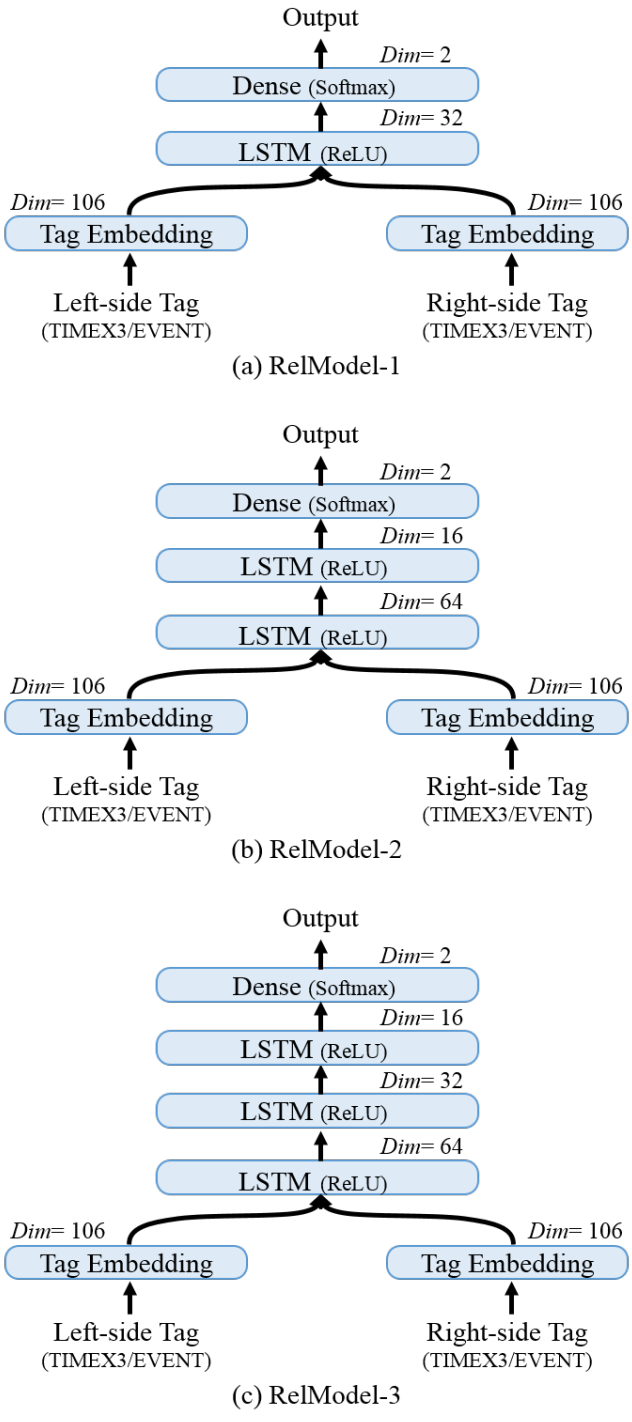


그림 1. 상대적 시간관계 추출을 위한 LSTM 기반 딥러닝 모델

을 하나의 벡터로 연결(concatenate)하여 제안모델의 입력으로 사용한다. 주어진 개체의 임베딩 벡터는 TIMEX3 또는 EVENT의 특정 클래스를 나타내도록 one-hot encoding된 벡터와 태그 범위의 텍스트로부터 얻어진 워드임베딩 벡터로 구성된다. 각 태그의 임베딩 벡터는 해당 개체의 클래스에 대한 6차원 벡터와 워드임베딩의 100차원 벡터를 결합하여 총 106차원을 가진다. 이러한 방식으로 TLINK 내의 2개 개체에 대한 임베딩 벡터를 생성한 후에, 하나로 연결하여 212차원의 벡터를 제안모델의 입력으로써 사용한다.

LSTM 레이어에서는 *tanh* 함수와 *ReLU*를 activation function으로써 사용한다. 그림 1(a)의 RelModel-1는 1개의 LSTM 레이어를 사용하고, 32차원의 벡터를 출력한다. 그림 1(b)의 RelModel-2는 2개의 LSTM 레이어들로 이루어져 있으며, 첫 레이어에서 64차원 벡터, 두 번째 레이어에서 16차원 벡터를 출력한다. 그림 1(c)의 RelModel-3은 3개의 LSTM 레이어들로 구성되어 있고, 첫 레이어에서 64차원 벡터, 두 번째 레이어에서 32차원 벡터, 마지막 세 번째 레이어에서 16차원 벡터를 출력한다.

그 다음 Dense 레이어는 activation function으로 *softmax* 함수를 사용하여 2차원 벡터의 형태로 최종 결과를 생성한다. 제안모델의 출력 벡터는 주어진 2개의 시간 또는 사건 개체들이 상대적인 TLINK를 형성하는지 여부에 대해서 one-hot encoding된 벡터이다. 이 출력 벡터에서 최대값을 가지는 쪽을 선택하여 TLINK 형성 여부를 결정한다.

4. 실험

본 논문에서 사용한 데이터셋은 2393건의 문서와 6190개의 한국어 문장을 포함하고 있는 *Korean TimeBank* [8]이다. 이 데이터에서 TIMEX3, EVENT 및 TLINK 개체의 수는 각각 3290, 17547, 4545이고, 상대적인 시간관계를 가진 TLINK 개체는 333개이다. LSTM 기반 TLINK 추출 모델을 구현하기 위해 Windows OS 환경에서 Keras 2.0.6 라이브러리와 Python 3.5.3을 사용하였다. 문자 기반 워드임베딩 모델의 어휘사전 크기는 2076이다.

제안모델의 학습 과정에서 상대적 관계를 가진 TLINK 개체를 모두 사용하고, 동일한 개수의 TLINK가 아닌 개체의 쌍을 샘플링하였다. 따라서, TLINK 및 비 TLINK를 포함한 데이터의 총 개수는 666이 된다. 전체 데이터를 분할하여 90%는 학습 데이터로, 나머지 10%는 테스트 데이터로 사용하였다.

실험에서는 그림 1과 같이 LSTM 레이어의 개수를 달리하며 설계된 3가지 모델들에 대한 상대적 시간관계의 추출성능을 측정하였다. 표 1은 실험결과를 나타낸 것이며, 제안모델들의 F1-score는 평균적으로 0.75 수준이었다. RelModel-1과 RelModel-2의 경우에는 0.9 이상의 높은 성능을 보였던 반면, RelModel-3는 상대적으로 낮은 0.33의 결과를 보였다. 이 결과로부터 주어진 2개의 TIMEX3/EVENT 개체들에 대해서 상대적인 시간관계의 유무를 알아내는 데 너무 많은 레이어를 구성하는 것은 오히려 좋지 않다는 것을 알 수 있다. 특히, RelModel-3는

표 1. 상대적 시간관계 추출 성능

제안모델	Precision	Recall	F1-score
RelModel-1	0.95	0.94	0.94
RelModel-2	0.97	0.97	0.97
RelModel-3	0.25	0.50	0.33
Avg.	0.72	0.80	0.75

테스트 데이터 중에서 상대시간관계가 존재하는 경우에는 성공적으로 분류했으나, 관계를 포함하지 않는 데이터에 대해서 모두 잘못 분류하는 결과를 보였다. 이러한 실험결과는 RelModel-3가 상대시간관계에 대해 과적합(overfitting)되었을 가능성이 높음을 시사한다.

또한, RelModel-1과 RelModel-2의 실험결과에서도 상대시간관계가 없는 테스트 데이터에 대해서 잘못 분류하는 비율이 높은 경향을 보였다. 우리는 이 문제가 발생한 이유는 상대적 시간관계가 없는 TIMEX3-EVENT 조합의 유형이 너무 다양하기 때문이라고 추정하였다. 추후 상대적 시간관계의 조합에 대한 패턴을 분석하고 규칙 기반 접근법을 함께 활용하는 시도가 필요하다고 생각된다.

5. 결론

본 논문에서는 시간 문맥을 파악하고 한국어 문장에서 상대적인 시간관계를 추출하기 위한 LSTM 기반의 딥러닝 모델들을 제안하였다. 본 논문의 주요 목적은 주어진 한국어 코퍼스에서 TIMEX3과 EVENT 개체들 사이의 상대시간관계를 설명하기 위한 TLINK를 발견하는 것이므로, 이미 추출된 TIMEX3와 EVENT 개체를 입력으로 제공하여 TLINK 추출 모델에만 집중하였다. 실험에서는 서로 다른 구조를 가진 LSTM 기반 모델들의 상대적 시간관계 추출의 성능을 비교하였다.

향후 연구에서는 상대시간관계 TLINK 추출의 전반적 성능을 향상시키기 위해 한국어 시간정보 주석 말뭉치의 규모를 충분히 확장시킨 후, 상대적 시간관계의 패턴을 분석하고 모든 관계유형에 대한 추출 성능을 고려하는 실험을 수행할 계획이다.

감사의 글

본 연구는 미래창조과학부 산업융합원천기술개발사업 "휴먼 지식증강 서비스를 위한 지능진화형 WiseQA 플랫폼 기술 개발" (No.2013-0-00131)과 ICT유망기술개발지원사업 “지능형 대화 서비스를 위한 화용 및 문맥 분석 기반 대화솔루션 개발(No. 2017-0-00868)”의 지원으로 수행되었음.

참고문헌

[1] N. UzZaman, H. Llorens, J. Allen, L. Derczynski, M. Verhagen, and J. Pustejovsky, “Tempeval-3: Evaluating events, time expressions, and temporal

relations," *arXiv preprint arXiv:1206.5333*, 2012.
[2] S. Bethard, G. Savova, W.-T. Chen, L. Derczynski, J. Pustejovsky, and M. Verhagen, "Semeval-2016 task 12: Clinical tempeval," In *Proceedings of SemEval*, pp. 1052-1062, 2016.
[3] P. Zhou, W. Shi, J. Tian, Z. Qi, B. Li, H. Hao, and B. Xu, "Attention-based bidirectional long short-term memory networks for relation classification," In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*,Vol. 2, pp. 207-212, 2016.
[4] F. Cheng and Y. Miyao, Classifying Temporal Relations by Bidirectional LSTM over Dependency Paths, In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, Vol. 2, pp. 1-6, 2017.
[5] P. Qin, W. Xu, and J. Guo, "Designing an adaptive attention mechanism for relation classification," *Neural Networks (IJCNN), 2017 International Joint Conference on*, pp. 4356-4362, IEEE, 2017.
[6] J. Tourille, O. Ferret, A. Neveol, and X. Tannier, "Neural Architecture for Temporal Relation Extraction: A Bi-LSTM Approach for Detecting Narrative Containers," In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, Vol. 2, pp. 224-230, 2017.
[7] Y. Meng, A. Rumshisky, and A. Romanov, "Temporal Information Extraction for Question Answering Using Syntactic Dependencies in an LSTM-based Architecture," *arXiv preprint arXiv:1703.05851*, 2017.
[8] Y.-S. Jeong, W.-T. Joo, H.-W. Do, C.-G. Lim, K.-S. Choi, and H.-J. Choi, "Korean TimeML and Korean TimeBank," In *Proceedings of the 10th edition of the Language Resources and Evaluation Conference (LREC)*, pp. 356-359, 2016.

딥러닝을 이용한 전이 기반 한국어 형태소 분석 및 품사 태깅

민진우[○†], 나승훈[†], 김영길[††]

전북대학교[†], ETRI[††]
Jinwoomin4488@gmail.com, nash@jbnu.ac.kr, kimyk@etri.re.kr

A Transition based Joint Model for Korean Morpheme Segmentation and POS Tagging Using Deep Learning

Jin-Woo Min[○†], Seung-Hoon Na[†], Young-Kil Kim[††]
Chonbuk National University[†], ETRI[††]

요 약

한국어 형태소 분석은 많은 자연어 처리 분야에서 핵심적인 역할을 수행하고 있기 때문에 형태소를 분류하고 형태소에 맞는 알맞은 품사를 결정하는 것은 매우 중요하다. 형태소의 품사를 태깅하는 대표적인 방법은 크게 음절 단위 형태소 분석과 단어 단위 형태소 분석의 두 가지로 나눌 수 있다. 본 논문에서는 의존 파싱 분야에서 널리 활용되고 있는 전이 기반 방식을 적용하여 전이 기반 단어 단위 한국어 형태소 분석 모델을 제안하고 해당 모델을 한국어 형태소 분석 데이터인 세종 품사 부착 말뭉치 셋에 적용하여 F1 97.77 %로 기존의 성능을 더욱 향상시켰다.

주제어: 딥러닝, 형태소 분석, 품사 태깅, 전이 기반

1. 서론

형태소 분석은 많은 자연어 처리 분야에서 핵심적인 수행하고 있다. 한국어 형태소 분석은 일반적으로 형태소 분석과 품사 태깅의 두 가지의 과정으로 구분하며 형태소 분석은 문장 내의 어절을 뜻을 지니는 최소의 단위인 형태소로 분해하고 해당 형태소의 품사 후보를 생성하는 작업이고 품사 태깅은 위의 품사 후보로부터 가장 적절한 품사를 결정하는 것이다[1].

형태소 분석은 크게 음절 기반 형태소 분석과 단어 기반 형태소 분석으로 나뉠 수 있으며 음절 기반 형태소 분석은 입력된 문장을 음절 단위로 나누고 순차 레이블링 문제로 보고 [B(Begin), I(inside)] 혹은 [B, I, E(End), S(Single)]가 태그가 포함된 품사태그를 결정하는 방식이다. 반면, 형태소 기반 방식은 분할된 형태소에 직접 바로 태그를 부여하는 방식이다[9]. 한국어 형태소 분석에 대한 연구는 CRF(Conditional Random Fields), SVM(Support Vector Machine)[2,4]등 기존의 기계학습 방법이 주를 이루었으나 최근 들어 한국어 형태소 분석에서도 다양한 자연언어 처리에서 각광받고 있는 RNN 계열의 딥러닝 모델들[5-7,9]을 적용하는 연구가 많이 진행되고 있다.

의존 파싱 문제에서 널리 연구되고 있는 전이 기반 방식[8]은 입력에 대한 버퍼와 스택의 상태로부터 자질벡터들을 얻어 결합한 후 딥러닝 신경망을 통해 해당 전이 액션을 결정하는 방식이다.

본 논문에서는 의존 파싱에서 널리 활용되고 있는 전이 기반 방식을 한국어 형태소에 맞는 액션을 정의하고 액션에 의해 형태소를 분할하고 품사를 부여하는 형태소 기반 방식으로 딥러닝 모델에 적용하여 세종 품사 부착 말뭉치 셋에서 F1 97.77%으로 기존 모델보다 높은 성능을 보였다.

2. 관련 연구

한국어 품사 태깅에 대한 다양한 연구가 진행되었다. 음절 기반 한국어 형태소 분석은 주로 순차 레이블링 기반으로 연구가 진행되었는데 이러한 순차 레이블링의 기계학습 모델은 CRF, SVM 모델 등이 있다. [4]에서는 Structural SVM를 활용하여 한국어 띄어쓰기 및 품가 태깅 결합 모델을 제안하였다. [2]에서는 CRF에 기반한 형태소 분석 모델을 제안하였으며 1) 형태소 분할 단계, 2) 품사 태깅 단계, 3) 복합 형태소 분할 및 태깅 단계의 세 단계로 품사 태깅을 진행한다.

딥러닝을 이용한 품사 태깅 연구도 진행되었는데 [5]에서는 음절 기반으로 형태소 분석을 진행하였으며 품사 태그의 빈도수를 계산 한 후 softmax로 수치화하여 벡터의 값으로 활용하고 품사 태깅, 개체명 인식 등 순차 레이블링 문제에서 우수한 성능을 보이는 Bi-LSTM CRF 모델을 적용하였다.

Sequence-to-Sequence 모델은 임의 길이의 한 종류의 시퀀스를 다른 한 종류의 시퀀스로 변환하는 딥러닝 모델로 기계번역 분야에서 탁월한 성능을 보여주고 있다. [6]에서는 입력문장을 해당 형태소와 품사 태그로 번역하는 모델로 보고 Sequence-to-Sequence 모델을 한국어 형태소 분석 및 품사 태깅 문제에 적용하는 연구가 진행되었다. [7]에서는 Sequence-to-Sequence 모델

표 1. 전이 액션 별 스택 및 버퍼 정보의 갱신 과정

S_t	B_t	Action	S_{t+1}	B_{t+1}
S	c, B	$Split(t)$	$(t,c), S$	B
S	c, B	$Merge$	S	B

표 2. 형태소 분석 전이 액션의 실행 예

Action	S	B	$Tagging$
$init$	[]	[내,<SP>,고,향,은,<SP>,서,울,이,다,.]	…
$Split(MM)$	[내]	[<SP>,고,향,은,<SP>,서,울,이,다,.]	내/MM
$Merge$	[내]	[고,향,은,<SP>,서,울,이,다,.]	…
$Split(NNG)$	[내,고]	[향,은,<SP>,서,울,이,다,.]	고향/NNG
$Merge$	[내,고]	[은,<SP>,서,울,이,다,.]	…
$Split(JX)$	[내,고,은]	[<SP>,서,울,이,다,.]	은/JX
$Merge$	[내,고,은]	[서,울,이,다,.]	…
$Split(NNP)$	[내,고,은,서]	[울,이,다,.]	서울/NNP
$Merge$	[내,고,은,서]	[이,다,.]	…
$Split(VCP)$	[내,고,은,서,이]	[다,.]	이/VCP
$Split(EF)$	[내,고,은,서,다]	[.]	다/EF
$Split(SF)$	[내,고,은,서,다,.]	[]	./SF

을 확장하여 입력 열의 단어들이 출력 열에도 등장하는 경우 해당 단어들을 복사하는 Copying Mechanism을 활용하는 연구도 진행되었다.

[9]에서는 중국어 형태소 분할 문제를 전이 기반 방식으로 적용하여 현재 음절을 현재 스택이 가리키는 형태소에 부착하는 액션, 현재 스택의 형태소를 결정짓고 버퍼가 가리키는 음절을 스택으로 이동하여 새로운 형태소의 시작 음절로 하는 2가지 액션으로 적용하여 중국어 형태소 분할 문제에서 기존의 성능을 향상시켰다.

본 논문에서는 [9]의 전이 기반 형태소 분할 문제를 한국어 형태소 분석 및 품사 태깅 문제로 확장하여 적용하여 기존의 한국어 형태소 분석의 성능을 향상시킬 수 있음을 보인다.

3. 전이 기반 한국어 형태소 분석 모델

본 논문에서 사용한 모델의 구조는 [9]의 전이 기반 형태소 분할 문제를 한국어 형태소 분석 및 품사 태깅 문제로 확장하였으며 형태소 분할 및 태깅을 전이 액션으로 하고 이를 딥러닝을 통해 결정하는 모델이다.

3.1. 형태소 분할 및 품사 태깅을 위한 전이 액션

본 논문에서는 [9]에서의 전이 액션을 한국어 형태소 분석 및 품사 태깅 문제에 알맞게 확장하였다. 형태소 분할 및 품사 태깅을 위한 액션은 Split Action, Merge Action 2가지이고 역할은 다음과 같다.

• *Split* Action : 현재 스택에 존재하고 있는 형태소의 끝 경계를 결정 짓고 현재 버퍼가 가리키고 있는 음절을 스택에 PUSH한 다음 품사태그를 부여한 후 해당 형태소의 시작으로 하는 액션

• Merge Action : 현재 스택의 top이 가리키고 있는 형태소에 현재 버퍼가 가리키고 있는 음절을 해당 형태소의 구성요소로 추가하는 액션. 현재 버퍼가 Focus를 다음 음절로 이동하는 동작만을 수행

Split Action, Merge Action의 두 전이 액션 별 스택 및 버퍼 정보의 갱신 과정은 다음 표 1로 설명한다. 전이 액션을 위한 버퍼와 스택은 B, S로 표기하고 기호 c, t는 각각 음절과 품사 태그로 정의한다. Split Action이 이루어지면 버퍼에 있던 음절 c가 스택으로 PUSH되고 음절(형태소의 시작)에 품사 t가 부여됨을 알 수 있다. 다음으로 Merge Action이 실행되면 형태소에 해당 음절을 추가하는 액션으로 실제로 하는 동작은 버퍼의 Focus를 다음 음절로 이동하는 역할만을 하게 된다.

표 2는 형태소 분석에 대한 전이 액션이 이루어지는 과정을 보여준다. 표 2의 초기 상태의 버퍼를 보면 입력은 공백을 포함한 한국어 원문의 음절 열이며 *Split* Action 이 수행 되면 해당 형태소의 시작 음절의 위치에 품사 태그를 부여하는 것을 볼 수 있다. 형태소의 경계가 결정지어지는 것은 스택에 새로운 형태소의 시작 음절이 Push되는 시점이다. 예를 들어, 스택의 TOP에 형태소 "고향"이 있을 때 "의"라는 음절이 스택에 Push되면 형태소의 경계가 결정된다. 또한, 공백 음절 역시 Merge Action 액션이 수행되지만 실제 형태소에는 포함되지 않는다.

3.2. 버퍼의 입력 표상

버퍼의 입력 표상은 임력열 $\boldsymbol{x} = \{\boldsymbol{x}_1, \cdots, \boldsymbol{x}_n\}$로부터 LSTM을 통해 얻어지게 되는데 음절 임베딩 벡터 $\boldsymbol{x}_t$를 얻기 위하여 입력 문장의 t번째 음절을 c_t라 하고 해당 음절을 기준으로 자질을 추출하여 $\boldsymbol{x}_t$를 구성한다. 추출 자질 유형은 다음 표 3에 제시하며 임베딩을 얻는 과정은 수식 (1)로 표현한다.

표 3. 입력으로 사용되는 자질 유형

음절 자질	Explanation
Unigram	c_{t-1}, c_t, c_{t+1}
Bigram	$c_t c_{t+1}$
Trigram	$c_t c_{t+1} c_{t+2}$

$$\boldsymbol{x}_i = lookup([uni(i); bi(i); tri(i)]) \quad (1)$$

위의 수식에서 uni, bi, tri함수는 각각 표 3에 대응되어 n-gram 자질을 추출하는 함수이고 $lookup$함수는 임베딩 Lookup Table을 보고 해당 임베딩 벡터를 얻어내는 함수이다. 버퍼의 입력 표상은 다음 식 (2)과 같이 계산할 수 있다.

$$\{\boldsymbol{h}_1, \cdots, \boldsymbol{h}_n\} = LSTM(\{\boldsymbol{x}_1, \cdots, \boldsymbol{x}_n\}) \quad (2)$$

식 (2)에서 보듯이 입력 열 $\boldsymbol{x}$을 LSTM을 통해 언어낸 은닉 열 $\boldsymbol{h} = \{\boldsymbol{h}_1, \cdots, \boldsymbol{h}_n\}$을 버퍼의 입력 표상으로 사용하게 된다. 버퍼의 입력 표상을 얻기 위해 사용된 LSTM 신경망의 유닛은 512개가 사용되었고 각 자질의 차원은 32차원으로 입력 임베딩은 160차원이다.

3.3. 모델의 구조

본 논문에서 제안한 전이 기반 형태소 분석 모델의 구조는 다음 그림 1와 같다.

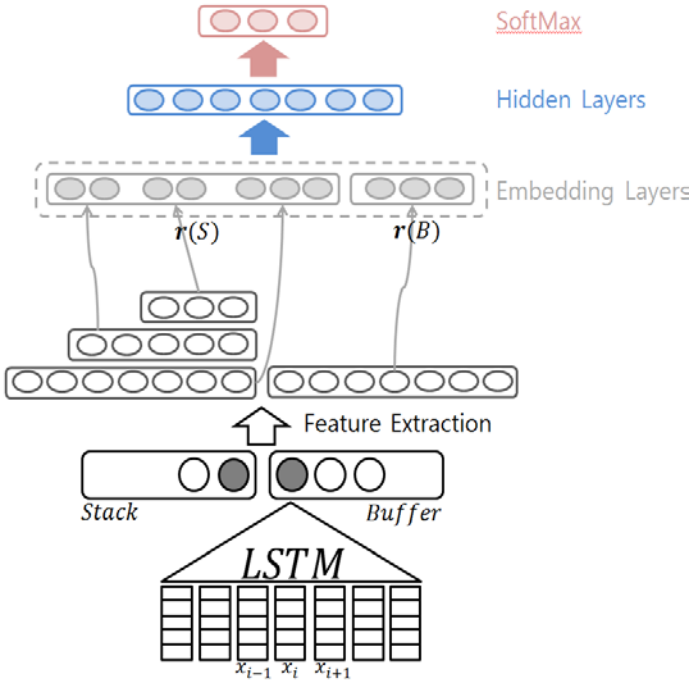


그림 1. 전이 기반 형태소 분석 모델의 구조

그림 1에서 보듯이 먼저 음절 단위로 LSTM을 통해 얻어진 은닉벡터들이 버퍼 B에 채워지게 된다. 전이 액션을 결정하기 위한 버퍼와 B, S의 해당 상태 벡터들을 각각 $\mathbf{r}(B)$, $\mathbf{r}(S)$라고 하자. S_0, S_1을 버퍼 혹은 스택의 TOP, 2번째 TOP 노드로 표현하고 해당 노드의 입력 음절에 대한 위치를 얻어내는 함수는 $cpos$이며 각 상태 백터는 다음 식 (3),(4)를 통해 얻어진다[1].

$$\mathbf{r}(B) = \boldsymbol{h}_{cpos(B_0)} \quad (3)$$

$$\mathbf{r}(S) = [\boldsymbol{h}_{cpos(S_0)}; lookup(tag(S_0)); lookup(mtag(S_1))] \quad (4)$$

식 (3)에서 보듯이 버퍼의 TOP 노드의 상태 백터 $\mathbf{r}(B)$은 단순히 입력 열 $\boldsymbol{x}$로부터 LSTM을 통해 얻어진 은닉열 $\boldsymbol{h}$에서 TOP 노드에 해당하는 위치의 은닉 벡터의 값을 취하게 된다. 식 (3)가 수행되는 과정을 표 2로 예를 들면 처음 $Merge$ Action이 수행되는 3번 째 줄 버퍼 B의 TOP노드는 "고"를 나타내고 문장 내 음절 열에서 위 음절은 세 번째에 위치하고 있으므로 LSTM을 통해 얻어진 은닉열 $\boldsymbol{h}$에서 세 번째 은닉 상태인 $\boldsymbol{h}_3$를 취하게 되는 것이다.

$\mathbf{r}(S)$는 식 (4)로 계산되며 은닉벡터를 취하는 것에 더하여 a노드에 해당하는 품사를 얻어내는 함수 tag(a)와 형태소 표층형과 해당 품사태그의 결합 정보를 얻어내는 함수인 mtag(a)를 각각 적용하여 얻어낸 후 각각 $lookup$함수를 통해 변환된 임베딩 벡터들과 결합하여 상태 벡터를 얻는다. 만약, 노드 a의 형태소가 "고향" 이면 tag(a), mtag(a)는 각각 "NNG", "고향/NNG"를 반환한다. mtag(a)가 수행 되는 노드는 3.1절에서 설명한 바와 같이 새로운 형태소가 스택에 PUSH되어 이전 형태소의 경계가 결정되는 시점이므로 S_1와 같이 스택의 2번째 TOP 형태소와 태그에 대해 적용된다.

추가로 mtag(a)에 의해 얻어져 적용되는 Lookup Table에 대해서는 [형태소-품사] 단위로 Glove 알고리즘을 적용하여 얻은 약 70만개의 형태소에 대한 100차원의 사전 학습한(pretrained) 임베딩 벡터를 사용하였다. 위에서 정의한 상태벡터 $\mathbf{r}(B)$, $\mathbf{r}(S)$을 연결하여 품사 태거 상태 표상 $\boldsymbol{p}_t$를 얻는다.

$$\boldsymbol{p}_t = [\mathbf{r}(B); \mathbf{r}(S)] \quad (5)$$

식 (5)을 통해 얻어지게 되는 품사 태거 상태 표상 집합을 $\boldsymbol{p} = \{\boldsymbol{p}_1, \cdots, \boldsymbol{p}_t\}$로 표현하며 식 (6)의 LSTM 신경망의 입력으로 하여 LSTM을 통해 인코딩 된 후 linear classifier를 사용하여 다음 전이액션으로의 점수 $\boldsymbol{score}_t$를 계산한다. 식 (6)의 $LSTM_t$함수는 LSTM을 통해 얻어진 은닉열로부터 t번째 은닉 벡터를 취하도록 한다.

$$\boldsymbol{score}_t = \boldsymbol{W} \cdot (LSTM_t(\{\boldsymbol{p}_1, \cdots, \boldsymbol{p}_t\})) + \boldsymbol{b} \quad (6)$$

얻어진 $\boldsymbol{score}_t$는 출력 층으로 연결되어 softmax를 통해 전

[1] 엄밀하게 정의하면 각 노드는 음절을 포함한 튜플의 형태로 존재하여 튜플의 음절을 얻어내는 함수인 char를 이용하여 $\boldsymbol{h}_{cpos(char(B_0))}$가 정확한 수식이지만 편의상 위의 형태를 사용한다. 식 (3),(4) 동일.

이 확률 중에 최대가 액션을 다음 전이 액션으로 하여 형태소의 분할 및 품사를 결정한다. 제안 모델의 장점은 이전 전이 액션으로 결정된 스택 내의 형태소와 해당 품사의 정보를 다음 전이 액션을 결정하기 위한 자질로 사용할 수 있어 추가적인 성능향상을 바라볼 수 있다.

4. 실험

4.1. 실험 셋팅

본 논문에서 제안한 모델을 평가하기 위해 [2]와 동일한 집합인 세종 품사 부착 말뭉치 약 25만 문장 중 75%를 학습 셋, 5%를 검증 셋 그리고 나머지 20%를 평가 셋으로 하여 본 모델을 학습하였고 모델의 학습률은 0.1로 설정하였고 모든 히든 레이어의 Dropout 비율은 0.8로 설정하였다.

4.2. 실험 결과

성능 비교를 위해 본 모델에 대한 비교 베이스 라인 모델으로는 순차 레이블링 문제에서 높은 성능을 보여주는 CRF 모델과 Bi-LSTM CRF 모델을 사용하였다. CRF에 대해서는 2가지 표기법을 사용하였다. 하나는 [B,I] 표기법이고 추가적인 표기법은 [B,I,E,S]표기법으로 각각 CRF, CRF(BIES)로 구분하며 [B,I,E,S] 표기법에 대한 설명은 다음과 같다.

- S: 형태소가 단일 음절일 경우 품사 태그
- B: 형태소의 시작 음절의 품사 태그
- E: 형태소의 마지막 음절의 품사 태그
- I: 형태소의 시작과 끝을 제외한 나머지 음절의 품사 태그

표 4. 모델 별 형태소 분석 성능

	F1(morph)
CRF * [3]	97.61%
CRF(BIES) *	97.75%
Bi-LSTM CRF *	96.96%
SVM [4]	98.03%
Seq2Seq [6]	97.15%
Copying Mechanism [7]	97.08%
전이기반 *	**97.77%**

(*는 평가셋이 동일)

표 4는 모델 별 형태소 분석 성능을 F1-measure로 보여 주고 있다. 동일 평가 셋에서 평가한 베이스 라인 모델인 CRF, Bi-LSTM CRF에 비해 제안한 전이 기반 형태소 분석 모델의 성능이 추가적인 성능 향상을 가져왔음을 확인 할 수 있다.

5. 결론

본 논문에서는 전이 기반 방식을 한국어 형태소 분석 문제에 알맞게 액션을 정의한 후 제안 모델 적용하여 기존의 방식에 비해 추가적인 성능 향상을 가져왔다. 차후 전이 기반 방식을 한국어 품사 태깅 및 의존 파싱 통합 모델에 대한 연구로 확장할 예정이다. 이에 나아가 전이 기반 방식을 개체명 인식 및 의미역 결정 문제에도 적용할 예정이다.

감사의 글

본 연구는 미래창조과학부 및 정보통신기술진흥센터의 정보통신·방송 연구개발사업의 일환으로 수행하였음. [R7119-16-1001, 지식증강형 실시간 동시통역 원천기술 개발]

참고문헌

[1] 이충희, 임준호, 임수종, 김현기. "기분석사전과 기계학습 방법을 결합한 음절 단위 한국어 품사 태깅." 정보과학회논문지 43.3 (2016): 362-369.

[2] 나승훈, 양성일, 김창현, 권오욱, 김영길. "CRF 에 기반한 한국어 형태소 분할 및 품사 태깅." HCLT 2012.

[3] Seung-Hoon Na . Conditional Random Fields for Korean Morpheme Segmentation and POS Tagging. ACM Transactions on Asian and Low-Resource Language Information Processing , 14(3), 2015

[4] 이창기. "Structural SVM 을 이용한 한국어 띄어쓰기 및 품사 태깅 결합 모델." 정보과학회논문지: 소프트웨어 및 응용 40.12 (2013): 826-832.

[5] 김혜민, 윤정민, 안재현, 배경만, 고영중. 품사 분포와 Bidirectional LSTM-CRFs를 이용한 음절 단위 형태소 분석기, HCL 2016

[6] 이건일, 이의현, 이종혁. "Sequence-to-sequence 모델을 이용한 한국어 형태소 분석 및 품사 태깅." 한국정보과학회 학술발표논문집 (2016)

[7] 황현선, 이창기. "Copying mechanism 을 이용한 Sequence-to-Sequence 모델기반 한국어 형태소 분석." 한국정보과학회 학술발표논문집 (2016): 443-445.

[8] Dyer Chris, M. Ballesteros, W. Ling, A. Matthews, N. A. Smith, "Transition-based dependency parsing with stack long short-term memory." arXiv preprint arXiv:1505.08075 (2015).

[9] Zhang, Meishan, Yue Zhang, and Guohong Fu. "Transition-Based Neural Word Segmentation." ACL (1). 2016.

개체명 사전 기반의 반자동 말뭉치 구축 도구

노경목[o†], 김창현[‡], 천민아[†], 박호민[†], 윤호[†], 김재균[†], 김재훈[†]
한국해양대학교[†], 한국전자통신연구원[‡]
kmq7542@gmail.com[o†], chkim@etri.re.kr[‡], minah0218@kmou.ac.kr[†],
homin2006@hanmail.net[†], 4168615@naver.com[†], jgk20000@naver.com[†], jhoon@kmou.ac.kr[†]

A Semi-automatic Annotation Tool based on Named Entity Dictionary

Kyung-Mok Noh[o†], Chang-Hyun Kim[‡], Min-Ah Cheon[†],
Ho-Min Park[†], Ho Yoon[†], Jae-Kyun Kim[†], Jae-Hoon Kim[†]
Korea Maritime and Ocean University[†], Electronics and Telecommunications Research Institute[‡]

요 약

개체명은 인명, 지명, 조직명 등 문서 내에서 중요한 의미를 가지므로 질의응답, 요약, 기계번역 분야에서 유용하게 사용되고 있다. 개체명 인식은 문서에서 개체명에 해당하는 단어를 찾아 개체명 범주를 부착하는 작업을 말한다. 개체명 인식 연구에는 개체명 범주가 부착된 개체명 말뭉치를 사용한다. 개체명의 범주는 연구 분야에 따라 다양하게 정의되므로 연구 분야에 적합한 개체명 말뭉치가 필요하다. 하지만 이런 말뭉치를 구축하는 일은 시간과 인력이 많이 필요하다. 따라서 본 논문에서는 개체명 사전 기반의 반자동 말뭉치 구축 도구를 제안한다. 제안하는 도구는 크게 전처리, 사용자 태깅, 후처리 단계로 나뉜다. 전처리 단계는 자동으로 개체명을 찾는 단계이다. 약 11만 개의 개체명을 기반으로 하여 트라이(trie) 구조의 개체명 사전을 구축한 후 사전을 이용하여 개체명을 자동으로 찾는다. 사용자 태깅 단계는 사용자가 수동으로 개체명을 태깅하는 단계이다. 전처리 단계에서 찾은 개체명 중 오류가 있는 개체명들은 수정하거나 삭제하고, 찾지 못한 개체명들은 사용자가 추가로 태깅하는 단계이다. 후처리 단계는 태깅한 결과로부터 사전 정보를 갱신하는 단계이다. 제안한 말뭉치 구축 도구를 이용하여 752개의 뉴스 기사에 대해 개체명을 태깅한 결과 7,620개의 개체명이 사전에 추가되었다. 제안한 도구를 사용한 결과 사용하지 않았을 때 비해 약 57.6% 정도 태깅 횟수가 감소했다.

주제어: 개체명 사전, 말뭉치 구축, 주석 도구

1. 서론

개체명 인식은 정보추출의 한 부분으로 문서 내에서 중요한 의미를 지닌 개체명을 찾아 그 범주를 결정하는 작업이다[1-3]. 개체명은 인명, 지명, 조직명 등으로 분류할 수 있으며 질의응답, 요약, 기계번역 분야에서 핵심어로써 매우 유용하게 사용되고 있다[1-3]. 개체명은 대부분이 고유 명사이며 시간의 흐름에 따라 계속 생성되기 때문에 모든 개체명을 사전에 등록하는 것은 불가능한 일이다[2]. 또한, 같은 단어라도 문맥에 따라 개체명의 범주가 달라지는 모호성(ambiguity) 문제도 존재한다[2-3]. 따라서 개체명을 찾는 일과 그 범주를 결정하는 일은 쉽지 않다[2-3].

개체명 인식을 위해 기계학습과 사전 정보를 활용한 다양한 연구가 진행되었다[2,4-6]. [2]는 기계학습 기반 개체명 인식에서 중의성을 자질에 명확하게 표기하기 위한 사전 자질 생성에 대해 연구했다. [4]는 한국어 위키피디아로부터 개체명 사전을 자동으로 구축하는 연구를 진행했으며 [5]는 원시 말뭉치로부터 추출한 문맥 패턴 정보와 개체명 사전을 이용한 제목 개체명 인식에 대해 연구했다. [6]은 기계 학습 모델을 이용한 개체명 인식을 연구했다.

개체명은 문맥에 따라 범주가 달라지므로 개체명 범주를 명확히 하기 위해서는 문맥 정보를 고려할 필요가 있다. 즉, 문장에서 등장하는 자질을 기준으로 문장 단위로 개체명의 범주를 결정하는 것보다 문서 전체의 맥락을 파악하여 각 단어의 개체명 범주를 결정하는 것이 더 정확하다. 그러나 사람이 수작업으로 모든 문서의 개체명을 찾아 범주를 부착하는 것은 시간이 오래 걸리고 힘든 일이다. 따라서 본 논문에서는 개체명 사전을 활용하여 사용자가 개체명 범주 부착 말뭉치를 효율적으로 구축하기 위한 반자동 말뭉치 구축 도구를 제안한다.

본 논문의 구성은 다음과 같다. 2장에서 현재 사용되고 있는 말뭉치 구축 도구에 대해 살펴본다. 3장에서 제안한 말뭉치 구축 도구를 소개한다. 마지막으로 4장에서 결론 및 향후 과제를 언급하며 끝을 맺는다.

2. 관련 연구

[7]은 [8]에서 제안한 시스템의 성능이 말뭉치의 크기에 비례하는 것이라 가정하고, 이를 보완하기 위해 말뭉치 확장을 위한 반자동 의미 부착 도구를 개발하였다. JSON 형식을 따르는 말뭉치 파일을 사용하여 미리 자동 태깅한 구문 분석 정보와 의미역 정보를 화살표로 표현하고 사람이 이해하기 쉽도록 GUI 기반의 인터페이스 제공했다. 그 결과 도구를 사용하지 않았을 때보다 말뭉치

구축 속도를 약 80% 향상시켰다.

[9]은 자동화된 정보 추출 도구의 훈련 및 실험, 평가를 위한 수동 주석(annotation) 지원 도구인 COAT를 소개하고 있다. 둘 이상의 사용자에게 동일한 문서를 배정하여 각각 주석을 달게 한다. 각 사용자로부터 받은 결과물을 통합하여 최종 말뭉치를 완성하는데, 교차 검증을 통해 주석 말뭉치의 신뢰성을 높이기 위한 방법을 사용했다. [9]의 도구도 사용자의 작업 효율을 위해 GUI 기반의 인터페이스와 단축키를 지원하여 말뭉치 구축 속도를 향상시켰다.

해외의 문서 말뭉치 구축 지원 도구는 GATE[10]와 brat[11] 등이 있다. GATE는 다양한 문서 분석 및 처리에 유용한 JAVA 기반 오픈 소프트웨어이다. GATE는 전체 코어 시스템을 재사용할 수 있는 JAVA 구성 요소로 분할 가능하다. 분할된 JAVA 구성 요소는 임베디드 시스템 등에서 재사용이 가능해서 확장성이 높다. GATE의 핵심 기능은 구문 분석, 형태소 분석, 정보 검색 도구, 주석 등이 있으며 이 외에도 텍스트를 처리하기 위한 다양한 구성 요소가 포함되어 있다. brat은 웹 기반의 문서 주석 도구이다. 정해진 형식의 구조화된 주석을 위해 설계되었으며, 주석, 정보 추출, 의존 구문, 정규화, 청킹(chunking) 등 다양한 기능을 지원한다. 시각화가 잘 되어 있어서 단어와 단어의 관계를 잘 보여주기 때문에 의존 관계를 정의할 때 유용하며 드래그 앤 드롭을 지원해서 편리한 UI를 제공한다.

3. 시스템 소개

본 논문에서 제안하는 반자동 말뭉치 구축 도구 시스템은 크게 전처리, 사용자 개체명 태깅, 후처리 단계로 나뉘며 구성도는 그림 1과 같다. 그림 1에서 실선 화살표는 작업 순서를 의미하고 점선 화살표는 개체명 사전과 관련된 정보의 흐름을 나타낸다.

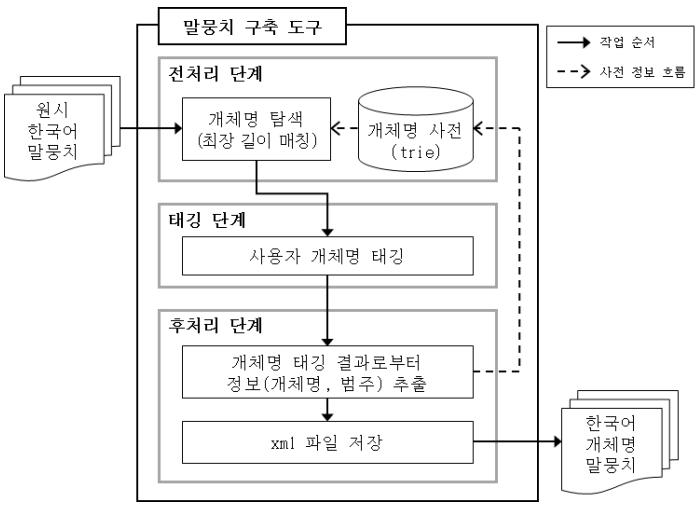


그림 1. 반자동 말뭉치 구축 도구 시스템 구성도

3.1 전처리 단계(자동)

전처리 단계에서는 개체명 사전을 이용하여 원시 말뭉치로부터 개체명을 자동으로 찾는 단계이다. 개체명 사전은 트라이(trie) 구조로 되어 있으며 개체명과 개체명의 범주, 개체명의 빈도수의 정보를 담고 있다. 사전에는 2음절 이상의 개체명만 등록되어 있다. 이러한 사전을 이용해서 문서의 시작부터 순차적으로 개체명을 탐색한다. 예를 들어, 개체명 사전에 “서울시”와 “서울시청”의 개체명이 존재할 때, “서울시는 서울시청에”란 문장의 탐색 과정은 그림 2와 같다.

그림 2. 예시 문장의 개체명 탐색 과정

3.2 사용자 개체명 태깅 단계(수동)

사용자 개체명 태깅 단계는 사용자가 개체명을 수동으로 태깅하는 단계이다. 사용자는 전처리 단계에서 잘못 부착된 개체명의 범주를 수정하거나 삭제하고 새로운 개체명을 발견하면 적절한 범주를 부착한다.

태깅 관련 기능은 세 가지가 있다. 첫 번째는 사전에 정보가 없는 개체명(unknown named entity)의 범주를 추가하는 것이고, 두 번째는 부착된 개체명의 범주를 다른 범주로 변경하는 것이다. 마지막은 부착된 개체명의 범주를 삭제하는 것이다.

3.2.1 개체명 범주 추가 기능

미등록 개체명에 범주를 추가하는 방법은 마우스나 키보드를 이용하여 개체명을 선택(드래그)하여 범주를 부착한다. 이때, 선택한 개체명과 같은 개체명이 문서 내에 존재하면 해당하는(선택한 개체명이 아닌) 모든 개체명에 같은 범주를 추가할 것인지를 “예”, “아니오” 형식으로 묻는다. “예”를 선택하면 아래의 표 1과 같

은 규칙이 적용되고 “아니오”를 선택하면 선택한 개체명만 범주를 추가한다.

표 1. 문서 내의 같은 개체명을 태깅하는 규칙

규칙	설 명
1	해당하는 개체명에 범주가 부착되어 있지 않으면 선택한 개체명의 범주를 부착한다.
2	해당하는 개체명이 어떠한 개체명의 일부분인 경우에는 태깅하지 않는다.
3	해당하는 개체명에 어떠한 개체명의 범주가 부착되어 있을 때, 선택한 개체명이 해당 개체명을 포함하거나 같다면 기존의 개체명 범주를 삭제하고 선택한 개체명의 범주를 다시 부착한다.

해당 규칙의 예시는 아래의 그림 3과 같다.

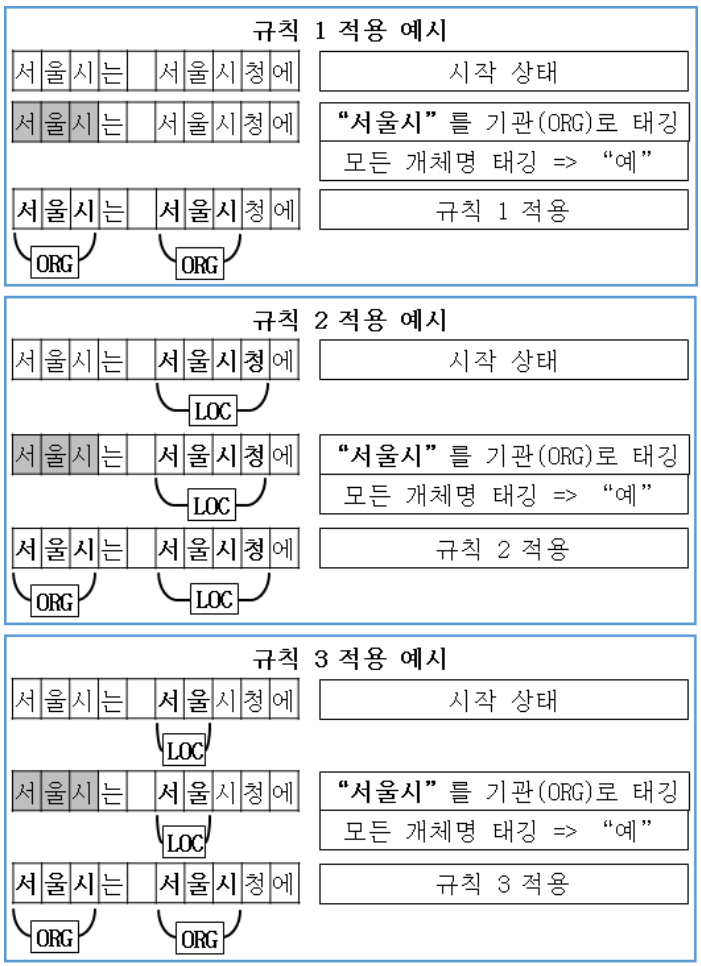


그림 3. 같은 개체명을 태깅하는 규칙 예시

3.2.2 개체명 범주 변경 기능

개체명의 범주를 변경하는 방법은 마우스로 해당 개체명을 좌클릭 또는 우클릭해서 다른 범주로 변경할 수 있다.

3.2.3 개체명 범주 삭제 기능

개체명의 범주를 삭제하는 기능은 두 가지가 있다. 첫째는 선택한 개체명의 범주만 삭제하는 것이고, 둘째는 선택한 개체명과 같은 모든 개체명의 범주를 삭제하는 것이다.

3.2.4 기타 편의 기능

그 외에 사용자 편의성을 위해 단어를 찾는 기능, 글씨의 크기를 조절하는 기능, 줄 간격을 조절하는 기능, 문서를 삭제하는 기능 등을 구현하였다.

3.3 후처리 단계

후처리 단계에서는 이때까지 진행했던 문서의 정보를 xml 파일 형식으로 저장하고 사전을 갱신하는 단계이다. 태깅을 완료한 문서로부터 2음절 이상인 개체명 정보(개체명, 범주, 빈도수)를 추출하여 개체명 사전 정보를 갱신한다.

3.4 실행 화면

그림 4는 말뭉치 구축 도구의 실행 화면 중 하나이다. 그림 4의 좌측 상단에 있는 원시 말뭉치 파일창에서 원시 말뭉치를 선택하면 전처리 단계를 거쳐 그림 4의 중앙 화면에 보이게 된다. 전처리 단계에서 찾은 개체명들은 음영처리 되어서 사용자에게 보이며 별도의 메모 없이 색상으로 구분된다. 사용자의 편의를 위해 키보드 단축키와 마우스를 이용한 태깅이 가능하다. 사용자가 개체명 태깅을 완료하면 개체명 범주 말뭉치 생성이 끝난다. 완료된 개체명 말뭉치는 그림 4의 좌측 하단에 있는 개체명 말뭉치 파일 창에 보이게 되며, 태깅한 결과를 재확인하거나 수정할 수 있다.

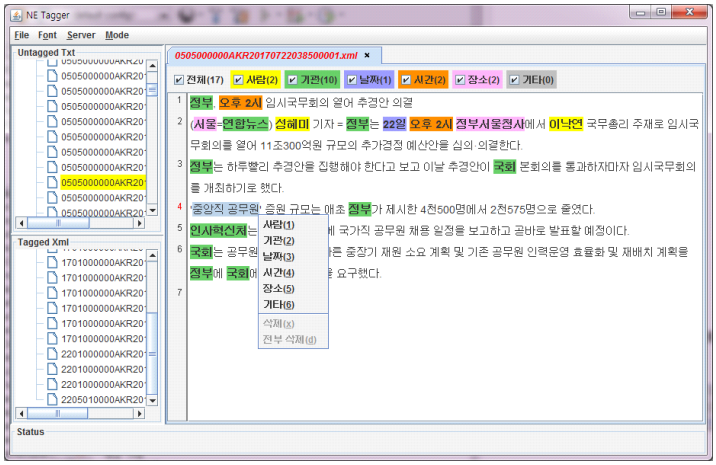


그림 4. 반자동 말뭉치 구축 도구 실행 화면

4. 결론 및 향후 과제

본 논문에서는 기존에 구현된 다양한 말뭉치 구축 도구 시스템과 개체명 인식 관련 논문을 참고하여 개체명 사전 기반의 반자동 말뭉치 구축 도구를 개발했다. 약 11만 개의 개체명이 포함된 사전 정보를 활용하여 개체명 말뭉치를 구축하였다. 원시 말뭉치는 2017년 5월~7월 기간의 뉴스 기사를 모았으며, 752개의 기사에서 개체명을 찾아 태깅하였다. 개체명의 범주는 사람, 기관, 날짜, 시간, 장소, 기타 이렇게 6가지를 선정했으며, 752개의 기사로부터 총 25,169개의 개체명을 태깅하였다. 중복을 제거한 개체명의 수는 7,620개였고 이 중 2음절 이상의 개체명은 7,546개였다. 25,169개의 개체명 중 기관이 8,365개로 가장 많았으며, 사람 5,929개, 장소 4,715개, 날짜 3,447개, 기타 2,484개, 시간 229개 순이었다. 기타는 행사명, 전쟁 명, 사건명, 프로그램명, 책 이름, 영화 제목, 노래 제목, 문화재 이름 등을 기타로 태깅했다.

개체명 말뭉치를 구축할수록 후처리 단계를 거쳐 개체명 사전이 확장되었다. 사전이 확장됨에 따라 전처리 단계에서 자동으로 태깅되는 개체명의 수가 늘어남으로써 사용자의 태깅 횟수가 감소하였다. 하지만 중의적인 표현을 가진 개체명으로 인해 사용자 태깅 단계에서 문맥에 맞지 않는 개체명을 변경, 삭제해야 하는 일이 생겼다. 예를 들어, "고려"의 경우 나라의 이름이기도 하지만 "고려하였다"와 같이 동사로 쓰이기도 하므로 오류가 발생했다. 평균적으로 한 기사에 약 33개의 개체명이 존재했으며 사용자 태깅 단계에서 사용자가 개체명 범주를 추가, 변경, 삭제해야 하는 횟수는 약 14회 정도였다.

개체명 사전을 이용하지 않고 개체명 말뭉치를 구축했을 때와 비교하면 전체적인 작업량은 약 57.6% 정도 줄었다. 하지만 불필요한 작업량이 발생하였으며, 이는 사전이 확장됨에 따라 더 늘어날 것으로 보인다. 향후 과제로는 전처리 단계에서 사용되는 개체명 사전을 기계학습 모델로 대체하여 개체명 인식의 정밀도를 높이는 방안 등 효율적으로 말뭉치를 구축할 방법을 연구할 계획이다.

감사의 글

이 논문은 2017년도 정부(과학기술정보통신부)의 재원으로 정보통신기술진흥센터의 지원을 받아 수행된 연구임(R7119-16-1001, 지식증강형 실시간 동시통역 원천기술 개발)

참고문헌

[1] David Nadeau and Stasoshi Sekine, "A survey of named entity recognition and classification", Journal of Linguisticae Investingations, vol. 30, no. 1, pp.3-26, 2007.

[2] 김재훈, 김형철, 최윤수, "기계학습 기반 개체명 인식을 위한 사전 자질 생성", 정보관리연구, 제41권, 제2호, pp.31-46, 2010.

[3] 이경희, "한국어 문서에서 개체명 인식에 관한 연구", 제12회 한글 및 한국어 정보처리 학술대회 논문집, pp.292-299, 2000.

[4] 배상준, "한국어 위키피디아를 이용한 분류체계 생성과 개체명 사전 자동 구축", 정보과학회논문지 : 컴퓨팅의 실제 및 레터, 제16권, 제4호, pp. 492-496, 2010.

[5] 이주영, "자동 구축된 문맥 패턴과 개체명 사전에 기반한 제목 개체명 인식", 제16회 한글 및 한국어 정보처리 학술대회 발표자료집, 제16권, 제1호, pp.40-45, 2004.

[6] 이창기, "Structural SVMs 및 Pegasos 알고리즘을 이용한 한국어 개체명 인식", 인지과학, 제21권, 제4호, pp.655-667, 2010.

[7] 배장성, "한국어 의미역 말뭉치 구축을 위한 반자동 태깅 도구 개발", 한국정보과학회 2014 한국컴퓨터종합학술대회 논문집, pp. 592-594, 2014.

[8] 이창기, "Structural SVM 기반의 한국어 의미역 결정", 한국정보과학회 2014 한국컴퓨터종합학술대회 논문집, pp. 574-576, 2014.

[9] 최동현, "COAT: 시멘틱 어노테이션 말뭉치 구축 지원 도구", 제23회 한글 및 한국어 정보처리 학술대회 논문집, pp.85-89, 2011.

[10] gate [Online] https://gate.ac.uk/
[11] bart [Online] http://brat.nlplab.org/

● 경진대회

Bidirectional Dynamic LSTM 을 이용한 음절 단위 개체명 추출 및 자동화된 말뭉치 구축

오성식○, 임창대, 안기호, 박외진
(주)아크릴
{Harvey, Dann, Kevin, Jin}@iacryl.com

Syllables-based Named Entity Extraction and Automatic Corpus Construction using Bidirectional Dynamic LSTM

Sungsik Oh○, Changdae Lim, Keeho Ahn, Weijin Park
Acryl Inc.

요 약

개체명 인식은 자연어 문장에서 장소, 제작물, 사람 등 분류를 통한 의미 부여가 가능한 단어를 파악하는 기술로서 의미 분석을 위한 핵심 기술이다. 현재 많은 개체명 분석 관련 연구들은 형태소 분석 결과에 의존적인 형태를 갖고 있어서, 형태소 분석 결과의 정확성이 개체명 분석 결과의 성능에 영향을 미치고 있다. 본 연구에서는 형태소 분석 과정을 거치지 않는 음절 기반의 개체명 분석 기술을 제안하여 형태소 분석의 정확도가 낮은 통신어, 신조어 분석 성능을 향상하였다. 또한, 자동화된 방법으로 음절 단위 개체명 말뭉치 및 개체명 사전을 구축하는 프로세스를 정의하여 개체명 분석의 정확도 향상 및 인지 범주의 확대를 도모하였다. 본 연구에서 제안한 개체명 인식 기술은 한국어 개체명 표준에 기반한 129가지의 개체명 분류가 가능하며, 이는 자연어 처리 기술이 필요한 산업계에서 상용화하는데 큰 기여를 할 것으로 판단된다.

주제어: 음절 단위 개체명 분석, Bidirectional dynamic LSTM, 개체명 인식 말뭉치 구축, 개체명 사전 자동생성

1. 서론

개체명 인식(Named Entity Recognition)은 자연어 문장에서 장소, 제작물, 사람 등 분류가 가능한 의미를 단어에 부여하는 기술로서 자연어 처리에서 의미 분석의 기반이 되는 기술이다.

개체명 인식은 자연어 처리의 중요 분야 중 하나로서 시계열 데이터 처리에 용이한 RNN(Recurrent Neural Network)을 이용한 순차 데이터 처리가 주로 행해진다. 최근에는 RNN의 Memory문제를 해결한 LSTM(Long Short-Term Memory)을 이용한 자연어 처리가 널리 연구되고 있으며, RNN의 양방향 순차 데이터 학습 방식인 BRNN(Bidirectional Recurrent Neural Networks)을 이용하여 순차 데이터 학습의 정확성을 향상 한다[1,2]. RNN 구현 시 입력 값의 고정된 길이를 맞추기 위하여 실제 입력 값과 맞지 않는 부분을 빈(NULL)값으로 채워 넣는데, 이는 학습 데이터의 일관성을 낮추고 정확도를 저하시키는 원인이 된다. 이런 문제를 해결하기 위하여 DRNN(Dynamic Recurrent Neural Networks)를 이용하여 가변적인 길이의 입력을 가능하게 한다[3].

기존 개체명 인식 연구 사례는 형태소 분석 후 분석된 형태소 중 개체명으로 인식될 수 있는 것을 추출하여 개체명을 분류 하였다[4]. 이러한 방법은 분석될 개체명을 추출하여 분석하므로 분석할 정보량을 비약적으로 줄이는 경제적인 방법이나 형태소 분석기의 정확도에 따라 개체명 추출이 불가능한 경우가 있다. 특히 통신어, 신조어 분석에서 낮은 정확도를 보인다. 이런 문제를 해결하기 위해 본 논문에서는 형태소 분석기에 의존하지 않는 개체명 분석기를 개발하고 음절 단위 개체명 말뭉치와 동시에 개체명 사전을 구축하는 시스템을 개발 하였다.

본 논문에서는 한국어 개체명 인식에 BRNN, LSTM, DRNN을 병합하여 Bidirectional Dynamic LSTM을 이용한 음절 단위로 개체명을 추출하는 방식을 제안한다. 현재 개체명 분석 연구는 사람, 기관, 시간 등 주요한 10가지 이하의 개체명 분류를 대상으로 하고 있는 경우가 많다. 그러나, 실제 의미 분석을 위해서는 모든 개체명 분류를 하는 것이 필수적이며, 이를 위해 본 논문에서는 실제 개체명 분류를 모든 텍스트에서 사용하기 위해 한국어 개체명 표준(개체명 태그 세트 및 태깅 말뭉치(Tag Set and Tagged Corpus for Named Entity Recognition, TTAK.KO-10.0852))을 이용하여 개체명 분류의 분석을 수행하였다.

본 논문의 구성은 다음과 같다. 2장 에서는 관련 연구에 대해 설명한다. 3장에서는 Bidirectional Dynamic LSTM을 이용한 음절 단위 개체명 추출 방법에 대하여 설명한다. 4장에서는 실험을 통해 제안 방법에 대하여 평가한다. 5장에서는 실험 내용 분석과 차기 연구 방향에 대하여 서술한다.

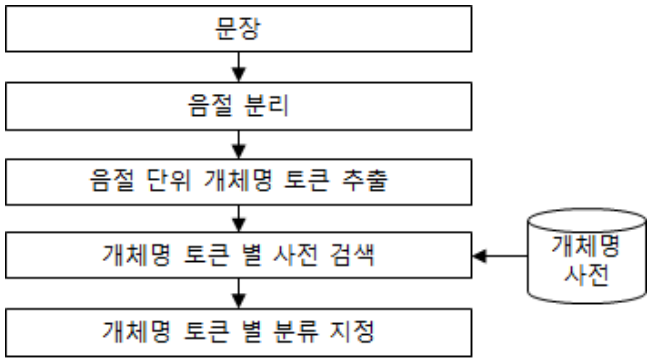
2. 관련 연구

현재까지 다양한 개체명 분석과 음절 단위 자연어 처리에 대한 연구들이 진행되었다[5-7]. [5]는 음절 단위 형태소 분석을 Bi-LSTM-CRF를 이용하여 분석하였다. 또한, [6]은 LSTM을 이용하여 개체명 분석을 시도하였으며, [7]은 Bi-LSTM-CRF를 이용하여 단어 임베딩 벡터, 품사 임베딩 벡터, 음절 입력을 이용하는 방법을 제안하였다.

3. 제안 방법

3.1 분석 시스템 구성

음절 기반의 개체명 태깅 시스템은 (그림 1)과 같은 전처리 과정과 인공 신경망의 예측 결과를 거쳐서 음절별 개체명 토큰 포함 여부를 결정하게 된다.

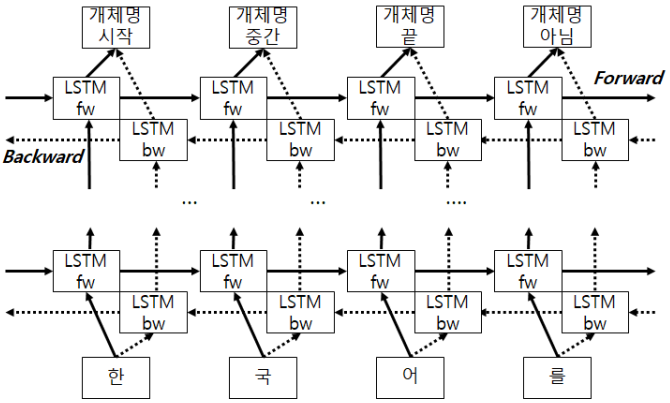


(그림 1). 음절 기반의 개체명 태깅 시스템 구성도

입력 문장을 음절 단위로 분리하여 사전에 구성되어 있는 음절 별 참조 번호로 변환 한다. 따라서, 입력 문장은 참조 번호 수열로 변환되며 이 수열은 Bidirectional Dynamic LSTM과 같은 기계학습 기반 분류기에 입력된다. 그리고 음절 별로 개체명 토큰 포함 여부를 분류 한다. 토큰으로 판명된 음절 토큰은 개체명 사전을 참조하여 개체명 토큰 별로 분류를 지정한다.

3.2 Bidirectional Dynamic LSTM을 이용한 개체명 추출 방식

입력 음절을 참조 번호로 변환하여 Bidirectional Dynamic LSTM에 입력한다. 본 과정에서, 입력 음절의 distributed representation 의 성질을 이용하기 위하여 벡터로 변환하는 음절 임베딩 알고리즘을 이용하지 않았는데, 이는 Bidirectional RNN의 각 LSTM Cell의 stack을 복수로 쌓으면 distributed representation이 자동으로 반영된다고 판단한 것에 기인한다. 개체명 추출을 위한 Bidirectional Dynamic LSTM에서는 5개의 Cell Stack을 이용하였다. 각 입력 음절에 대하여 개체명 토큰 포함 여부를 예측 하게 하였는데 '㉠ 개체명 시작', '㉡ 개체명 중간', '㉢ 개체명 끝' 으로 음절 별 개체명 토큰 포함 여부를 예측하게 하였고 '㉣ 개체명 아님' 으로 개체명에 속하지 않는 음절을 분리해 내었다. 개체명으로 판명된 음절 토큰은 개체명 사전을 참조하여 개체명 토큰 별로 개체명 분류를 지정한다. (그림 2)는 음절 입력 방식의 개체명 추출을 위한 Bidirectional Dynamic LSTM을 도식화 한 그림이다.



(그림 2). 음절 입력 방식의 Bi-Dynamic-LSTM 예

3.3 학습 말뭉치 및 개체명 사전 구축

3.3.1 학습 말뭉치 구축

음절 별 개체명 학습 말뭉치를 구축하기 위해 실제로 웹에서 사용되고 있는 문장을 수집하여 개체명 태깅을 진행하였다. 말뭉치 구축에는 30명의 사용자들이 약 17만 개의 문장에 대하여 문장에서 개체명에 해당하는 음절을 추출하고 개체명 분류 정보를 태깅 하였다. 개체명 분류는 정보통신단체표준인 '개체명 태그 세트 및 태깅 말뭉치(Tag Set and Tagged Corpus for Named Entity Recognition, TTAK.KO-10.0852)' 를 가공하여 18개의 대분류, 129개의 소분류로 개체명 분류 말뭉치를 구축 하였다[9].

3.3.2 개체명 사전 구축

말뭉치를 구축하면서 사용자들이 표기한 개체명은 개체명 사전으로 저장 된다. 구축된 개체명 사전은 개체명 추출 알고리즘의 결과에 참조되어 추출된 개체명의 분류를 지정해 준다. (그림 3)은 음절 별 개체명 설정과 분류 사전 구축 예이다.

(그림 3). 음절 별 개체명 말뭉치, 개체명 사전 구축 예

사용자들이 웹 수집 말뭉치를 기반으로 구축한 개체명 사전과 동시에 위키피디아의 표제어에 대해서도 개체명 사전을 함께 구축하였다. DBPedia ontology를 현재 사용중인 개체명 분류로 변환 하여 개체명 사전을 확장하였다[10]. 이를 통하여 45,000건의 사용자 태깅 개체명 사전과 위키피디아 표제어를 이용한 750,000건의 개체명 확장 사전을 구축했다.

4. 실험 및 성능 평가

Bidirectional Dynamic LSTM을 구현하기 위하여 Tensorflow 1.2.1을 사용하였으며 개체명 말뭉치 구축을 위한 클라이언트에서 Komoran3.0을 사용하였다[11]. 개체명 사전의 데이터베이스로 ElasticSearch 5.1.1를 사용하였다.

4.1 데이터 셋

음절 단위 개체명 태깅을 시행하기 위한 데이터로서 웹 수집 텍스트를 활용하여 개체명 태깅, 학습, 평가를 시행하였다. 음절 단위 개체명 분석기의 개발 취지에 맞게 형태소 분석이 용이하지 않은 통신어, 문법 파괴어의 학습과 평가를 하였다. 태깅된 말뭉치는 약 17만 문장이며 16만 문장을 가지고 학습하였고 평가는 4,000문장에 대하여 시행하였다.

4.2 성능 분석

성능 분석은 첫째로 각 입력 음절 분류인 '㉠ 개체명 시작', '㉡ 개체명 중간', '㉢ 개체명 끝', '㉣ 개체명 아님' 의 4종의 분류 정확도 측정과 둘째로 추출된 개체명의 분류가 말뭉치와 얼마나 일치하는지를 보는 크게 2가지 평가 방식으로 나뉜다.

표 1. 음절 단위 개체명 분류 예

⋮
한 - ㉠ 개체명 시작
국 - ㉡ 개체명 중간
어 - ㉢ 개체명 끝
를 - ㉣ 개체명 아님
⋮

개체명으로 시작과 끝으로 인식 된 문자열을 개체명 사전에서 검색하여 개체명 분류 태그를 부착한다.

4.2.1 평가 결과

평가 방식에 따라 세 가지 평가 척도를 상정 하여 평가하였다.

1) 개체명 음절 추출 정확도

$$= \frac{\text{분류가 맞은 음절수}(B)}{\text{전체 문장 음절수}(A)}$$

$$= \frac{\sum_{j=1}^{\text{문장별 음절수}} \text{일치음절수}(\text{말뭉치분류}_j, \text{예측치분류}_j)}{\sum_{i=1}^{4} \text{음절수}(\text{말뭉치}, \text{음절분류}_i)}$$

$$\left(\text{일치 음절수}(a,b) = \begin{cases} 1 & (a=b) \\ 0 & (a \neq b) \end{cases}\right)$$

2) 개체명 음절 인식 정확도

$$= \frac{\text{개체명 해당 음절수 중 분류가 맞은 음절수}(B)}{\text{개체명에 해당되는 음절수}(A)}$$

$$= \frac{\sum_{j=1}^{\text{문장별 음절수}} \text{일치음절수}(\text{말뭉치분류}_j, \text{예측치분류}_j)}{\sum_{i=1}^{3} \text{음절수}(\text{말뭉치}, \text{음절분류})}$$

(음절분류가 ㉣인 경우는 일치음절수에서 계산하지 않음.)

3) 개체명 분류 인식 정확도

$$= \frac{\text{개체명 분류가 맞은 토큰수}(B)}{\text{문장별 개체명수}(A)}$$

평가 방식 1), 2), 3) 에 따라 평가한 결과는 표 2와 같다.

표 2. 음절 단위 개체명 분류 결과

	정확도	(A)	(B)
1) 개체명 음절 추출	0.889	101523	90265
2) 개체명 음절 인식	0.847	25216	21367
3) 개체명 분류 인식	0.406	10929	4433

4.2.1 결과 분석

129종의 개체명 분류 대상에 대하여 '1) 개체명 음절 추출' 이 90%에 가까운 정확도를 보이고 '2) 개체명 음절 인식' 의 정확도가 80%를 상회 하는 것으로 나타났으며, 음절 단위의 개체명 추출을 Bidirectional Dynamic LSTM을 이용한다면 개체명 추출 및 인식에 높은 성능을 보이고 있음을 확인하였다. 그러나 개체명 분류 인식 성능은 40% 정도로 나타났는데, 이러한 성능의 원인은 다음과 같은 이유로 판단된다. 첫째, 추출된 개체명의 분류가 개체명 사전에 없는 경우이다. 이것은 지속적인 개체명 사전 구축을 통해 해결 될 수 있을 것으로 판단된다. 둘째로는, 개체명 말뭉치 구축을 30명의 사용자가 진행하였으므로 실제로 예측된 분류가 맞음에도 불구하고 말뭉치에 기재된 개체명 분류와 실험을 통해 예측한 분류가 달라서 오답으로 기재 된 경우에 기인한다. 4,000건의 테스트 데이터에 대하여 모두 개체명 분류의 정확성 판별을 직접 할 수 없었기 때문에 '3) 개체명 분류 인식' 의 정확도는 추후 지속적인 실험이 필요할 것으로 판단된다.

5. 결론

본 논문에서는 Bidirectional Dynamic LSTM 기반 한국어 개체명 분류를 위해 음절 단위 개체명 분류 방법 및 개체명 사전 구축, 개체명 말뭉치 구축 방법론을 제안하였다. 실험 결과, 제안 방법은 129가지의 개체명 추출에서 현재 10가지 이하의 개체명 분류 방식에서 도달한 수준과 같은 개체명 추출 정확도를 구현하였다[7]. 별도의 휴리스틱 알고리즘과 word2vec 없이 음절 단위 입력만으로도 개체명 추출에서 우수한 성능을 보임을 확인 할 수 있었다. 향후에는 개체명 사전을 참조하는 현재의 방식에서 현재 사용하는 RNN의 구조를 바꾸지 않고 state 벡터를 이용한 개체명 자동 분류 방식에 대한 연구가 고려되어야 할 것으로 판단된다.

감사의 글

본 연구는 미래창조과학부 및 정보통신기술진흥센터의 정보통신·방송 연구개발사업[R0114-16-0012, 개인화 서비스를 위한 통합 인지 컴퓨팅 플랫폼 개발]과 2017국어정보처리시스템경진대회(국립국어원) 일환으로 수행하였음.

참고문헌

[1] Sepp Hochreiter, Jurgen Schmidhuber, "LONG SHORT-TERM MEMORY", Neural Computation 9(8), pp.1735-1780, 1997.

[2] Schuster, Mike, and Kuldip K. Paliwal, "Bidirectional recurrent neural networks", IEEE TRANSACTIONS ON SIGNAL PROCESSING, VOL. 45, NO. 11, pp.2673-2681, 1997.

[3] Barak A. Pearlmutter, "Dynamic recurrent neural networks", Tech. Rep. CMU-CS-90-196, 1990.

[4] 이창기, 김준석, 김정희, 김현기, "딥 러닝을 이용한 개체명 인식", 한국정보과학회 제 41 회 정기총회 및 동계학술발표회, pp. 423-425, 2014.

[5] 김혜민, 윤정민, 안재현, 배경만, 고영중, "품사 분포와 Bidirectional LSTM CRFs를 이용한 음절 단위 형태소 분석기", 제28회 한글 및 한국어 정보처리 학술대회 논문집, pp.3-8, 2016.

[6] 이창기, "Long Short-Term Memory 기반의 Recurrent Neural Network를 이용한 개체명 인식", 한국정보과학회 2015 한국컴퓨터종합학술대회 논문집,pp.645-647, 2015.

[7] 유홍연, 고영중, "품사 임베딩과 음절 단위 개체명 분포 기반의 Bidirectional LSTM CRFs를 이용한 개체명 인식", 제28회 한글 및 한국어 정보처리 학술대회 논문집, pp.105-110, 2016.

[8] Tomas Mikolov, Kai Chen, Greg Corrado, Jeffrey Dean, "Efficient Estimation of Word Representations in Vector Space", arXiv:1301.3781, 2013.

[9] Acryl, https://goo.gl/o1pQrT, 2017.

[10] DBPedia, http://mappings.dbpedia.org/server/ontology/classes/, 2017.

[11] SHINEWARE, https://github.com/shin285/KOMORAN, 2017.

Bidirectional LSTM CRFs를 이용한 한국어 개체명 인식

송치윤, 양성민, 강상우

가천대학교 소프트웨어학과
{a2c222, ysm0622, swkang}@gachon.ac.kr

Named-entity Recognition Using Bidirectional LSTM CRFs

Chi-Yun Song, Sung-Min Yang, Sangwoo Kang
Department of Software Engineering, Gachon University

요 약

개체명 인식은 문서 내에서 고유한 의미를 갖는 인명, 기관명, 지명, 시간, 날짜 등을 추출하여 그 종류를 결정하는 것을 의미한다. Bidirectional LSTM CRFs 모델은 연속성을 갖는 데이터에 가장 적합한 RNN기반의 심층 학습모델로서 개체명 인식 연구에 가장 우수한 성능을 보여준다. 본 논문에서는 한국어 개체명 인식을 위하여 Bidirectional LSTM CRFs 모델을 사용하고, 입력 자질로 단어뿐만 아니라 품사 임베딩 모델과, 개체명 사전을 활용하여 입력 자질을 구성한다. 또한 입력 자질에 대한 벡터의 크기를 최적화 하여 기본 모델보다 성능이 향상되었음을 증명하였다.

주제어 : 개체명 인식, 개체명 사전, Bidirectional LSTM CRFs, 워드 임베딩

1. 서론

개체명(Named-Entity)이란 문장 내에서 인명, 기관명, 지명 등과 같은 고유한 의미가 있는 명사를 의미한다. 개체명 인식은 문장 내에서 개체명을 추출하여 개체명의 카테고리를 파악하는 것이다.

전통적인 개체명 인식 방법은 사전기반, 규칙기반 방법을 사용하였지만 최근 기계 학습 방법을 적용한 연구들이 많이 시도되고 있다. 지도 학습 방법으로는 HMM(Hidden Markov Model)[1], SVM(Support Vector Machine), CRFs(Conditional Random Fields)를 사용하는 방법들이 활발히 연구 및 제안되었고, 최근에는 RNN(Recurrent Neural Network), FFNN(Feed-Forward Neural Network), CNN(Convolutional Neural Network)[3] 등의 심층 신경망을 이용한 방법들이 기존의 지도 학습 방법보다 상대적으로 높은 성능을 보인다. 심층 신경망 모델 중 RNN 모델은 연속성을 갖는 데이터에서 우수한 성능을 보이지만 기울기 손실(vanishing gradient)이라는 문제점을 갖고 있지만 최근 개선된 모델인 LSTM(Long Short-term Memory Network) 모델은 memory cell과 3개의 gate를 통해 RNN의 기울기 손실 문제를 해결한다[2].

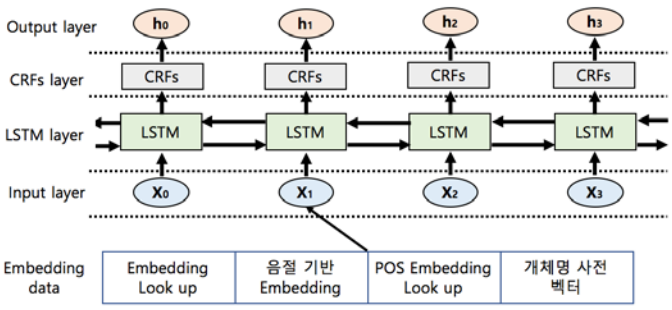
본 논문에서는 기존의 LSTM 모델을 기반으로 데이터를 양방향성(Bi-directional)으로 입력하고, 모델의 출력 값들 사이의 전이 확률을 포함시킴으로써 연속적인 데이터에 적합한 Bi-directional LSTM-CRFs 모델을 적용한다.

개체명 인식의 입력은 기본적으로 형태소 단위를 사용한다. 형태소 단위는 비지도 학습(unsupervised learning)을 통해 사전 학습된 단어 및 품사 임베딩 모델을 사용하며 추가적으로 단어 임베딩 벡터, 개체명 사전 자질 벡터를 통해 입력 단어의 표상을 확장한다.

본 논문의 구성은 다음 장에서 Bidirectional LSTM-CRFs 모델 및 단어 표상 확장 방법에 대해 소개한다. 3장에서는 본 논문에서 제시한 방법에 대한 실험 결과를 평가 및 분석하고, 마지막으로 결론을 기술한다.

2. 제안 방법

본 논문은 [그림 1]과 같이 임베딩 데이터가 Input layer통해 입력 되고 LSTM layer와 CRFs layer를 통해서 예측한 개체명(h)이 출력이 되는 계층 구조를 제안한다.



[그림 1] 제안 모델의 전체 구성도

본 연구는 과학기술정보통신부 및 정보통신기술진흥센터의 SW중심대학 지원사업의 연구결과로 수행되었음 (2015-0-00932)

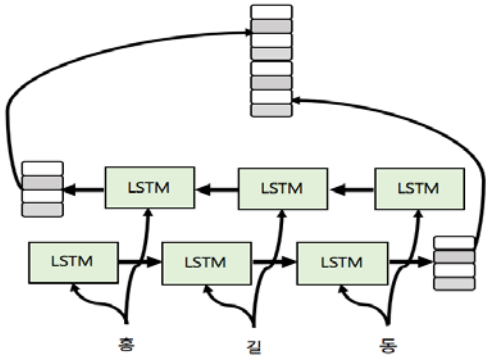
2.1 Bidirectional LSTM-CRFs를 이용한 학습 모델

Bidirectional LSTM-CRFs 모델은 LSTM 모델에서 문자열을 양방향으로 입력 받는다. 각 들은 은닉계층을 통과하고, CRFs를 통해 전이확률을 추가하여 결과 간의 의존성을 고려한다.

2.2 입력 자질

2.2.1 워드 임베딩

제안하는 모델에서는 입력 문장을 벡터로 변환하여 입력 자질로 사용한다. 워드 임베딩 모델을 구축하기 위하여 국립국어원에서 제공된 학습 말뭉치와 세종코퍼스, 위키피디아에서 수집한 말뭉치를 사용하였다. gensim 모델을 통해 워드 임베딩 모델을 사전 학습시켰다. 품사 정보를 반영하기 위해 단어와 품사 태그를 결합한 형태로 적용하였다. 또한, 미등록어 문제를 보완하기 위해서 음절 기반 워드 임베딩을 사용하였다. 이 방법은 미등록어가 입력되었을 때 말뭉치 내에서 유사한 단어들을 기반으로 벡터를 생성해줌으로써 미등록어 문제를 완화할 수 있다. 해당 음절 기반의 워드 임베딩은 [그림 2]과 같이 Bidirectional LSTM을 통해 변환된다.



[그림 2] 입력 벡터 생성 예시

2.2.2 품사 임베딩

품사는 개체명 인식에서 매우 중요한 자질이다. 따라서 품사 정보를 사용하여 입력데이터의 정보량을 확장하였다. 일반적으로 품사 자질을 추가하기 위해서는 원-핫 인코딩(one-hot encoding)을 통해 벡터로 변환하지만 품사들간의 연속적인 의미를 반영하기 위해서 품사 기반의 사전 학습된 벡터를 생성한다. 품사 임베딩 모델은 워드 임베딩 모델과 마찬가지로 국립국어원에서 제공된 학습 말뭉치와 gensim 모델을 사용하여 학습하였다.

2.2.3 개체명 사전

개체명 사전의 정보를 이용하여 입력 단어로부터 자질을 확장한다. 개체명 사전은 2016 ~ 2017년 국어 정보처리 시스템 경진대회에서 제공된 말뭉치와 세종코퍼스, 위키피디아의 데이터에서 추출한 개체명을 활용하였다. 추출된 개체명들은 n-gram 자질로 구성하고 카이제곱 통계량을 사용하여 자질을 선택하였다.

3. 실험 및 평가

제안한 개체명 인식 방법의 성능 평가를 위해 Bidirectional LSTM CRFs를 Tensorflow로 구현하였다. 개체명 인식 평가 데이터로는 2017년 국어 정보 처리 시스템 경진대회에서 배포한 개체명 말뭉치 문장을 사용하였다. 총 4,259 문장 중 3,000 문장을 학습 데이터로, 1,259 문장을 평가 데이터로 사용하였다. 전체적인 실험 성능은 가장 높은 성능을 보인 30 epoch로, 워드 임베딩, 음절 기반 워드 임베딩, 품사 임베딩은 각각 50차원으로 실험하였다. 실험 성능은 F_1-score로 평가하였다.

[표 1] 제안 모델 성능 평가 (%)

Accuracy	RNN	LSTM	Bi-LSTM CRFs
Baseline	73.26	78.16	80.16
+ Word Embedding	75.72	79.30	81.78
+ POS Embedding	76.18	80.07	83.26
+ GAZETTEER	77.27	82.06	**84.62**

[표 1]과 같이 학습 모델로는 RNN, LSTM, Bidirectional LSTM CRFs 모델을 비교하고 입력 자질로는 워드 임베딩, 음절 기반 워드 임베딩, 품사 임베딩, 개체명 사전을 조합하여 구성하였을 때의 성능을 실험하였다. 실험 결과 [표 1]과 같이 Bidirectional LSTM CRFs 모델에 모든 자질을 입력으로 사용한 경우 84.62%로 가장 높은 성능을 나타내었다.

4. 결론

본 논문에서는 한국어 개체명 인식을 위하여 Bidirectional LSTM CRFs 모델을 적용하고 자질로서 단어 임베딩, 품사 임베딩 그리고 사전 정보 자질을 이용하는 방법을 제안하였다.

향후에는 개체명을 인식할 때 개체명의 접두어, 접미어와 같은 세분화된 의미 정보를 사용하여 입력 자질을 일반화 및 최적화 하는 방법을 연구할 계획이다.

참고문헌

[1] N. V. Patil, A. S. Patil, B. V. Pawar, “HMM based Named-entity Recognition for inflectional language,” Computer, Communications and Electronic s, 2017.

[2] G Lample, M Ballesteros, S Subramanian, "Neural Architectures for Named Entity Recognition," NAACL, 2016.

[3] Y Luo, Y Cheng, O Uzuner, P Szolovits, "Segment convolutional neural networks (Seg-CNNs) for

classifying relations in clinical notes," JAMIA, 2017.

KACTEIL-NER: 딥러닝과 앙상블 기법을 이용한 개체명 인식기

박건우○, 박성식, 장영진, 최기현, 김학수
강원대학교 컴퓨터정보통신공학과
parkku01@kangwon.ac.kr, a163912@kangwon.ac.kr, buwak07@kangwon.ac.kr, pluto32@kangwon.ac.kr,
nlpdrkim@kangwon.ac.kr

KACTEIL-NER: Named Entity Recognizer Using Deep Learning and Ensemble Technique

Geonwoo Park○, Seongsik Park, Yoengjin Jang, Kihyoen Choi, Harksoo Kim
Kangwon National University Computer and Communication Engineering

요 약

개체명 인식은 입력 문장에서 인명, 지명, 기관명, 날짜, 시간 등과 같은 고유한 의미를 갖는 단어 열을 찾아 범주를 부착하는 기술이다. 기존의 연구에서는 단어 단위나 음절 단위를 입력으로 사용하였다. 하지만 단어 단위의 경우 미등록어 처리가 어려우며 음절 단위의 경우 단어 고유의 의미가 희석되는 문제가 발생한다. 이러한 문제들을 해결하기 위해 본 논문에서는 형태소 단위 개체명 인식기와 음절 단위 개체명 인식기를 앙상블하여 보정된 결과를 예측하는 개체명 인식기를 제안한다. 제안된 모델은 각각의 단일 입력 모델보다 향상된 F1-점수(0.8049)를 보였다.

주제어: 개체명 인식기, 인공 신경망, 앙상블, 형태소 확률 자질, 지명 사전 자질

1. 서론

개체명 인식은 입력 문장에서 고유한 의미를 갖는 단어 열을 찾아 인명, 지명, 기관명, 날짜, 시간과 같은 범주들을 부착하는 기술로, 정보 추출의 핵심 요소이다. 최근 자연어처리 분야에서 효율적인 입력 단위에 대해 많은 연구가 진행되고 있다[1-5]. 많은 개체명 인식 연구들은 단어(형태소) 단위나 음절 단위로 입력된다. 기존의 연구에서 단어 단위 개체명 인식기의 경우 미등록어 개체명 처리가 어려운 문제가 있고[2,3], 음절 단위 개체명 인식기의 경우 단어 고유의 의미가 희석되는 문제가 발생한다[4,5]. 이러한 문제들을 해결하기 위해 본 논문에서는 형태소 단위 개체명 인식기와 음절 단위 개체명 인식기의 결과를 이용해 상호 보완하는 모델을 제안한다.

2. 관련 연구

딥러닝(Deep learning)을 이용한 인공 신경망 기반 개체명 인식에서 성능을 향상시키는 방법 중 하나는 입력되는 단어 표상을 확장시키는 방법이 있다. 단어 표상을 확장시키는 방법 중에는 음절 단위 임베딩을 사용하는 방법, 단어 단위 임베딩을 사용하는 방법, 음절 단위 임베딩으로부터 단어 단위 임베딩 벡터를 유도하는 방법 등 여러 방법들이 존재한다[2-5].

앙상블 모델은 하나 이상의 단일 모델들을 통해 나온 다중 결과 값들을 이용해 단일 결과 값을 도출하는 모델이다[6]. 앙상블 모델에는 결과 값들 중 가장 많이 나온 결과 값을 선택하는 Voting 방법[2]과, 기계학습을 통해 최적의 결과 값을 선택하는 방법[7] 등이 있다.

본 논문에서는 다양한 입력 단위에 존재하는 문제점을 해결하기 위해 형태소 단위 개체명 인식기와 음절 단위 개체명 인식기를 이용한 앙상블 모델을 제안한다.

3. 형태소 음절 상호 보완 개체명 인식기

3.1 모델 구조도

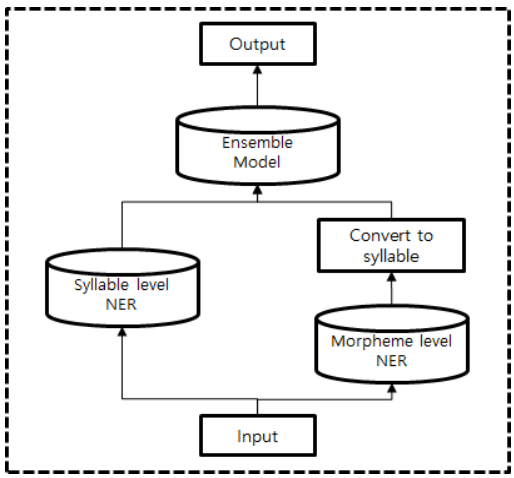


그림 1 형태소 음절 상호 보완 개체명 인식기 구조도

그림 1은 입력 문장을 각각 형태소 단위 개체명 인식기와 음절 단위 개체명 인식기에 입력하여 나온 값들을 이용해 앙상블 모델에 적용하여 최적의 출력 값을 도출

하는 모델의 구조도이다. 입력 문장은 형태소 단위, 음절 단위로 사용하여 형태소 단위 개체명 인식기와 음절 단위 개체명 인식기로 입력된다. 각 모델이 결과를 예측한 후 두 모델의 정보 결합을 위해 형태소 단위 개체명 인식기의 예측 값을 음절 단위로 변환하여 음절 단위 개체명 인식기의 예측 값과 함께 앙상블 모델의 입력이 된다. 마지막으로 두 입력을 통해 상호 보완된 개체명 예측 값을 최종 결과로 출력한다.

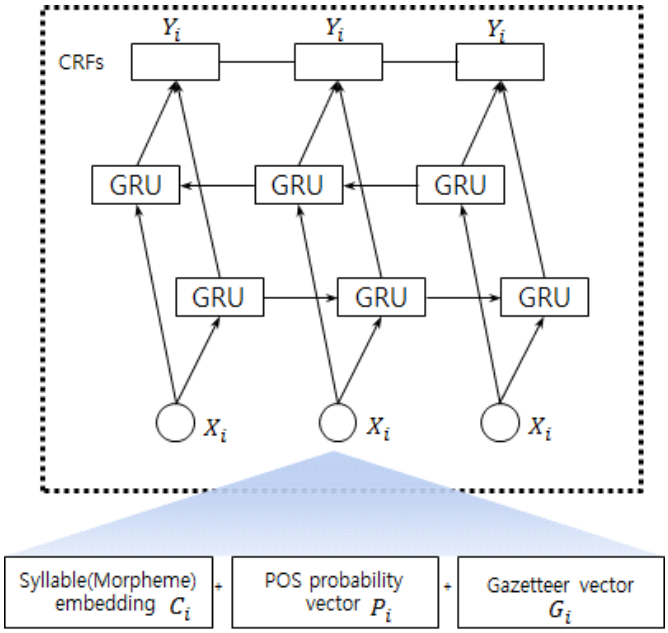


그림 2 개체명 인식기 구조도

그림 2는 음절 단위 개체명과 형태소 단위 개체명에 사용된 Gate Recurrent Unit Recurrent Neural Network Conditional Random Fields(GRU-CRF)[8]의 구조도 이다. 입력 단위의 임베딩과 함께 자질로 사용되는 형태소의 품사 확률 벡터, 지명 사전(gazetteer) 벡터를 결합하여 GRU-CRF의 입력으로 사용한다.

3.2 형태소 단위 개체명 인식기

그림 2와 같이, 형태소 단위 개체명 인식기의 입력은 세 개의 벡터로 구성된다. 형태소 임베딩(morpheme embedding: C_i)은 입력 형태소와 태그 쌍의 임베딩 벡터이고 형태소의 품사 확률 벡터(POS probability vector: P_i)는 형태소가 주어졌을 때 해당 형태소가 어떤 품사를 가지는지 출현 확률을 표현한 벡터이다. 지명 사전 벡터(gazetteer vector: G_i)는 개체명 사전 벡터이다. 개체명 사전 벡터는 개체명 사전의 형태소 분석된 개체명이 발생 하는지를 나타내는 벡터이다.

3.3 음절 단위 개체명 인식기

형태소 단위 개체명 인식기와 마찬가지로, 그림 2와 같이, 음절 단위 개체명 인식기의 입력은 세 개의 벡터로 구성된다. 음절 임베딩(Syllable embedding: C_i)은 입력 음절의 임베딩 벡터이고 음절 품사 확률 벡터(POS probability vector: P_i)는 n-gram 음절의 품사 확률 벡터이고 지명 사전 벡터(gazetteer vector: G_i)는 개체명 사전 벡터이다. 개체명 사전 벡터는 n-gram으로 이루어진 개체명 사전에 개체명이 발생 하는지를 나타내는 벡터이다.

음절의 품사 등장 확률 벡터 P_i를 얻기 위해 본 논문에서는 세종 형태소 말뭉치에서 내용어(고유 명사, 명사, 동사 등)로부터 n-gram 음절을 추출한다. 그 후, 각 n-gram 음절의 빈도를 계산하고 빈도를 확률로 변환한다. 표 1은 '대한민국'의 n-gram 음절을 보여준다. 표 2에서 보이듯이, '대한민국'은 고유 명사의 빈도가 다른 품사 태그의 빈도 보다 높다. 이 사실은 확률 벡터가 고유 명사와 일반 명사를 구별하는 단서가 될 수 있으며 대부분의 개체명이 고유 명사로 이루어져있기에 효과적인 자질이 될 수 있다.

표 1 n-gram 음절의 빈도

Bi-gram	고유명사	명사	동사	형용사	...
대한	1,191	511	0	7	0
한민	454	67	0	0	0
민국	396	29	0	0	0
Tri-gram	**고유명사**	**명사**	**동사**	**형용사**	**...**
대한민	344	2	0	0	0
한민국	344	2	0	0	0

지명 사전 벡터 G_i를 얻기 위해 우리는 개체명 사전에서 n-gram 음절을 추출 한다. 그 후, n-gram 음절 개체명의을 카이 제곱 분포의 확률 밀도 함수[9]를 사용하여 상위 N개의 n-gram 음절을 선택한다. 마지막으로, 입력된 n-gram 음절을 상위 N개의 n-gram 음절 개체명과 일치하는지 여부를 나타내는 이진 벡터로 변환한다. 표 2은 선택된 n-gram 음절의 일부를 보여준다.

표 2 범주별 상위 4개의 n-gram 음절

Bi-PER	Tri-PER	Bi-LOC	Tri-LOC	Bi-ORG	Tri-ORG
_김	_남궁	리_	1동_	사_	식회사
_박	레스_	시_	2동_	주식	주식회
_최	_황보	동_	1리_	식회	학교_
_이		면_	2리_	업_	_한국

표 2에서, 'Bi'와 'Tri'는 각각 bi-gram과 tri-gram을 의미하고 'PER', 'LOC', 'ORG'는 각각 인명, 지명, 기관명을 의미한다. 그리고 '_' 기호는 어절간 띄어쓰기를 의미한다. 'PER'의 bi-gram과 tri-gram은 사람의 이름이고 'LOC'의 bi-gram과 tri-gram은 행정 구역의 부분을 의미하고 'ORG'의 bi-gram과 tri-gram은 회사를 의미하는 단어이다. 이러한 사질은 지명 사전 벡터가 개체명 인식기에 대한 단서가 될 수 있음을 보여준다.

3.4 앙상블 모델

그림 1과 같이 형태소 단위 개체명 인식기와 음절 단위 개체명 인식기의 결과들을 앙상블 모델의 입력으로 이용한다. 앙상블 모델은 두 쌍의 연속된 개체명 태그들을 GRU-CRF의 입력으로 사용하고 한 개의 연속된 개체명 태그를 출력한다. 입력된 개체명 태그들은 각각 50차원의 임베딩 벡터로 구성되어있고 하나의 입력으로 사용하기 위해 100차원의 벡터로 결합한다.

4. 실험 및 결과

4.1 실험 환경

본 논문의 개체명 인식기에서는 2017 국어 정보 처리 시스템 경진대회의 말뭉치를 실험에 사용한다. 전체 말뭉치 3,660개에 Active Bagging[10,11]을 이용하여 추가적인 말뭉치를 생성하였고 평가에 사용된 데이터는 366개를 사용하였다. 개체명 태그는 인명, 지명, 기관명, 날짜, 시간 5개 이며 개체명 경계는 잘 알려진 음절 단위 BIO 태그를 이용하였다.

4.2 실험 결과

표 3은 제안한 모델의 성능을 보여준다. 표 3에서, 'M_NER' 은 각각 초기 값이 다른 3개의 형태소 단위 개체명 인식기이고 'S_NER' 은 각각 초기 값이 다른 3개의 음절 단위 개체명 인식기이다. Ensemble NER' 은 형태소 단위 개체명인식기와 음절 단위 개체명 인식기의 결과를 이용한 앙상블 모델이다.

표 3 성능 비교

	recall	precision	f1-measure
M_NER	0.6736	0.7511	0.7511
S_NER	0.6890	0.8191	0.7484
Ensemble NER	0.7683	0.8452	0.8049

표 3에서 보이듯이, Ensemble 모델이 M_NER 보다 F-1 점수 5.38%p 향상이 있었고, S_NER 보다 F-1 점수 5.65%p 향상이 있었다.

5. 결론

본 논문에서는 형태소 단위 개체명 인식기의 단점인 미등록어 처리 문제와 음절 단위 개체명 인식기의 단점인 단어 고유의 의미를 희석시키는 문제를 상호 보완하여 성능을 향상시키는 형태소 음절 상호 보완 개체명 인식기를 제안하였다. 실험 결과 형태소 및 음절을 입력하는 개체명 인식기보다 0.0538, 0.0565의 성능 향상을 보였다.

감사의 글

이 논문은 2016년도 정부(과학기술정보통신부)의 재원으로 정보통신기술진흥센터의 지원을 받아 수행된 연구임(No.R-20160906-004163, 빅데이터 자동 태깅 및 태그 기반 DaaS 시스템 개발) 또한, 2016년도 정부(미래창조과학부)의 재원으로 한국연구재단의 지원을 받아 수행된 연구임(No.2016R1A2B4007732)

참고문헌

[1] 심광섭. "기분석 어절 사전과 음절 단위의 확률 모델을 이용한 한국어 형태소 분석기 복제", 정보과학회 컴퓨팅의 실제 논문지, 제22권, 제3호, pp. 119-126, 2016.

[2] 이성희, 송영길, 김학수. "원거리 감독과 능동 배깅을 이용한 개체명 인식", 정보과학회논문지, 제43권, 제2호, pp. 269-274, 2016.

[3] 최윤수, 차정원. "Word Embedding 자질을 이용한 한국어 개체명 인식 및 분류", 정보과학회논문지, 제43권, 제6호, pp. 678-685, 2016.

[4] 나승훈, 민진우. "문자 기반 LSTM CRF를 이용한 개체명 인식", 한국정보과학회 학술발표논문집, pp. 729-731, 2016.

[5] 유홍연, 고영중. "품사 임베딩과 음절 단위 개체명 분포 기반의 Bidirectional LSTM CRFs를 이용한 개체명 인식", *한글 및 한국어 정보처리 학술대회 논문집*, pp. 105-110, 2016.

[6] 배지윤, 이민혁, 김유중, 태동현, 석준희. "재표집을 활용한 앙상블 인공 신경망 모델", 한국정보과학회 학술발표논문집, pp. 669-671, 2016.

[7] J. Feng, T. Zahavy, B. Kang, H. Xu and S. Mannor, Ensemble Robustness of Deep Learning Algorithms, *arXiv preprint arXiv:1602.02389*, 2016.

[8] C. Lee, LSTM-CRF Models for Named Entity Recognition. *IEICE Transactions on Information and Systems* 100(4), pp. 882-887.2017.

[9] A. D. Ball and G. D. Buckwell, in Statistics A Level, *Edited Macmillan Education*, UK, pp. 186-201. 1991.

[10] S. Lee, Y. Song, M. Choi and H. Kim. Bagging-based active learning model for named entity recognition with distant supervision. *Big Data and Smart Computing (BigComp)*, International Conference on. IEEE, 2016.

[11] 박건우, 이성희, 김학수. "개체명 사전과 원시 말뭉치를 이용한 준지도 학습 기반 개체명 인식 모델", 한국정보과학회 학술발표논문집, pp. 1757-1759. 2016.

Bi-directional LSTM-CNN-CRF를 이용한 한국어 개체명 인식 시스템

이동엽○, 임희석

고려대학교 컴퓨터학과
judelee93@korea.ac.kr, limhseok@korea.ac.kr

Korean Entity Recognition System using Bi-directional LSTM-CNN-CRF

Dong-Yub Lee○, Heui-Seok Lim
Dept. of Computer Science and Engineering, Korea University

요 약

개체명 인식(Named Entity Recognition) 시스템은 문서에서 인명(PS), 지명(LC), 단체명(OG)과 같은 개체명을 가지는 단어나 어구를 해당 개체명으로 인식하는 시스템이다. 개체명 인식 시스템을 개발하기 위해 딥러닝 기반의 워드 임베딩(word embedding) 자질과 문장의 형태적 특징 및 기구축 사전(lexicon) 기반의 자질 구성 방법을 제안하고, bi-directional LSTM, CNN, CRF과 같은 모델을 이용하여 구성된 자질을 학습하는 방법을 제안한다. 실험 데이터는 2017 국어 정보시스템 경진대회에서 제공한 2016klpNER 데이터를 이용하였다. 실험은 전체 4258 문장 중 학습 데이터 3406 문장, 검증 데이터 426 문장, 테스트 데이터 426 문장으로 데이터를 나누어 실험을 진행하였다. 실험 결과 본 연구에서 제안하는 모델은 BIO 태깅 방식의 개체 청크 단위 성능 평가 결과 98.9%의 테스트 정확도(test accuracy)와 89.4%의 f1-score를 나타냈다

주제어: NER, sequence labelling, deep learning

1. 서론

개체명 인식(Named Entity Recognition) 시스템은 문서에서 인명(PS), 지명(LC), 단체명(OG)과 같은 개체명을 가지는 단어나 어구를 해당 개체명으로 인식하는 시스템이다. 개체명 인식을 위한 전통적인 방법으로는 주로 hand-craft된 자질(feature)을 기반으로 학습하는 HMM(Hidden Markov Models), CRF(Conditional Random Fields)와 같은 통계 기반의 모델이 있다[1, 2]. 또한 개체명 인식이나 품사 태깅(Part-of-speech Tagging)과 같이 순서 라벨링(sequence labeling) 문제를 해결하기 위해 자질을 보강(augment) 하기 위한 방법으로 RNN(Recurrent Neural Networks)와 LSTM(Long-short Term Memory)를 활용한 연구가 있다[3, 4]. 최근에는 Bi-directional LSTM와 CNN(Convolutional Neural Network) 그리고 CRF 모델들을 함께 이용하여 end-to-end learning 방식으로 개체명 인식이나 품사 태깅 모델을 학습 할 수 있도록 딥러닝 기반의 모델을 활용한 연구가 있다[5].

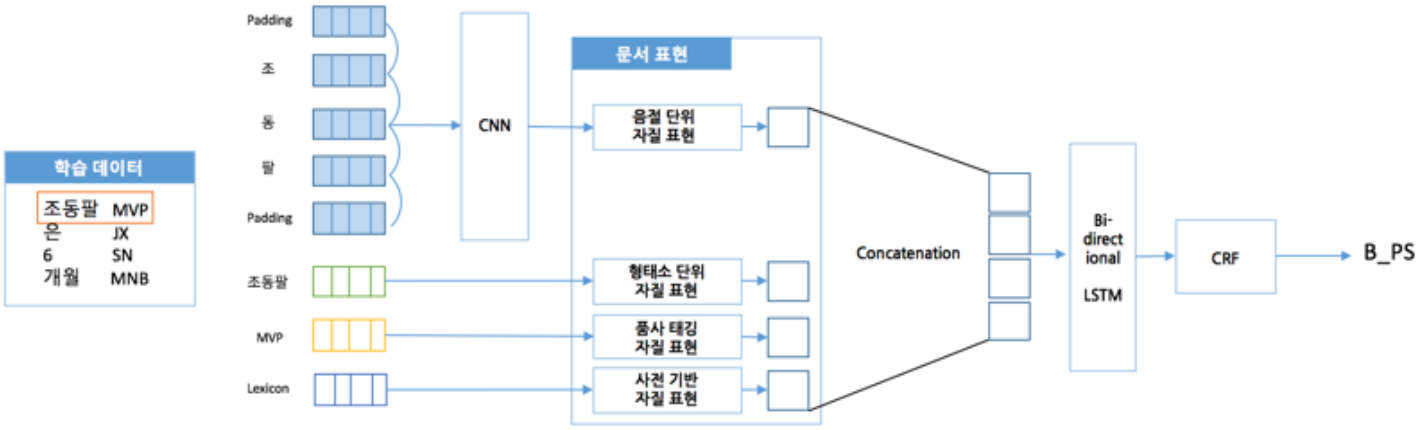
본 연구에서는 개체명 인식 시스템을 개발하기 위해 딥러닝 기반의 워드 임베딩(word embedding) 자질과 문장의 형태적 특징 및 기구축 사전(lexicon) 기반의 자질 구성 방법을 제안하고, bi-directional LSTM, CNN, CRF과 같은 모델을 통해 구성된 자질을 학습하는 방법을 제안한다. 실험은 2017 국어 정보 시스템 경진대회에서 제공한 2016klpNER 데이터를 이용하여 진행하였다. 전체 4258 문장 중 학습 데이터 3406 문장, 검증 데이터 426 문장, 테스트 데이터 426 문장으로 데이터를 나누어 실험을 진행하였다. 실험 결과 본 연구에서 제안하는 모델은 BIO 태깅 방식의 개체 청크 단위 성능 평가 결과 98.9%의 테스트 정확도(test accuracy)와 89.4%의 f1-score를 나타냈다.

2. 제안하는 과정

본 연구에서 제안하는 한국어 개체명 시스템의 전체 구조도는 [그림 1]과 같다. 2017 국어 정보 시스템 경진대회에서 제공하는 학습데이터는 문장이 형태소 단위로 나누어져 있다. “조동팔” 이라는 단어와 그에 해당하는 품사 태깅 결과가 주어졌을 때 제안하는 시스템은 4가지의 방법으로 문서를 표현한다. 음절 단위의 자질 표현을 구성하기 위해 단어를 이루는 음절 단위로 임베딩을 구성한 뒤 CNN을 통하여 음절의 자질을 추출 한 후 이를 음절 단위 자질 표현으로 활용한다. 형태소 단위로 나눠진 단어에 대해 glove vector를 이용한 워드 임베딩을 구성하여 이를 형태소 단위의 자질 표현으로 활용한다. 또한 학습 데이터에 포함되어 있는 품사 태깅 정보를 기반으로 품사 태깅에 대한 임베딩을 구성하여 이를 자질로 활용할 수 있다. 마지막으로 형태소 단위로 나눠진 단어를 대상으로 미리 구축된 기구축 사전을 이용하여 사전 기반의 자질을 표현할 수 있다. 학습 데이터를 이용하여 표현된 각각의 자질들을 연결(concatenation)한 뒤 이를 bi-directional LSTM의 입력으로 사용한다. 그 결과 LSTM은 은닉 상태(hidden states)를 계산하여 출력하고, 이 은닉 상태를 CRF의 입력으로 사용하여 최종적으로 형태소에 대응하는 개체명을 예측한다.

2.1 문서 표현(Document Representation)

2.1.1 CNN을 이용한 한국어 음절 단위의 자질 표현



[그림 1] 본 연구에서 제안하는 한국어 개체명 시스템의 전체 구조도

글자(character) 임베딩을 기반으로 자질을 추출하기 위해 CNN을 이용할 수 있다[6]. 본 실험에서는 CNN의 필터 크기(filter size)와 필터 개수를 다양하게 설정하여 성능 비교를 진행하였다. 최종적으로 2, 3, 4, 5 만큼의 필터 크기와 128 개의 필터 개수를 사용하였을 때, 가장 좋은 성능을 보였다. 또한 글자의 자질 표현 방법에 따른 성능 비교를 진행하기 위해 자소 단위와 음절 단위의 자질 표현 방법을 비교하였다. 그 결과 음절 단위로 글자를 구성하여 자질을 표현할 경우 자소 단위로 글자를 구성하여 자질을 표현하는 경우보다 f1-score가 약 2% 정도 만큼 높은 성능을 보였다.

2.1.2 Glove vector를 이용한 형태소 단위의 자질 표현

형태소 단위의 자질을 표현하기 위해 glove vector를 이용한 임베딩 공간을 구성하였다. Glove vector를 학습하기 위해 한국어로 구성된 위키피디아 데이터 약 345만건을 이용하였다.

2.1.3 품사 태깅 정보를 이용한 자질 표현

학습 데이터에는 앞, 뒤 단어에 대한 연관 성을 고려 할 때 활용될 수 있는 단어에 대한 품사 태깅 정보가 존재하여 이를 자질 표현으로 활용하였다. 그 결과 품사 태깅 정보를 이용하였을 때, 품사 태깅 정보를 이용하지 않았을 때 보다 f1-score가 약 1.9% 향상되었다.

2.1.4 기구축 사전 정보를 이용한 자질 표현

기구축 사전 정보를 이용하여 사전 기반의 자질을 표현하기 위해 gazette 기구축 사전을 이용하였다. 그 결과 기구축 사전 정보를 이용하였을 경우, 기구축 사전 정보를 이용하지 않았을 때 보다 f1-score가 약 1% 향상됨을 알 수 있었다.

2.2 Bi-directional LSTM과 CRF를 이용한 구성된 자질 학습

2.1절에서 구성된 문서 표현을 bi-directional LSTM의 입력으로 사용하여 각 형태소의 정보에 대한 은닉 상태를 계산할 수 있다. Bi-directional LSTM은 주어진 문장에 대해 각각 전향(forward), 후향(backward)으로 문장의 정보를 고려하여 보다 풍부하게 문장 정보를 표현하여 은닉 상태를 계산할 수 있다. CRF는 전 단계에서 계산된 은닉 상태에 대해 조건부 확률(conditional probability)를 계산하여 주어진 형태소에 대응하는 개체명을 예측한다.

2.3 데이터 구성 정보 및 하이퍼 파라미터(Hyper-Parameter) 설정

[표 1]은 본 실험에서 사용한 데이터의 구성 정보 및 각 모델들의 하이퍼 파라미터 설정 값을 나타낸다.

[표 1] 데이터 구성 정보 및 하이퍼 파라미터 설정 값

	Hyper-parameter	Value
Training data	word vocab size	6516
	char vocab size	1899
	entity tag vocab size	7
	morphological vocab size	45
	lexicon vocab size	6
Glove	window size	20
	dimension	100
CNN	filter sizes	2,3,4,5
	number of filters	128
	dropout	0.8
LSTM	initial state	0.0
	state size	600
	dropout	0.8
	training epoch	17
	initial learning rate	0.01
	decay rate	0.9
	char dimension	100

3. 실험 결과

제공된 데이터 전체 4258 문장 중 학습데이터 3406 문장, 검증 데이터 426 문장, 테스트 데이터 426 문장으로 데이터를 나누어 실험을 진행하였다. 성능 평가는 BIO 태깅 방식의 개체 청크 단위 성능 평가를 이용하여 진행하였다. [표 2]는 문서 표현을 하기위한 자질 표현 방법에 따른 한국어 개체명 인식 시스템의 성능을 나타낸다.

[표 2] 문서 표현 방법에 따른 한국어 개체명 인식 시스템 성능

Feature Representation		Accuracy	F1-score
형태소 단위		97.4	78.4
형태소 단위 + 글자	자소 단위	97.5	84.1
	음절 단위	97.8	86.2
형태소 단위 + 음절 단위 + 품사 태깅 정보		98.3	88.1
형태소 단위 + 음절 단위 + 품사 태깅 정보 + 사전 정보		**98.9**	**89.4**

4. 결론

본 연구에서는 2017 국어 정보시스템 경진대회에서 제공한 2016klpNER 데이터를 이용하여 한국어 개체명 인식 시스템을 개발하였다. 실험은 전체 4258 문장 중 학습 데이터 3406 문장, 검증 데이터 426 문장, 테스트 데이터 426 문장으로 데이터를 나누어 진행하였다. 문서 표현을 구성하기 위해 CNN을 이용한 한국어 음절 단위의 자질 표현, glove vector를 이용한 형태소 단위의 자질 표현, 품사 태깅 정보, 기구축 사전 정보를 이용하였다. 구성된 문서 표현을 bi-directional LSTM의 입력으로 사용하여 은닉 상태를 계산한 후 CRF의 입력으로 사용하여 최종적으로 형태소에 대응하는 개체명을 예측 하였다. 실험 결과 본 연구에서 제안하는 모델은 98.9%의 테스트 정확도(test accuracy)와 89.4%의 f1-score를 나타냈다.

참고문헌

[1] Lev Ratinov and Dan Roth. 2009. Design challenges and misconceptions in named entity recognition. In Proceedings of CoNLL-2009, pages 147-155.

[2] Gang Luo, Xiaojiang Huang, Chin-Yew Lin, and Zaiqing Nie. 2015. Joint entity recognition and disambiguation. In Proceedings of EMNLP-2015, pages 879-888, Lisbon, Portugal, September.

[3] Alan Graves, Abdel-rahman Mohamed, and Geoffrey Hinton. 2013. Speech recognition with deep recurrent neural networks. In Proceedings of ICASSP- 2013, pages 6645-6649. IEEE.

[4] Jason PC Chiu and Eric Nichols. 2015. Named entity recognition with bidirectional lstm-cnns. arXiv preprint arXiv:1511.08308.

[5] Ma, X. and Hovy, E. (2016). End-to-end sequence labeling via bi-directional LSTM-CNNs- CRF. In Proc. of ACL.

[6] Kim, Yoon. 2014. Convolutional neural networks for sentence classification. arXiv preprint arXiv:1408.5882 .

음절 기반의 CNN를 이용한 개체명 인식

박혜웅, 송영숙
아이리마인즈, 경희대학교
bage79@gmail.com, klanguage1004@gmail.com

Named Entity Recognition using CNN for Korean syllabic character.

Hye-woong Park, Young-Sook Song
Iriminds, KyungHee University

요 약

개체명 인식(Named Entity Recognition, 이하 NER)은 인명(PS), 기관명(OG), 장소(LC), 날짜(DT), 시간(TI) 등에 해당하는 개체명에 일정한 태깅 값을 주어 그 정보를 가시화하는 작업이다. 한국어 개체명 인식은 아직 그 자질이 충분히 밝혀져 있지 않아 자연어 처리 분야의 발전을 더디게 하는 한 요소로 작용하고 있다.

한국어가 음절 기반으로 단어를 형성하고 비교적 어순이 자유롭다는 특성이 있기에, 이런 특징을 잘 포착할 수 있는 "음절 기반의 Convolutional Neural Network(CNN)"의 아키텍쳐를 제안하여 66.80%의 성능을 보였다. 이 방법을 사용하면 형태소 분석등 개체명 이전 단계에서 발생하는 오류에 의해 개체명 인식(NER)의 성능이 떨어지는 문제를 해결할 수 있고, 조사나 어미 등을 제거하기 위한 후처리를 생략할 수 있다.

Convolutional Neural Network, Named Entity Recognition, 음절 기반

1. 서론

본고에서는 다른 전처리를 하지 않은 자연어 코퍼스에서 국어의 음절 임베딩을 통해 인명, 기관명, 장소 날짜, 시간의 다섯 가지 개체명에 속하는 단어를 추출하여 자동 태깅하려고 한다. 성능 평가를 위한 개체명 말뭉치로는 2017 국어 정보 처리 시스템 경진대회에서 배포한 6259개의 문장을 사용하였다. 본고의 구성은 2장에서 개체명 분야에서의 관련 연구를 소개하고 3장에서는 CNN 모델을 바탕으로 음절 단위의 임베딩 벡터를 통해 개체명을 포함하고 있는 단어를 인식할 수 아키텍쳐를 제안하고자 한다. 4장에서는 모델링의 구체적 특징과 테스트 결과를 분석한다.

2. 관련 연구

CoNLL(2003) shared task에서 사람, 위치, 조직, 기타의 개체명에 대한 데이터셋을 구축한 이후 개체명에 대한 연구는 크게 두 가지 방향으로 발전해 가고 있다고 할 수 있다. 먼저 Collobert et al. (2011)에서 지명을 추가하는 등 꾸준히 그 개체명의 범주를 늘리는 작업이 이어졌고 국내에서도 조은경(2014)에서 정보와 요구 기능을 하는 개체명과 그 동의어를 찾아서 개체명의 범주가 확장 될 수 있음을 밝히기도 했다.

또 한편에서는 RNN, CRF 등의 다양한 방법론(4~6)이 시도되고 있다. 국내에서는 아직 CNN을 활용한 개체명 연구가 활발하지 않지만 J. P.C. Chiu(2015)나 Emma Strubell(2017) 등의 연구를 통해 CNN도 개체명 인식에서 충분히 좋은 성능과 속도를 낼 수 있음을 확인할 수 있었다. 본 논문에서는 태깅되지 않은 원시 말뭉치를 직접 입력으로 사용하여, 음절 자체 정보만으로 음절 단위의 개체명 인식을 수행하는 있는 새로운 신경망 아키텍처를 제시한다.

3. 특징

CNN이 이미지 처리분야에서 높은 성능을 낼 수 있는 것은 이미지 전체 영역에 대하여 필터를 이용해 패턴을 스스로 학습하는 능력이 뛰어 나기 때문이다. CNN은 패턴이 이미지 안의 어떤 위치에 있는지 상관 없고, 회전, 대칭, 확대 등 어떠한 변형에도 강건한 학습이 가능하다.

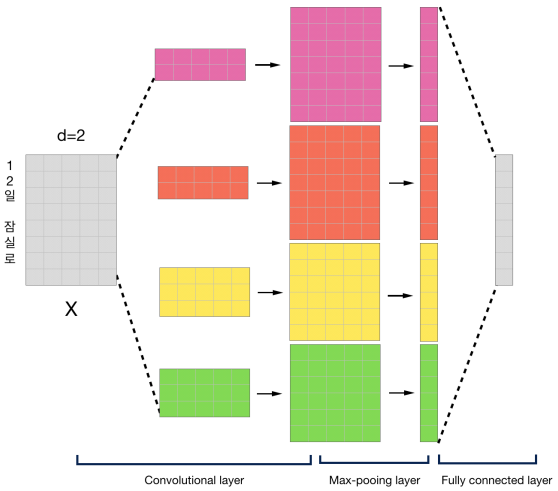
CNN의 이런 특성을 응용하면 문장안의 어떤 위치에 있든지, 다른 위치의 개체명에 영향을 주는 패턴을 인식할 수 있다고 가정하였다. 특히 한국어는 어순이 자유로워서 의존소와 지배소의 위치 관계가 일정치 않다. 또한 어미와 조사에 의한 음절의 변형이 많아 어절 단위의 임베딩을 이용하는 경우 저빈도 어절의 특징을 놓치는 일이 발생할 수 있다.

1음절 단어를 제외하면 음절 정보는 특별한 그 자체만으로 의미를 지닌 것은 아니라고 여겨서 개체명 연구에서는 음절 임베딩을 사용하는 모델이 적다. 본 논문에서 음절 임베딩을 사용한 이유는 복잡한 전처리 또는 후처리 단계를 생략하고 그 단계에서 발생하는 오류를 최소화하기 위함이다. 일반적으로 개체명 인식을 하기 위해서는 형태소 분석 결과가 필요하다. 조사, 어미 등이 제거된 명사 단위에 태깅을 해야 하기 때문이다. 하지만 음절 단위의 임베딩을 이용하면, 이러한 전처리가 생략될 수 있다.

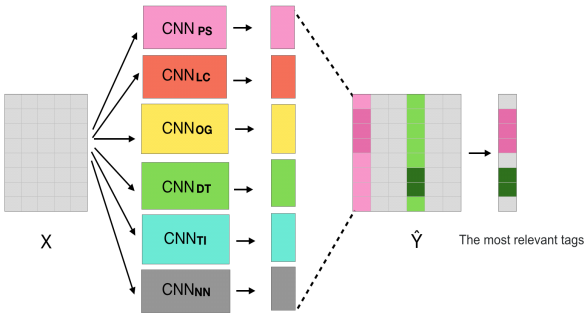
개체명을 인식하기 위해서는 품사 정보, 주위의 단어 또는 음절 정보, 사전 어휘부에 포함되어 있는지 여부 등을 특징으로 사용하기도 한다. 하지만 본 모델은 입력되는 원시 말뭉치의 순수한 음절 정보만을 학습하므로 매우 사용하기 쉬우면서 각 단계에서는 CNN 모델의 특성이 자연어 처리에서 잘 작동할 수 있도록 응용한 모델이다.

4. 모델링

클래스의 분류는 PS, LS, OG, DT, TI 를 포함하여, 개체명이 아닌 명사(NN)를 하나의 분류로 추가하여 총 6개로 정의하였다.



각각의 문자에 대한 분류 문제이기 때문에, 윈도우안의 음절의 위치를 최대한 보존하기 위함이다. Max-pooling된 결과를 모두 붙여 Fully-connected Layer의 입력으로 하였고, 최종적으로 각 윈도우 대하여, 한 개의 클래스에 대한 확률을 결과(Y^ps)로 출력한다.



한 개의 CNN을 사용하여 여러 개의 클래스로 분류하는 방법에 비하여 여러 개의 CNN이 각각의 클래스 분류 문제를 담당하게 되면 몇 가지 장점이 있다. 먼저 입력되는 클래스별로 임베딩 벡터를 각각 학습하여 성능을 향상시킬 수 있다. 마찬가지로 Convolution Layer 필터의 크기나 개수도 클래스별로 최적화 시킬 수 있다. 최종 각각의 CNN의 출력을 종합하여, 각 클래스에 대한 확률값(Y^)을 얻게 된다.

5. 평가방법

입력된 문장을 고정된 크기의 윈도우로 슬라이딩 하면서 여러 개의 입력 임베딩(X)을 얻는다.
예를 들어 윈도우의 크기가 5이고 입력 문장이 "12일 잠실로 이사간다."인 경우, 아래와 같이 첫 번째 윈도우에 대한 입력 임베딩은 "12일 잠"이고 레이블은 [TI, TI, TI, NN, LC] 이다. 마찬가지로 한 문자씩 슬라이딩하면 입력 임베딩은 "2일 잠실", "일 잠실로", " 잠실로 "의 순서로 얻게 된다.
모든 윈도우를 입력하여 모델에서 얻은 결과를 모두 합산하면, 각 문자에 대하여 가장 많이 예측된 클래스로 분류를 할 수 있다. 일부의 결과에서 예측 오류가 발생하더라도 다른 결과들에 의해서 오류를 보정해 주는 이점이 있다.

	정밀도 (Precision)	재현율 (Recall)	F-score
DT	69.43	69.43	69.43
LC	60.75	79.75	68.97
OG	58.39	66.20	62.05
PS	71.86	80.40	75.90
TI	60.00	55.56	57.69
합	64.09	70.29	66.80

6. 결론

본 논문에서는 CNN을 개체명 인식에 적용하였다. 원시 말뭉치의 음절 정보만을 이용하여, 분류외의 분야에 CNN을 활용할 수 있음을 보였다. 앞으로도 다양한 신경망 모델을 조합하여 인위적인 특징 추출 없이 성능을 개선하고자 한다.

참고문헌

[1] R. Collobert, J. Weston, L. Bottou, M. Karlen, K. Kavukcuogluand P. Kuksa. Natural Language Processing (Almost) from Scratch, Journal of Machine Learning Research (JMLR),2011.
[2] 조은경, 정보 기술 분야에서 개체명 동의어 연구, 언어과학연구 69, 2014.

[3] 나승훈, 민진우. 문자 기반 LSTM CRF를 이용한 개체명 인식. 한국정보과학회 학술발표논문집, 729-731, 2016
[4]이창기. Long Short-Term Memory 기반의 Recurrent Neural Network를 이용한 개체명 인식, 한국컴퓨터 종합학술대회 논문집, No.6, pp.645-647, 2015.
[5]이창기, 김준석, 김정회, 김현기, 딥러닝을 이용한 개체명 인식, 한국정보과학회 동계학술발표회 논문 집, No.12, pp.423-425, 2014.
[6]유홍연, 고영중, Bidirectional LSTM CRF 기반의 개체명 인식을 위한 단어 표상의 확장. 정보과학회논문지, 44(3), 306-313, 2017.
[7] J. P. C. Chiu, E. Nichols, Named Entity Recognition with Bidirectional LSTM-CNNs, arXiv: 1511. 08308, 2015
[8]Emma Strubell, Patrick Verga, David Belanger, Andrew McCallum, Fastand Accurate Entity Recognition with Iterated Dilated Convolutions, arXiv: 1702. 02098, 2017

언어 모델 다중 학습을 이용한 한국어 개체명 인식

김병재, 박찬민, 최윤영, 권명준, 서정연

서강대학교 컴퓨터공학과

Wizard3021@naver.com, cksals302@gmail.com, chldbsdu3773@gmail.com, corundum240@gmail.com,

profseojy@gmail.com

Korean Named Entity Recognition using Joint Learning with Language Model

Byeong-Jae Kim, Chan-min Park, Yoon-Young Choi, Myeong-Joon Kwon, Jeong-Yeon Seo

Sogang University, Dept. of Computer Engineering

요 약

본 논문에서는 개체명 인식과 언어 모델의 다중 학습을 이용한 한국어 개체명 인식 방법을 제안한다. 다중 학습은 1 개의 모델에서 2 개 이상의 작업을 동시에 분석하여 성능 향상을 기대할 수 있는 방법이지만, 이를 적용하기 위해서 말뭉치에 각 작업에 해당하는 태그가 부착되어야 하는 문제가 있다. 본 논문에서는 추가적인 태그 부착 없이 정보를 획득할 수 있는 언어 모델을 개체명 인식 작업과 결합하여 성능 향상을 이루고자 한다. 또한 단순한 형태소 입력의 한계를 극복하기 위해 입력 표상을 자소 및 형태소 품사의 임베딩으로 확장하였다. 기계 학습 방법은 순차적 레이블링에서 높은 성능을 제공하는 Bi-directional LSTM CRF 모델을 사용하였고, 실험 결과 언어 모델이 개체명 인식의 오류를 효과적으로 개선함을 확인하였다.

주제어: 개체명 인식, 다중 학습, 단어 표상, 심층 학습

1. 서론

다중 학습은 서로 다른 여러 작업들을 동시에 학습하는 방법을 말한다. 여러 작업을 동시에 학습하는 동안 작업들 사이의 공통점과 차이점을 활용하여 효과적으로 모델을 학습 할 수 있다. 다중 학습은 개별 모델을 학습하는 것 보다 예측 정확도에 대한 성능 향상을 기대 할 수 있다. 이와 같은 다중 학습은 자연어처리[1]뿐만 아니라 컴퓨터비전[2] 및 음성인식[3]에서도 성공적으로 연구되었다. 하지만 대다수 작업의 경우, 학습되는 데이터에 태그를 부착해야 하는 문제가 발생한다. 이를 해결하기 위해 본 논문에서는 별도의 태그 작업이 필요하지 않은 한국어 언어 모델과 개체명 인식 모델을 다중 학습하는 모델을 제안한다.

개체명 인식은 지명, 인명, 기관명, 날짜, 시간과 같은 고유한 의미를 갖는 단어를 문서에서 추출하고 그 종류를 결정하는 자연어 처리의 한 분야이다. 기존에 기계학습 알고리즘을 사용한 개체명 인식 연구는 사람이 직접 추출한 자질을 입력으로 사용했다. 이러한 방법은 자질을 추출하는데 많은 어려움과 시간이 요구된다. 하지만 최근 개체명 인식을 비롯한 다양한 자연어 처리 분야에 사용되는 심층 학습 모델은 자질 추출 작업 없이 모델을 학습시킬 수 있다는 장점이 있다. 특히 Bi-LSTM-CRF 모델은 많이 쓰이는 심층 학습 모델 중 하나로써 개체명 인식을 비롯한 순차적 레이블링 작업에서 우수한 성능을 보이고 있는 모델로, 입력 단어를 양방향 LSTM 의 입력으로 사용하여 각 입력에 상응하는 출력 계층의 태그간 의존성을 CRF 를 사용하여 모델링한 기법이다.

이와 같은 Bi-LSTM-CRF 모델의 성능은 입력 단어 표상에 의존적이다[4]. 따라서 단어 표상을 확장시켜 개체명 인식 시스템의 성능을 높이기 위한 연구[4,5]가 수행되었다.

본 논문에서는 단어 표상에 사전 학습된 단어 임베딩을 사용하였다. 추가적으로 품사 임베딩, 자소 임베딩 및 개체명 사전을 사용하여 단어 표상을 확장하였다.

또한 본 논문에서는 데이터에 대한 추가적인 레이블링 작업이 필요 없는 언어 모델의 특성을 이용하여 한국어 언어 모델과 한국어 개체명 인식 모델을 동시에 학습하는 다중 학습 모델을 제안한다. 제안하는 다중 학습 모델은 동일한 입력데이터를 효율적으로 학습 할 수 있다. 개체명 인식 모델이 학습되는 동안 입력으로 사용되는 학습 코퍼스에 대해서 동시에 언어 모델을 학습하게 된다. 결과적으로 다중 학습 모델은 사용 가능한 학습 코퍼스를 최대한 활용하는 방향으로 모델을 학습하게 된다. 학습 코퍼스를 최대한 활용함으로써, 은닉 계층을 효율적으로 학습시켜 기존 모델보다 기존 모델에 비해 우수한 성능을 얻을 수 있다.

본 논문의 구성은 다음과 같다. 2 장에서는 관련 연구를 소개하고, 3 장에서 제안하는 모델인 Bi-LSTM-CRFs 모델, 단어 표상의 확장 및 멀티 태스크 모델에

대해 소개한다. 4 장에서는 실험 결과를 분석하고, 마지막으로 5 장에서는 결론에 대해서 기술한다.

2. 관련 연구

개체명 인식에서 사용되는 기계학습 알고리즘은 사람이 추출한 자질을 입력으로 받아 최적의 가중치를 학습한다. 대표적인 방법으로 HMM, CRF, Structural SVM[6] 등이 있다. 하지만 최적의 자질 조합을 추출하는 과정에는 많은 연구와 시간이 필요했다. 이와 같은 문제를 해결하기 위해 딥 러닝 기반의 개체명 인식 연구가 많이 진행되고 있다. 특히 순환신경망의 종류 중 하나인 LSTM 모델과 CRF 를 결합한 Bi-LSTM-CRF[7]를 사용한 모델이 좋은 성능을 보였다. 이는 입력 단어의 앞뒤 문맥을 고려한 모델로써, 정방향(forward)과 역방향(backward)을 나타내는 두개의 LSTM 의 은닉 계층을 결합한다. 입력 단어의 Bi-LSTM 출력 결과와 인접 단어의 출력 결과 간의 의존성을 모델링 하기 위해 CRF 를 사용한 모델이다.

LSTM 의 입력으로 사용되는 형태소 단위의 단어 표상을 음절 단위로 세분화 시킨 연구로써 [8]에서는 각 입력 단어 문자열에 K 개의 합성 필터를 적용하여 음절 별 임베딩을 추출했고 이를 각 입력 단어 표상에 확장시켰다. 단어 단위보다 더 작은 문자 단위 입력을 표현하므로 처음 등장한 단어에 대해서도 유연하게 단어 표상을 표현 할 수 있다는 장점이 있다.

[9,10]에서는 학습되는 모델의 은닉 계층을 효율적으로 학습시키기 위해 멀티 태스크 학습을 연구하였다. 멀티 태스크 학습이란 두 가지 이상의 태스크를 파라미터를 공유하며 동시에 학습하는 방법으로 주 태스크에 대해 성능을 높일 수 있다는 장점이 있다. 멀티 태스크 기반 학습은 크게 두가지로 분류되는데 1) Hard parameter sharing 2) Soft parameter sharing 기법이 있다. Hard parameter sharing 은 일반적으로 멀티 태스크 학습에서 사용하는 기법으로써 학습이 진행되는 동안 모든 태스크 사이에 은닉계층을 공유한다. [11]에서는 더 많은 모델을 동시에 학습 할 수록 학습하고자 하는 모델의 과적합을 줄일 수 있음을 보여준다. 반면, Soft parameter sharing 은 Hard parameter sharing 과는 반대로 각 태스크의 모델은 고유의 은닉계층을 소유하는 기법으로 각 모델의 매개변수들이 유사해지도록 하기 위해 모델의 파라미터 사이의 거리를 정규화하는 특징이 있다.두 기법 모두 서로 다른 태스크를 동시에 학습하여 성능을 향상시켰다는 장점이 있다.

본 논문에서는 단어 표상을 표현하기 위해 사전 학습된 단어 임베딩과 품사 임베딩을 사용했고, 추가적으로 자소 단위 임베딩과 개체명 사전 자질을 사용하여 단어 표상을 확장하였다. 또한 모델의 과적합을 줄이고 성능을 높이기 위해 Hard parameter sharing 기법을 적용한 멀티 태스크 학습 모델을 제안한다.

3. 제안 모델

본 논문에서는 한국어 언어 모델 다중 학습을 이용한 한국어 개체명 인식 모델을 제안한다. 제안 하는 모델의 전체 구조는 그림 1 과 같다.

본 논문에서는 개체명 인식 연구에서 좋은 성능을 보이는 Bi-LSTM-CRFs 을 기본 모델을 사용하였다. 입력 단어를 표현하기 위해 사전 학습된 단어, 품사 임베딩을 사용하였고, 자소 임베딩 및 개체명 사전 자질을 사용해 단어 표상을 확장 시켰다. 또한 동일한 입력 데이터를 최대한 활용하기 위해 개체명 인식 모델과 한국어 언어 모델을 동시에 학습시키는 다중 학습 모델을 제안한다.

3.1 Bi-LSTM-CRFs 모델

양방향 LSTM 은 순차적으로 형태소 단위의 단어표상을 입력으로 사용한다. LSTM 의 출력 계층에선 입력 받은 단어 표상의 출력 결과와 인접 출력 결과 간의 의존성을 모델링하기 위해 각 출력 결과를 CRF 에 전달한다. 그림 2 은 기본적인 Bi-LSTM-CRFs 모델의 구조도이다.

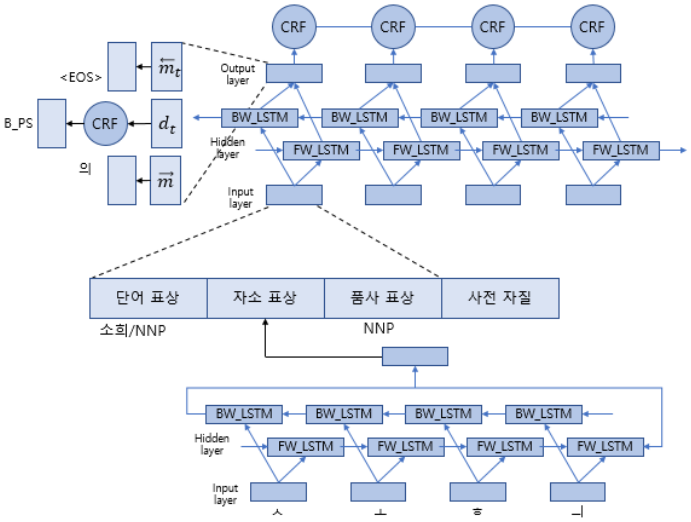


그림 1. 언어 모델 다중 학습을 이용한 개체명 인식 모델의 전체 구성도

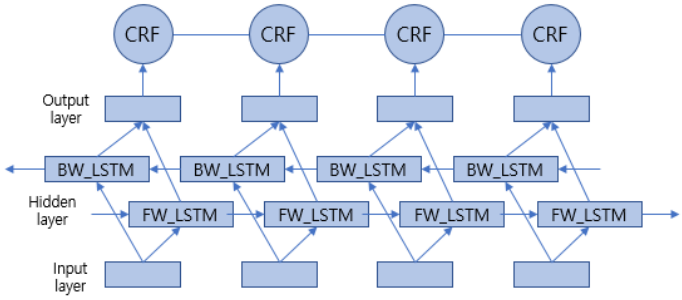


그림 2. Bi-LSTM-CRFs Model

3.2.1 단어 표상 확장

Bi-LSTM 모델의 성능은 단어 표상에 의존적이다[1]. 본 논문에서는 단어 표상을 확장하기 위해 사전 학습된 단어 임베딩, 문자열의 자소 임베딩, 형태소 임베딩, 음절 분포 그리고 개체명 사전 자질을 추가하여 성능을 향상시켰다.

3.2.2 단어 임베딩

제안하는 개체명 인식 모델의 기본 입력은 형태소 단위의 단어이다. 따라서 본 논문에서는 사전 학습된 단어 임베딩을 사용하였다. 또한 한국어의 언어적 특성상 형태소의 품사도 개체명 인식에서 중요하게 사용될 수 있기에 품사 임베딩을 사용하여 단어 표상을 확장했다. 원-핫 인코딩 방식 대신 임베딩 차원을 통해 품사를 표현함으로써, 품사가 가지고 있는 언어적 특성을 잘 표현하였다.

3.2.3 자소 임베딩

단어 표상을 확장하기 위해 입력되는 각 단어를 자소 단위로 분리하였다. 단어마다 구성된 자소의 개수가 다르기 때문에 분리된 자소는 Bi-LSTM 입력으로 사용된다. Bi-LSTM 의 마지막 은닉 계층을 결합하여 하나의 벡터로 변환하고 이를 각 단어를 표현하는데 사용하였다.

3.2.4 음절 분포 임베딩

단어를 구성하고 있는 음절 역시 3.2.3의 자소 임베딩과 같이 Bi-LSTM의 입력으로 사용된다. 음절 분포는 올해 경진대회에서 배포했던 훈련 셋에서 추출하였고 각 태그 별 분포를 나타내는 임베딩을 하였다[4]. Bi-LSTM 의 양방향 최종 출력 값 벡터를 결합하여 이를 전체 임베딩과 결합하여 사용하였다. 자소 임베딩과 달리 학습되지않는다.

3.2.5 개체명 사전

개체명 사전이란 개체명이 될 수 있는 명사들을 사전 형식으로 저장해 놓은 일종의 데이터베이스로, 성능 향상에 중요한 역할을 하는 자질로 사용 된다. 표(1)은 벡터로 표현된 개체명 사전 자질의 예시이다. “소희” 란 입력 단어가 개체명 사전에 등장할 경우, 등장한 태그의 값을 1로 표기하고, 등장하지 않으면 0으로 표기한다. 개체명 사전 벡터의 차원은 개체명 태그 카테고리 수와 같은 5차원이다.

PS	OG	LC	DT	TI
1	0	0	0	0

표 1. “소희”의 개체명 사전 자질 벡터

3.3 멀티 태스크 학습

앞선 3.1의 Bi-LSTM-CRFs 모델은 입력 문장과 이에 대응하는 개체명 태그에만 최적화 된 모델이다. 본 논문에서는 개체명 인식 모델의 학습이 진행되는 동안 입력 문장에 대한 한국어 언어 모델을 동시에 학습하는 다중 학습을 사용한 모델을 제안한다. 그림 1은 제안하는 멀티 태스크 학습 기반 개체명 인식 모델의 전체 구성도이다. 3.1에서 설명한 바와 같 Bi-LSTM-CRFs의 은닉계층인 $\overrightarrow{h_t}$와 $\overleftarrow{h_t}$는 개체명 태그를 예측하기 위해 개체명 출력 계층인 d_t에 서로 합쳐져서 연결된다. 이와 동시에 다중 학습 모델은 양방향 한국어 언어 모델을 학습한다. 예를 들어, “눈물을 훔치며 소희의 마지막을 보러 가야 할 사람은 두환만이 아니었다.” 란 문장이 있고 현재 입력 단어가 “소희” 일 때, $\overrightarrow{h_t}$와 $\overleftarrow{h_t}$는 인접 단어인 “며” 와 “의” 를 예측해야 한다. 따라서 양방향 언어 모델 계층인 $\overrightarrow{m_t}$와 $\overleftarrow{m_t}$는 식(1)과 같이 계산한다.

$$\overrightarrow{m_t} = \tanh\left(\overrightarrow{W_m}\,\overrightarrow{h_t}\right) - (1)$$
$$\overleftarrow{m_t} = \tanh\left(\overleftarrow{W_m}\,\overleftarrow{h_t}\right) - (2)$$

$\overrightarrow{W_m}$과 $\overleftarrow{W_m}$은 학습 가중치를 의미한다. 양방향 언어 모델의 성능은 제안하는 다중 학습 개체명 인식 모델의 주 목적이 아니다. 따라서 본 논문에서는 학습 속도 상향을 위해 m_t의 차원을 h_t에 비해 축소 시켰다. 축소된 차원을 통해 학습 모델은 한국어 언어적 특성에 일반화되어 과적합을 피할 수 다는 장점이 있다[10].

최종적으로 언어 모델 계층인 $\overrightarrow{m_t}$와 $\overleftarrow{m_t}$는 다음 등장 할 단어를 예측하기 위해 식(3,4)와 같이 소프트맥스 함수를 적용해 확률값을 계산한다.

$$P(w_{t+1}|\overrightarrow{m_t}) = softmax\left(\overrightarrow{W_q}\,\overrightarrow{m_t}\right) - (3)$$
$$P(w_{t-1}|\overleftarrow{m_t}) = softmax\left(\overleftarrow{W_q}\,\overleftarrow{m_t}\right) - (4)$$

각 언어 모델의 에러 함수는 순차 입력열 내 단어의 예측 확률에 대한 NLL(Negative Log Likelihood)으로 식(5), (6)와 같이 정의한다.

$$\overrightarrow{E} = -\sum_{t=1}^{T-1} \log\left(P(w_{t+1}|\overrightarrow{m_t})\right) - (5)$$
$$\overleftarrow{E} = -\sum_{t=2}^{T} \log\left(P(w_{t-1}|\overleftarrow{m_t})\right) - (6)$$

LSTM-CRFs 모델의 에러 함수와 각 언어 모델의 에러 함수를 더해 다중 학습을 이용한 개체명 인식 모델의 에러 함수를 정의한다. 최종적으로 본 논문에서 제안하는 개체명 인식 모델의 에러 함수는 다음 식(7)과 같다.

$$\tilde{E} = E + \gamma\left(\overrightarrow{E} + \overleftarrow{E}\right) - (7)$$

가중치 γ는 개체명 인식 모델과 언어 모델 사이의 상대적 중요성을 제어하는 역할을 한다. 본 논문에는 0.1을 사용하였다.

이와 같은 학습 과정을 통해 모델의 은닉 계층은 양방향 한국어 언어 모델을 학습함으로써 한국어의 문법적, 의미적 정보를 학습하게 된다. 은닉계층에 학습된 언어적 자질은 개체명 인식 모델이 개체명 태그를 예측하는데 재사용 된다. 따라서 개체명 인식 모델은 언어의 문법적 특성과 의미적 특성을 활용해 개체명 태그를 예측 할 수 있다.

결과적으로 멀티 태스크 학습 모델은 다음 단어의 등장 확률과 이전 단어의 등장 확률 그리고 현재 입력 단어에 대한 개체명 태그를 예측하는 학습에 최적화된다.

실험을 통해 멀티 태스크 학습을 통한 개체명 인식은 기존 Bi-LSTM-CRFs 모델보다 높은 성능을 보였다.

4. 실험

4.1 실험 환경

제안하는 멀티 태스크 학습을 이용한 Bi-LSTM-CRF 모델의 성능 평가를 위해 사용된 데이터는 2017 년 국어 정보 처리 시스템 경진 대회[12]에서 배포한 데이터를 사용하였다. 학습으로는 3,814 문장, 평가 데이터로 445 문장을 사용하였다. 모든 실험 성능은 F1-measure 평가 방법을 사용하였다. 개체명 인식 모델은 TensorFlow[13]로 구현하여 실험하였다.

4.2 단어 임베딩 실험

사전 학습된 단어 임베딩은 2016 년 국어 정보 처리 시스템 경진대회에서 배포한 데이터를 사용하였다. 임베딩의 차원은 50 차원이며 약 240,000 개의 단어로 구성되어 있다. 단어 임베딩 벡터를 랜덤으로 초기화한 경우보다 사전 학습된 단어 임베딩 벡터를 사용한 경우, F1-score 가 0.68% 높았다.

	prec	recall	F1
RandomVec	85.38	83.94	84.63
Pre-trained	86.02	84.67	85.31

4.3 자소 임베딩 및 품사 임베딩 실험

자소 임베딩의 차원은 50 차원으로 설정하였다. 자소 임베딩은 학습시 랜덤으로 초기화했으며 임베딩에 필요한 LSTM 의 은닉 계층의 차원은 50 차원이다. 자소 임베딩을 학습 할 시 자소 임베딩을 추가하지 않은 경우보다 0.71% 높은 성능을 얻을 수 있었다.

	prec	recall	F1
자소-	86.02	84.67	85.31
자소+	87.01	85.04	86.02

4.4 음절 분포 벡터

음절 임베딩의 차원은 12차원으로 설정하였다. LSTM의 은닉층의 차원도 12차원으로 설정하였다.

	prec	recall	F1
음절-	87.01	85.04	86.02
음절+	86.41	86.57	86.49

4.5 품사 임베딩 실험

품사 임베딩 차원은 16차원으로 설정하였다. 품사 임베딩은 약3.8G의 wiki 데이터를 사용하였다. 학습 모델로는 Word2Vec을 사용하였다.

	prec	recall	F1
품사-	86.41	86.57	86.49
품사+	88.08	86.57	87.75

4.6 개체명 사전 자질 실험

학습에 사용된 개체명 사전은 국어 정보 처리 시스템 경진대회에서 배포한 사전, 위키 코퍼스 인명사전 및 학습데이터에서 추가로 추출한 개체명 사전을 사용하였다. 개체명 사전을 자질로 사용 할 경우, 사용하지 않은 경우보다 1.43%의 향상된 성능을 얻을 수 있다.

	prec	recall	F1
개체명 사전-	88.08	86.57	87.75
개체명 사전+	88.91	89.45	89.18

4.7 멀티 태스크 학습 실험

학습에 사용된 언어 모델 계층 m_t의 차원은 50 차원으로 사용하였다. 동일한 개체명 데이터에 대해 언어 모델을 추가하여 멀티 태스크 학습을 진행 할 경우 개체명 인식 모델만 학습한 경우 보다 성능이 1.49% 증가하였다.

	prec	recall	F1
멀티태스크-	83.98	82.56	83.14
멀티태스크+	85.38	83.94	84.63

5. 결론

본 논문에서는 Bi-LSTM-CRFs를 사용한 개체명 인식 모델을 기반으로 단어 표상을 확장하기 위해 모델 학습 자소 단위 임베딩을 추가하였다. 또한 입력 데이터를 최대한 활용하기 위해 개체명 인식 모델과 한국어 언어 모델을 동시에 학습시키는 멀티 태스크 학습 기법을 적용된 모델을 제안하였다. 실험 결과, 제안하는 멀티 태스크 학습 모델은 멀티 태스크를 사용하지 않은 기본 Bi-LSTM-CRFs 보다 한국어 개체명 인식에서 향상된 성능을 보였다.

참고문헌

[1]Collobert, Ronan, and Jason Weston. "A unified architecture for natural language processing: Deep neural networks with multitask learning." *Proceedings of the 25th international conference on Machine learning*. ACM, 2008.

[2]Deng, Li, Geoffrey Hinton, and Brian Kingsbury. "New types of deep neural network learning for speech recognition and related applications: An overview." *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*. IEEE, 2013.

[3]Girshick, Ross. "Fast r-cnn." *Proceedings of the IEEE international conference on computer vision*. 2015.

[4]유홍연, 고영중. "Bidirectional LSTM CRF 기반의 개체명 인식을 위한 단어 표상의 확장." 정보과학회 논문지, 44.3 (2017.3): 306-313.

[5]나승훈, 민진우. "문자 기반 LSTM CRF 를 이용한 개체명 인식." 한국정보과학회 학술발표논문집, (2016.6): 729-731.

[6]Changki Lee, Junseok Kim, Jeonghee Kim, Hyunki Kim, "Named Entity Recognition using Deep Learning," Korean Institute of Information Scientists and Engineers(KIISE), No. 12, pp. 423-425, 2014.

[7]Huang, Zhiheng, Wei Xu, and Kai Yu. "Bidirectional LSTM-CRF models for sequence tagging." *arXiv preprint arXiv:1508.01991* (2015).

[8]Chiu, Jason PC, and Eric Nichols. "Named entity recognition with bidirectional LSTM-CNNs." *arXiv preprint arXiv:1511.08308* (2015).

[9]Zhang, Yu, and Qiang Yang. "An Overview of Multi-Task Learning." *National Science Review* (2017).

[10]Rei, Marek. "Semi-supervised Multitask Learning for Sequence Labeling." *arXiv preprint*

[11]Baxter, Jonathan. "A Bayesian/information theoretic model of learning to learn via multiple task sampling." *Machine learning* 28.1 (1997): 7-39.

[12]https://ithub.korean.go.kr/user/contest/contestIntroLastView.do

[13]https://www.tensorflow.org/

상대적 가중치 자질을 반영한 CRF 기반의 개체명 인식

정진욱
nlfactory@naver.com

Named Entity Recognition based on CRF reflecting relative weight

Jin-Wook Jeong

요 약

본 논문은 개체명 인식을 위해 CRF 모델을 이용해 분류를 수행했다. 개체명 후보를 개체명으로 식별에서 중의성 문제가 필요하다. 본 논문에서는 이러한 중의성 문제 해결을 위해 학습 셋으로부터 패턴과 형태적 특성을 고려해 개체명 후보를 최대로 선택하고 선택된 개체명 후보의 중의성과 정확도를 높이기 위해 주변의 문맥 자질과 분별 확률 모델인 CRF를 이용해 중의성 문제를 해결한다.

1. 서 론

개체명 인식(NER: Named Entity Recognition)은 입력된 자연어 문장에 나타난 특정 어휘가 사람, 장소, 기관, 날짜, 시간과 같은 미리 정의된 개체명 중 어떤 개체명인지를 식별해 태깅(tagging)하는 작업이다. 개체명 인식은 정보 추출에 있어서 하위 분야에 해당하며 최근 많은 관심을 받는 질의응답 시스템의 필수 기술이다.

잘 알려진 바와 같이 개체명에 해당하는 어휘는 중의성이 존재한다. 개체명 인식에서 개체명 사전을 이용할 수 있지만, 중의성 문제로 제대로 된 성능을 발휘할 수 없다. 본 논문은 개체명 인식 률을 높이기 위해 개체명 후보 선택의 커버리지를 넓게해 인식 가능성을 높이고, 개체명 후보 주변에 나타나는 문맥 자질(context feature)들과 확률 모델인 CRF를 이용해 개체명 인식을 수행한다.

2. 관련연구

개체명 성능을 높이기 위해 많은 방식이 도입되고 있다. 개체명 인식 초창기에는 개체명 사전이나 언어 문법(linguistic grammar) 기반으로 접근이 이뤄지다 현재는 지도 학습(supervised) 방식이나 빈지도 학습(semi-supervised) 학습 기반도 활용됐다. 최근에는 통계 모델(statistical model) 기반인 CRF와 LSTM 기반으로 개체명 인식 성능을 높이려는 연구가 있다.

이와 관련해 본 논문의 관련 연구로 최윤수 등[1]은 개체명 인식 자질 부족 문제를 활용하기 위해 워드 임베딩 자질로부터 추출한 벡터들에 대한 군집 정보를 CRFs의 워드 임베딩 자질로써 사용했다. 이 연구에서 사용한 자질의 종류로 형태소 자질, POS 태그, 형태소 길이, 어절의 위치, 개체명 사전에 존재 여부 값, 명사 유무 값을 활용했다.

박용민 등[2]이 있다. 이 연구는 도서, 영화, 노래, 음악, TV 프로그램에 대한 개체명 식별을 위한 데이터로 뉴스 기사를 활용했다. 이 연구는 주변 문맥 단어 및 거리를 이용하여 SVM을 활용해 제목 후보들을 추출하고 고유명사나 미등록어에 대한 사전을 구축하는 연구다.

Xuezhe 등[3]의 연구는 시퀀스 레이블 시스템을 구축하기 위한 지식을 이용하지 않고 뉴럴넷인 LSTM, CNN 그리고 CRF를 결합한 모델을 이용해 WSJ 코퍼스에 대한 개체명 인식을 수행했다.

본 논문에서는 개체명 인식의 중의성 문제 해결을 위해 패턴 학습을 통해 개체명 후보를 선택하고 중의성을 해결 하기위해 개체명 주변의 자질을 상대적으로 반영한 CRF 모델을 통해 애노테이션을 수행한다.

3. 규칙을 활용한 개체명 후보 선택

본 논문에서는 개체명을 분류하기 위한 전 단계로 개체명 후보를 선택한다. 개체명 후보는 개체명 식별을 위한 전 단계에 수행되는데 개체명이 될 수 있는 최대 후보가 될 수 있도록 개체명 후보를 최대한 선택함으로써 정답 커버리지를 높인다.

개체명 후보를 선택하려는 방법은 크게 두 가지다. 첫 번째로 학습 셋으로부터 패턴을 학습한다. 이를 위해 학습 셋에 존재하는 애노테이션된 개체명을 수집한다. 패턴 생성의 예로 <나무병원:OG>이라는 복합명사에 해당하는 개체명이 존재하면 복합명사의 형태소를 분리해 *병원의 형태로 분리하는 방식으로 패턴들을 수집한다. 이렇게 수집한 패턴에 부합한 문자열을 개체명 후보로 선택한다.

개체명 후보로 고려되는 대상으로 조사와 어미라는 형태소 앞에 위치하는 문자열을 개체명 후보로 선택한다. 이때 추상 명사인 경우 개체명 후보로 제외한다.

학습한 패턴과 형태적 특성을 고려해 수집된 개체명 후보는 개체명 후보의 커버리지를 최대한 높일 수 있게 한다.

4. 중의성 해결

4-1 중의성 해결을 위한 상대적 가중치 자질 수집

개체로 식별하기 위해 개체명 후보 중 동음이의로서 중의성이 있는지를 판단할 필요가 있다. 특히 중의성은 단음절 혹은 2음절인 단어는 중의성이 존재할 가능성이 높다.

이 때문에 규칙 기반의 방식으로 수집된 개체명 후보에 대해 중의성 해결이 필요하다. 예를 들어 '태봉'이라는 개체명 후보가 존재할 때 실제 문장에서는 다음과 같이 등장할 수 있다.

- <태봉:LC>은 901년 궁예에 의해 건국되어
- <태봉:PS>이가 제 몫을 헤다 말고

태봉은 장소(LC)를 의미하기도 하지만 사람(PS)을 의미하기도 한다. 중의성 문제를 해결하기 위한 접근 방법으로 형태적 특성을 고려한다. 이를 위해 접미사에 대한 음소(ㅇㅐㄴㅡ ?ㄴ)에 해당한다.
개체명 후보 주변에 존재하는 음절을 고려한다. 이때 개체명 후보에 멀수록 페널티를 부여하기 위해 개체명 후보에 가까울수록 가중치를 높게 해 개체명 후보 주변에 나타나는 인접 자질의 중요도를 반영했다. 이때 개체명 후보 주변에 나타난 자질이더라도 여러 개체명에 동시에 자주 나타나는 경우 중의성 자질로 분류해 패널티를 부여하거나 제외했다.

4-2 CRF 모델 구축

중의성 해결을 위해 분별 확률 모델(Discriminative Regression Models) 중 하나인 CRF(Conditional Random Field)를 이용한다. CRF에서 레이블을 선택하는 식은 다음과 같다..

$$y^* = \mathrm{argmax}_y p(y|x)$$

연속된 문자열에 대한 레이블(label)을 결정을 위해 벡터 x에 대해 p(y|x) 확률을 최대화하는 레이블 y*을 선택되도록 한다.

5. 실험

대회에서 제공한 개체명 태깅 말뭉치를 활용해 실험했다. 사용된 개체명 종류는 다음과 같다.

표 1 실험에서 사용된 개체명과 예

개체명	축약어	태깅예
Person	PS	<김:PS>기자, <이순신:PS>
Location	LC	<한국:LC>, <63빌딩:LC>
Organization	OG	<정부:OG>, <청와대:OC>
Date	DT	<10월 1일:DT>, <지난 1일:DT>
Time	TI	<3시 30분:TI>, <3시간 전:TI>

애노테이션은 크게 두 가지 원칙을 따른다. 애노테이션 방식은 정보통신단체 표준의 개체명 태그 세트 및 태깅 말뭉치 표준 권고안에 따라 최소 태깅 원칙을 태깅 지침으로 따른다. 최소 태깅 원칙이란 한 문장에서 개체명이 연이어 나올 때 최소 단위로 태깅하는 것을 의미한다.

<3월 20일:DT>에서 <4월 20일>까지

최소 태깅 원칙의 예외로 고유명사에 포함된 개체명인 경우 고유 명사를 먼저 개체명 후보로 선정한다. 예를 들어 '대한민국 임시정부'라는 고유 명사가 있을 때 고유 명사 내에 장소(LC)인 대한민국이 있더라도 고유 명사를 먼저 고려해 <대한민국 임시정보:OG>이라고 인식된다.

하지만 복합명사이지만 해당 복합명사가 개체명이 아닌 경우 분리해 식별한다. 예를 들어 "한미 양국"이라면 "<한:LC><미:LC> 양국"이라고 식별한다. 만약 테스트 세트가 표준 권고안의 태깅 지침을 따르지 않을 경우 제안한 방법에 페널티가 발생해 성능 측정이 제대로 이뤄지지 않을 수 있다.
구축된 CRF 모델을 활용해 개체명에 속하지 않았다면 애노테이션을 하지 않는 방식으로 게 된다. 본 논문에서 제안한 방법으로 구축된 NER을 이용해 대회에서 제공한 학습데이터를 학습 셋과 테스트세트를 7:3으로 나눠 실험한 결과 약 Precision 73.77%의 성능을 확보했다. 실험의 제한사항으로 학습 셋은 정답 셋이 아니여서 태깅이 되어있지 않거나 일부 태깅 오류의 문제가 있었다. 학습 셋을 개선하고 학습 데이터를 늘리면 테스트 셋에 대한 성능을 보다 높일 수 있을 것으로 기대된다.

6. 결론

본 논문은 개체명 식별을 위해 패턴을 학습을 통해 개체명 후보를 선택했고 개체명 선택의 중의성 문제를 해결하기 위해 연속 문자열의 특성을 반영하기 위해 CRF 모델을 적용했다. 본 논문에서 제안하는 패턴 학습의 방법은 개체명 사전에 존재하지 않는 개체명 후보라 할지라도 개체명 식별이 가능하며, 동음이의어일 지라도 자질의 상대적 가중치를 적용해 중의성을 해결할 수 있다. 후속 연구로 축된 개체명 인식기는 질의응답 시스템에 활용할 수 있도록 개체명 인식의 범위를 확장해 적용할 예정이다.

참 고 문 헌

[1] Yunsu Choi, Jeongwon Cha, "Korean Named Entity Recognition and Classification using Word Embedding Features)", Journal of KIISE, Vol. 43, No. 6, pp. 678-685, 2016.
[2] Yongmin Park, Jae Sung Le, "Named Entity Recognition and Dictionary Construction for Korean Title: Books, Movies, Music and TV Programs", Vol.3, No.7 pp.285-292, 2014.
[3] Xuezhe Ma and Eduard Hovy, End-to-end Sequence Labeling via Bi-directional LSTM-CNNs-CRF, pp. 1064-1074, 2016.

한글 및 한국어 정보처리 학술대회 제29권 제1호

인 쇄: 서기 2017년 10월 13일
발 행: 서기 2017년 10월 13일
발행인: 김병창, 박성배
편집인: 김병창, 박성배, 이공주, 고언숙
발행처: 사단법인 한국정보과학회
(137-849) 서울특별시 서초구 방배로 76 (방배동)
전화: 1588-2728 FAX: (02)521-1352
e-mail: kiise@kiise.or.kr
홈페이지: http://www.kiise.or.kr