ISSN 2005-3053

2013년도

제25회 한글 및 한국어 정보처리 학술대회

한글 및 한국어 정보처리

일시: 2013년 10월 11일(금) ~ 12일(토)

장소: 국민대학교 경상관

- 주최: 한국정보과학회, 한국인지과학회
- 주관: 한국정보과학회 언어공학연구회, 국민대학교 공개소프트웨어연구소
- 후원: 한국전자통신연구원, KISTI, (주)다음소프트, SK 플래닛, 네이버, 국민대학교, (주)나라인포테크, 유니닥스(주)

ISSN 2005-3053

2013년도

제25회 한글 및 한국어 정보처리 학술대회

한글 및 한국어 정보처리

일시: 2013년 10월 11일(금) ~ 12일(토)

장소: 국민대학교 경상관

■ 주최: 한국정보과학회, 한국인지과학회

■ 주관: 한국정보과학회 언어공학연구회, 국민대학교 공개소프트웨어연구소

■ 후원: 한국전자통신연구원, KISTI, (주)다음소프트, SK 플래닛, 네이버, 국민대학교, (주)나라인포테크, 유니닥스(주)

2013년도
제25회 한글 및 한국어 정보처리 학술대회

➽위원회

- **조직위원장** : 강승식(국민대)
- **조직위원** :

권혁철(부산대), 김경선(다이퀘스트), 김덕봉(성공회대), 김덕준(윕스),
김성묵(SK텔레콤), 김영길(ETRI), 김영섬(코난테크놀로지), 나동열(연세대),
박동인(KISTI), 박상규(ETRI), 박세영(경북대), 박종철(KAIST),
성열원(SK텔레콤), 심광섭(성신여대), 안동언(전북대), 옥철영(울산대),
윤준태(다음소프트), 이근배(POSTECH), 이민행(연세대), 이용석(전북대),
이재성(충북대), 이하규(성공회대), 이호석(호서대), 임해창(고려대),
임희석(고려대), 장두성(KT), 최재웅(고려대), 황규백(숭실대)

- **학술위원장** : 남기춘(고려대), 이재원(성신여대)
- **학술위원** :

강현규(건국대), 고영중(동아대), 김나리(코난테크놀로지), 김병창(대구가톨릭대),
김성동(한성대), 김유섭(한림대), 김재훈(해양대), 김판구(조선대),
김학수(강원대), 김현기(ETRI), 문유진(호남대), 박성배(경북대),
박소영(상명대), 양단희(평택대), 여상화(경인여대), 윤보현(목원대),
윤성희(상명대), 이경순(전북대), 이공주(충남대), 이도길(고려대),
이상곤(전주대), 이승우(KISTI), 이창기(강원대), 이현아(금오공대),
장명길(ETRI), 정한민(KISTI), 차정원(창원대), 최성필(KISTI),
최시영(싸이브레인 연구소), 최호섭(국립현대미술관), 황영숙(SK플래닛),

➽ 후원기관:

한국전자통신연구원, KISTI, (주)다음소프트, SK 플래닛, 네이버, 국민대학교, (주)나라인포테크, 유니닥스(주)

➽ 초대의 말씀

한글 및 한국어 정보처리 학술대회는 1989년 10월에 제1회 학술대회를 시작으로 매년 한글날 전후에 개최하였는데 지난 25년 동안 한해도 거르지 않고 올해로 25번째 학술대회를 개최하게 되었습니다. 언어처리 기술을 주요 내용으로 하고 있는 우리 학술대회는 자연어처리 기술을 기반으로 기계번역과 정보검색, 말뭉치 구축, 시맨틱웹, 온톨로지, 대화체 질의응답 시스템, 텍스트 마이닝, 빅 데이터, SNS 분석 등 다양한 분야로 발전되어 왔습니다.

미국의 경우 오래전부터 마이크로소프트, 구글 등 선도적인 기업들이 자연어 처리의 중요성을 인지하고 언어처리 그룹을 구성하여 차세대 핵심 기술로 매우 활발하게 연구-개발을 진행하고 있습니다. 국내에서는 대학, 연구소, 벤처기업을 중심으로 기계번역과 정보검색 솔루션을 개발해 왔으나, 최근에는 매우 많은 기업체들이 자연어처리 기술을 적용한 소프트웨어를 개발하고 있습니다.

언어처리 분야의 연구, 개발이 활성화됨에 따라 우리 학술대회는 2009년부터 문화체육관광부와 국립국어원의 후원으로 "국어 정보 처리 경진대회"를 시작하면서부터 5년 동안 계속해서 학술대회와 함께 경진대회를 개최해 왔으며, 2011년부터는 한국음성학회 워크샵을 병행 개최함으로써 우리 학술대회의 규모가 커지고 시너지 효과를 가져오고 있습니다.

이번 학술대회에는 37편의 논문이 접수되어 소정의 심사과정을 통하여 16편을 구두발표 논문으로 선정하였으며, 21편의 논문에 대해서는 전시(poster) 발표를 통해서 연구가 한층 더 발전되고 여러 연구자들의 다양한 의견을 수렴할 수 있는 기회를 제공하였습니다. 바쁘신 와중에서도 논문 모집에서부터 심사에 이르기까지 많은 수고를 아끼지 않으신 학술위원장을 비롯한 학술위원님들께 진심으로 감사드립니다.

언어처리 기술과 관련된 많은 학자들이 모여서 연구 및 학술의 교류와 활발한 토론이 이루어지는 학술대회가 되기를 기원합니다. 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.

2013년 10월 11일

제 25회 한글 및 한국어 정보처리 학술대회 조직위원장 강승식(국민대학교)
한국정보과학회 언어공학연구회 운영위원장 강승식(국민대학교)
한국인지과학회 회장 이성환(고려대학교)

프로그램 전체 일정표

■ 10월 11일(금요일)

장소: 국민대학교 경상관, 7호관

<table>
<tr><th>시 간</th><th colspan="2">일 정</th></tr>
<tr><td></td><td>한국음성학회 워크샵, 경상관 301호</td><td>국어정보처리 경진대회, 7호관 629호</td></tr>
<tr><td>13:00</td><td></td><td>개회식</td></tr>
<tr><td>13:20</td><td>개회식</td><td rowspan="2">지정공모 발표 작품
(주제: 세종 말뭉치 형태소 분석기)</td></tr>
<tr><td>13:30</td><td rowspan="4">주제발표-1
- 감성 세부분류 및 소셜 빅데이터 분석 기술,
김현기 박사(ETRI)
- 감성기반 대화형 탐색 및 큐레이션 기술,
장두성 박사(KT)
- 감성과 인지의 상호작용 연구,
최문기 교수(위덕대)</td></tr>
<tr><td>14:00</td><td rowspan="2">자유공모 발표 작품
(주제: 구문분석기, 표절검사 S/W,
옛한글입력기, 유사문서 분석기)</td></tr>
<tr><td>14:20</td><td>BREAK</td></tr>
<tr><td>14:40</td><td rowspan="4">주제발표-2
- 감정상태에 따른 발성변이, 음향특징 분석,
윤기덕 박사(서울대)
- 트렌드 분석 및 마이닝 검색,
윤준태 박사(다음소프트)</td></tr>
<tr><td>15:00</td><td>BREAK</td></tr>
<tr><td>15:10</td><td rowspan="2">자유공모 발표 작품
(주제: 구문분석기, 표절검사 S/W,
옛한글입력기, 유사문서 분석기)</td></tr>
<tr><td>15:40</td></tr>
<tr><td>16:00</td><td>BREAK</td></tr>
<tr><td>16:10</td><td colspan="2" rowspan="3">[초청강연] 사회: 옥철영 교수(울산대)
"언어자원 활용 방안: 구문분석 코퍼스를 중심으로", 이민행 교수(연세대 독문과)
"ExoBrain: 지능진화형 WiseQA", 박상규 센터장 (ETRI)</td></tr>
<tr><td>16:40</td></tr>
<tr><td>17:00</td></tr>
<tr><td>17:20</td><td colspan="2" rowspan="2">패널 토의: “언어 자원과 빅데이터, 그리고 NLP 활용 기술”
(Language Resources, Big Data, and NLP Technologies)</td></tr>
<tr><td>17:40</td></tr>
<tr><td>18:00</td><td colspan="2">시상식</td></tr>
<tr><td></td><td colspan="2">SIG-HLT 운영위원회 회의(장소: 북악정)
(참석자: 운영위원, 조직위원, 학술위원, 튜토리얼/패널토의 연사)</td></tr>
</table>

■ 10월 12일(토요일)

장소: 국민대학교 경상관

<table>
<tr><th>시 간</th><th colspan="2">일 정</th></tr>
<tr><td></td><td>경상관 301호</td><td>경상관 113호</td></tr>
<tr><td>09:00</td><td rowspan="4">논문발표 1A
좌장: 김성동 교수(한성대)</td><td rowspan="4">논문발표 2A
좌장: 박혁로 교수(전남대)</td></tr>
<tr><td>09:20</td></tr>
<tr><td>09:40</td></tr>
<tr><td>10:00</td></tr>
<tr><td>10:20</td><td colspan="2" rowspan="2">포스터 발표(경상관 301호 입구)</td></tr>
<tr><td>10:40</td></tr>
<tr><td>11:00</td><td rowspan="4">논문발표 1B
좌장: 이상곤 교수(전주대)</td><td rowspan="4">논문발표 2B
좌장: 이창기 교수(강원대)</td></tr>
<tr><td>11:20</td></tr>
<tr><td>11:40</td></tr>
<tr><td>12:00</td></tr>
<tr><td>12:20</td><td colspan="2">폐회</td></tr>
</table>

프로그램

10월 11일(금)

◉ 한국음성학회 워크샵 (국민대학교 경상관 301호)

13:20 ~ 16:10

13:20 ~ 13:30 개회식

13:30 ~ 15:00 - 감성 세부분류 및 소셜 빅데이터 분석 기술, 김현기 박사(ETRI)

- 감성기반 대화형 탐색 및 큐레이션 기술, 장두성 박사(KT)
- 감성과 인지의 상호작용 연구, 최문기 교수(위덕대)

15:00 ~ 15:10 휴식

15:10 ~ 16:10 - 감정상태에 따른 발성변이, 음향특징 분석, 윤기덕 박사(서울대)

- 트렌드 분석 및 마이닝 검색, 윤준태 박사(다음소프트)

◉ 초청강연 (국민대학교 경상관 301호)

16:10 ~ 18:00 사회: 옥철영 교수 (울산대)

초청강연 - 언어자원 활용 방안: 구문분석 코퍼스를 중심으로, 이민행 교수(연세대 독문과)

- ExoBrain: 지능진화형 WiseQA, 박상규 센터장(ETRI)

패널 토의 언어 자원과 빅데이터, 그리고 NLP 활용 기술

(Language Resources, Big Data, and NLP Technologies)

◉ 시상식, 감사패 (국민대학교 경상관 301호)

18:00 ~ 18:30

시상식 2013년 국어 정보 처리 시스템 경진대회 시상식

감사패 이근배(POSTECH), 박종철(KAIST)

10월 12일(토)

◉ 구두발표 1A (국민대학교 경상관 301호)

09:00 ~ 10:20	좌장: 김성동 교수 (한성대)
09:00 ~ 09:20	래티스상의 구조적 분류에 기반한 한국어 형태소 분석 및 품사 태깅 / 나승훈, 김창현, 김영길
09:20 ~ 09:40	Semi-CRF or Linear-Chain CRF? 한국어 형태소 분할 및 품사 태깅을 위한 결합 모델 비교 / 나승훈, 김창현, 김영길
09:40 ~ 10:00	CRF기반 한국어 형태소 분할 및 품사 태깅에서 두 단계 복합형태소 분해 방법 / 나승훈, 김창현, 김영길
10:00 ~ 10:20	대화형 개인 비서 시스템을 위한 하이브리드 방식의 개체명 및 문장목적 동시 인식 기술 / 이창수, 고영중

◉ 구두발표 1B (국민대학교 경상관 301호)

11:00 ~ 12:20	좌장: 이상곤 교수 (전주대)
11:00 ~ 11:20	중간언어와 단어정렬을 통한 이국어 사전의 자동 추출에 대한 성능 개선 / 권홍석, 서형원, 김재훈
11:20 ~ 11:40	준지도 학습을 통한 세부감성 분류 / 조요한, 오효정, 이충희, 김현기
11:40 ~ 12:00	바이오-이벤트 추출을 위한 피쳐 개발 / 이석준, 김영태, 황민국, 임수종, 나동열
12:00 ~ 12:20	질의 응답 시스템을 위한 반교사 기반의 정답 유형 분류 / 박선영, 이동현, 김용희, 류성한, 이근배

◉ 구두발표 2A (국민대학교 경상관 113호)

09:00 ~ 10:20	좌장: 박혁로 교수 (전남대)
09:00 ~ 09:20	상품 평가 텍스트에 암시된 사용자 관점 추출 / 장경록, 이강욱, 맹성현
09:20 ~ 09:40	읽기 매체의 다양성과 흥미도를 고려한 가독성 측정 / 김아영, 박성배, 이상조
09:40 ~ 10:00	토픽 모델 표현에 기반한 모바일 앱 설명 노이즈 제거 / 윤희근, 김솔, 박성배
10:00 ~ 10:20	Latent Structural SVM을 확장한 결합 학습 모델 / 이창기

◉ 구두발표 2B (국민대학교 경상관 113호)

11:00 ~ 12:20	좌장: 이창기 교수 (강원대)
11:00 ~ 11:20	토픽 모델을 이용한 수학식 검색 결과 재랭킹 / 양선, 고영중
11:20 ~ 11:40	개체명 인식을 위한 개체명 사전 자동 구축 / 전원표, 송영길, 최맹식, 김학수
11:40 ~ 12:00	한국어 의존 파싱을 이용한 트리플 관계 추출 / 곽수정, 김보겸, 이재성
12:00 ~ 12:20	P 언어를 이용한 한글 프로그래밍 / 최시영

◉ 포스터발표 (국민대학교 경상관 301호 입구)

10:20 ~ 11:00

P01	영한 기계번역 시스템의 개선을 지원하는 영어 구문 규칙 관리 도구 / 김성동, 김창희, 김태완
P02	음식메뉴 개체명 인식을 위한 음식메뉴 사전 자동 구축 방법 / 구영현, 유성준
P03	주식 관련 기사 분류 및 긍정 부정 판단을 통한 종목 추천 시스템 / 이유준, 박정우, 전민재, 최준수, 한광수
P04	학생 답안 정보를 활용한 반자동 정답 템플릿 구축 도구 / 장은서, 강승식
P05	상품평 분석을 통한 상품 평가 요약 시스템 / 김제상, 정군영, 권인호, 이현아
P06	LDA를 이용한 트윗 유저의 연령대, 성별, 지역 분석 / 이호경, 천주룡, 송남훈, 고영중
P07	문장 길이 축소를 이용한 구 번역 테이블 용어 추출 성능 향상 / 정선이, 이공주
P08	빅데이터 기반의 오피니언 마이닝을 이용한 기업 가치 평가 시스템 개발 / 이정태, 천민아, 임상우, 전병석, 김재훈, 한영우
P09	CopyCheck : 한글문서 표절검사 시스템 / 박소영, 장은서, 권도형, 강승식
P10	연관 어휘 추출을 통한 질의어 관련 이슈 탐지 / 김제상, 김동성, 조효근, 이현아
P11	CRFs를 이용한 의존구조 구문 레이블링 / 정석원, 최맹식, 김학수
P12	음절 ngram 기반의 미등록 어휘 추정기 구현 / 신준수
P13	한국어 부분언어에 대한 문법 정의 및 GLR 파싱 / 김지현, 정병채, 이재성
P14	등급 재현율: 이중 언어사전 구축에 대한 평가 방법 / 서형원, 김재훈
P15	한국어 품사 및 동형이의어 태깅을 위한 마르코프 모델과 은닉 마르코프 모델의 비교 / 신준철, 옥철영
P16	Y-HisOnto: Q&A 시스템에서의 활용을 위한 역사 온톨로지 모형 / 이인근, 정재은, 황도삼
P17	블로그 포스트의 자동 분류 시스템 / 김수아, 조희선, 이현아
P18	코사인 유사도 기법을 이용한 뉴스 추천 시스템 / 김상모, 김형준, 한인규
P19	어휘지도(UWordMap)를 이용한 용언의 다의어 처리 / 배영준, 옥철영
P20	모바일 기기에서 일정 관리를 위한 개체명 인식 / 장은서, 강승식, 이재원, 김도현
P21	접속 부사의 사용에 따른 설득과 보도의 대응 분석 / 김혜영, 강범모

➽ 목차

- **포스터발표 (국민대학교 경상관 301호 입구)**

영한 기계번역 시스템의 개선을 지원하는 영어 구문 규칙 관리 도구
- 김성동, 김창희, 김태완

음식메뉴 개체명 인식을 위한 음식메뉴 사전 자동 구축 방법
- 구영현, 유성준

주식 관련 기사 분류 및 긍정 부정 판단을 통한 종목 추천 시스템
- 이유준, 박정우, 전민재, 최준수, 한광수

학생 답안 정보를 활용한 반자동 정답 템플릿 구축 도구
- 장은서, 강승식

상품평 분석을 통한 상품 평가 요약 시스템
- 김제상, 정군영, 권인호, 이현아

LDA를 이용한 트윗 유저의 연령대, 성별, 지역 분석
- 이호경, 천주룡, 송남훈, 고영중

문장 길이 축소를 이용한 구 번역 테이블 용어 추출 성능 향상
- 정선이, 이공주

빅데이터 기반의 오피니언 마이닝을 이용한 기업 가치 평가 시스템 개발
- 이정태, 천민아, 임상우, 전병석, 김재훈, 한영우

CopyCheck : 한글문서 표절검사 시스템
- 박소영, 장은서, 권도형, 강승식

연관 어휘 추출을 통한 질의어 관련 이슈 탐지
- 김제상, 김동성, 조효근, 이현아

CRFs를 이용한 의존구조 구문 레이블링
- 정석원, 최맹식, 김학수

음절 ngram 기반의 미등록 어휘 추정기 구현
- 신준수

한국어 부분언어에 대한 문법 정의 및 GLR 파싱
- 김지현, 정병채, 이재성

등급 재현율: 이중 언어사전 구축에 대한 평가 방법
- 서형원, 김재훈

한국어 품사 및 동형이의어 태깅을 위한 마르코프 모델과 은닉 마르코프 모델의 비교
- 신준철, 옥철영

Y-HisOnto: Q&A 시스템에서의 활용을 위한 역사 온톨로지 모형
- 이인근, 정재은, 황도삼

블로그 포스트의 자동 분류 시스템
- 김수아, 조희선, 이현아

코사인 유사도 기법을 이용한 뉴스 추천 시스템
- 김상모, 김형준, 한인규

어휘지도(UWordMap)를 이용한 용언의 다의어 처리
- 배영준, 옥철영

모바일 기기에서 일정 관리를 위한 개체명 인식
- 장은서, 강승식, 이재원, 김도현

접속 부사의 사용에 따른 설득과 보도의 대응 분석
- 김혜영, 강범모

제25회 한글 및 한국어 정보처리 학술대회
25th Annual Conference on Human and Cognitive Language Technology

구두발표 1A

래티스상의 구조적 분류에 기반한 한국어 형태소 분석 및 품사 태깅

나승훈○, 김창현, 김영길
한국전자통신연구원
nash@etri.re.kr, chkim@etri.re.kr, kimyk@etri.re.kr

Lattice-based discriminative approach for Korean morphological analysis and POS tagging

Seung-Hoon Na○, Chang-Hyun Kim, Young-Kil Kim
Natural Language Processing Laboratory
Electronics and Telecommunication Research Institute

요 약

본 논문에서는 래티스상의 구조적 분류에 기반한 한국어 형태소 분석 및 품사 태깅을 수행하는 방법을 제안한다. 제안하는 방법은 입력문이 주어질 때 어휘 사전을 참조하여, 형태소를 노드로 취하고 인접형태소간의 에지를 갖도록 래티스를 구성하며, 구성된 래티스상 가장 점수가 높은 경로상에 있는 형태소들을 분석 결과로 제시하는 방법이다. 실험 결과, ETRI 품사 부착 코퍼스에서 기존의 1차 linear-chain CRF에 기반한 방법보다 높은 어절 정확률 그리고 문장 정확률을 얻었다.

주제어: 래티스, 구조적 분류, 형태소 분석, 품사 태깅

1. 서론

한국어 형태소 분석을 위한 *규칙기반 방법* [6,7,8,9,18,20,22,23,26]은 분석에 필요한 규칙을 수작업으로 구축하기 때문에, 개발 비용이 높고, 새로운 도메인에 대해 적응력이 떨어지는 단점을 지닌다. *통계기반 방법*은 대규모 품사 부착 말뭉치로부터 형태소 분할 및 품사 태깅에 필요한 규칙 및 확률 모델을 자동 또는 반자동으로 학습하는 방식으로 [3,4,11,14,15,16,24,25,27], 수작업이 거의 필요 없고, 성능이 우수하며, 타 도메인으로의 적용성이 높고, 기존의 어휘 사전과의 하이브리드가 가능하다는 점 등의 장점으로 인해, 현대의 대부분의 품사 태깅 연구가 이에 기반을 두고 있다.

그러나, 자동 번역 등과 같은 실제 응용을 목표로 하는 *응용 지향 형태소 분석기*에서는 어휘 사전은 여전히 필수적인 자원이다. 특히, 자동 번역에서는 목적언어의 대역어가 원시언어의 어휘로부터 얻어지기 때문에, 어휘 사전은 기본적인 리소스가 된다. 응용 지향 형태소 분석기에서는 최종 응용에서의 성능을 높이기 위해, 오랜 기간 동안 튜닝과정을 거쳐 사전의 규모화가 이루어진 경우가 많다. 이러한 사전의 규모화는 최근 웹의 크기가 빅데이터화되고, 개체명 (named entity)을 자동으로 추출하는 웹 마이닝 기법이 지속적으로 발전되고 있는 추세 속에서 더욱 용이해지고 있다.

반면, 응용 지향 형태소 분석기에서는 일반적으로 통용되는 형태소 단위 (Sejong코퍼스에서 정의되는 단위)를 사용하지 않고 자체적으로 형태소 단위를 재정의하여 사용하기도 한다 [19]. 그런데, 이러한 경우 해당 단위의 품사 부착 말뭉치는 규모화가 이루어져 있지 않고 소규모에 그치고 있기도 하다. 예를 들면, ETRI의 자동번역을 위해 구축된 품사 태깅 말뭉치는 10만 문장 정도로, 이는 2011년에 배포된 세종 말뭉치의 80만 문장보다 현저히 적은 규모이다.

본 논문에서는 어휘 사전은 규모화되어있으나 학습 데이터가 소규모인 경우, 형태소 분석의 정확도를 높이기 위해 학습 기반 *구조적 분류 모델 (structured classification)*을 적용하는 방법을 제안한다. 제안하는 방법은 래티스 기반의 구조화 분류 방법으로, 먼저 입력문으로부터 어휘 사전을 참조하여, 사전에 나타난 개별 형태소를 노드로 취하고 인접 형태소간에 에지를 구성하여, 입력문의 래티스 (*lattice*)를 구성한다. 다음으로, 이렇게 얻어진 래티스상에서 가장 점수가 높은 최적의 경로를 찾아, 이 최적 경로 상에 있는 형태소열을 분석 결과로 제시한다. 경로의 점수는 구성하는 에지의 점수의 합으로 이루어지며, 각 에지의 점수는 에지의 자질 벡터와 자질 가중치 벡터의 내적으로 정의된다. 이때, 자질 가중치 벡터는 학습 데이터로부터 구조적 분류 알고리즘을 통해 학습된다. 이러한 래티스 기반 한국어 형태소 분석은 기존의 전통적인 규칙 기반 방법에서 활용되었던 방식이나, 기존의 연구는 모두 HMM의 생성 모델에 기반을 둔 반면, 본 논문은 래티스상의 구조적 분류 모델을 적용한다는 점에서 기존 연구와 차별점이 있다.

ETRI 한국어 품사 부착 코퍼스상에서 실험 결과 제안 방법은 1차 linear-chain CRF기반 방법에 비해 높은 성능을 보여주었다.

2. 관련 연구

제안 방법과 가장 유사한 기존 연구는 일본어 형태소 분석을 위해 CRF를 적용한 연구인 [2]이다. [2]에서는, 본

연구와 마찬가지로, 입력문에 대해 어휘 사전으로부터 래티스를 구성하고, 래티스상의 임의의 경로에 대한 확률을 입력문에 대한 조건부 확률 (conditional probability)로 모델링하여, 가장 높은 확률을 갖는 경로를 찾는 문제로 형태소 분석 문제를 형식화했다. 그러나, 제안 방법과 기존 연구[2]와는 다음의 차이가 있다. 첫째, [2]에서는 가질 가중치 학습 방법이 CRF로, 분류 모델(discriminative model)에 속하나, 제안 방법은 별도의 조건부 확률을 정의하지 않는 SVM과 같은 분류 함수(discrimiantive function)방법에 속한다. 둘째, [2]에서는 사용되는 자질이 에지에 참여하는 두 노드로 국한된 first-order 방식만을 사용하고 있으나, 제안 방법은 참고하는 자질이 에지의 갯수가 2개인 second-order의 자질의 사용까지 포함한다. 셋째, [2]는 미등록어에 대한 문제를 남겨두었으나, 본 연구에서는 linear-chain기반 CRF를 이용하여 미등록어를 추가하는 방법도 함께 제시한다.

본 연구과 밀접하게 관련된 다른 연구로 [5]의 그래프 기반 의존 파싱을 들 수 있다. 제안 방법과 유사하게, [5]에서는 의존 파싱 문제를, 입력문에 대해 의존성을 에지로 하여 그래프를 구성하고, 해당 그래프에서 최적의 트리 (tree)를 찾는 문제로 파싱문제를 형식화했다. 본 논문의 방법은 [5]의 그래프 기반 의존 파싱에서 그래프가 래티스로, 트리가 경로로 제한된 특수한 경우라고 볼 수 있다. 그러나, 본 논문은 그래프 기반 분류 방법을 품사 태깅 문제에 적용하여, [5]의 의존 파싱 목적과 근본적인 차이가 있다. 저자의 지식에 따르면, 현재까지 래티스 기반 구조적 분류 방식을 한국어에 적용하여 성능 평가를 수행한 연구는 없었다.

3. 제안 방법

그림 1은 제안하는 래티스상 구조적 분류에 기반한 한국어 형태소 분석의 흐름도를 보여준다. 다음 절에서 제안 방법을 보다 형식적으로 기술하기로 한다 (이해를 위해, [5]의 표기를 주로 채택한다).

3.1 문제 정의

입력 문 $\boldsymbol{x}$에 대해서, 래티스 $G=(V,E)$는 방향성 그래프로, V은 입력문 x에서 부분문자열 중 사전에 있는 모든 형태소 [표층 문자열, 품사 태그]를, E는 인접하는 모든 형태소간의 가중치를 갖는 에지 (weighted edge)집합을 의미한다. 모든 래티스는 {0,1}이라는 2개 특수 노드를 갖는다. 여기서, 0은 문장의 시작을 가리키는 시작 형태소를 1은 문장의 끝을 가리키는 마지막 형태소이다. 에지 (i,j)는 노드 i에서 노드 j로의 전이 (transition)를 의미한다.

입력문의 래티스상의 경로 (path) $\boldsymbol{y}$는 시작 노드 0과 마지막 노드 1를 연결하는 경로로, 구성하는 에지 순차열 (edge sequence)이라고 볼 수 있다.

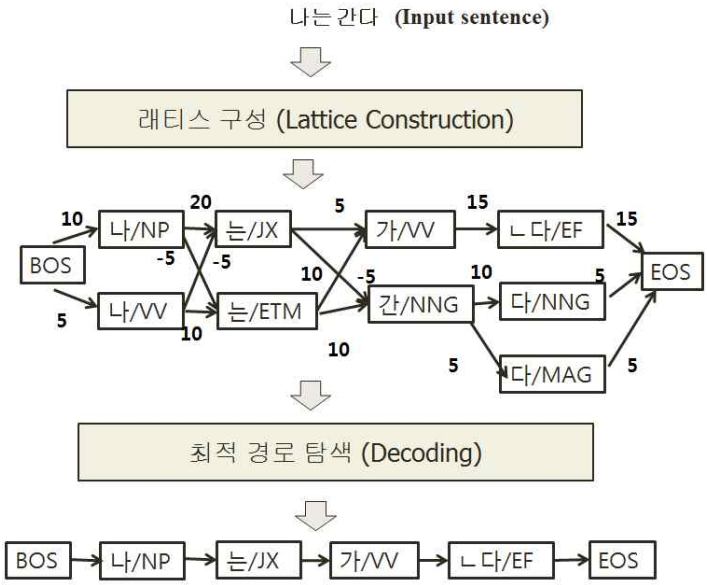


그림 1. 제안 방법의 흐름도

경로 $\boldsymbol{y}$의 점수는 $s(\boldsymbol{y})$로 표기하며, 다음과 같이 구성 에지의 점수의 합으로 정의된다.

$$s(\boldsymbol{y}) = \sum_{(i,j)\in \boldsymbol{y}} s(i,j) = \sum_{(i,j)\in \boldsymbol{y}} \boldsymbol{w} \cdot \boldsymbol{f}(i,j) \qquad (1)$$

식 (1)에서, $\boldsymbol{f}(i,j)$는 래티스 G의 노드 i,j에 대한 자질 벡터 (feature vector)를, $\boldsymbol{w}$는 자질 가중치 벡터 (feature weight vector)이다. 입력문 $\boldsymbol{x}$에 대해 시작노드 0과 마지막 노드 1을 연결하는 모든 가능한 경로 집합을 $P(\boldsymbol{x})$라고 하자. 형태소 분석 문제는 $P(\boldsymbol{x})$중에서 점수가 가장 높은 경로를 찾는 것으로, 다음의 최적화 문제를 푸는 것으로 귀결된다.

$$\hat{\boldsymbol{y}} = \arg max_{\boldsymbol{y}\in P(\boldsymbol{x})} (s(\boldsymbol{y})) \qquad (2)$$

그림 1은 입력문 "나는 간다"에 대한 래티스의 예를 보여준다. 그림에서, BOS와 EOS는 각각 시작 노드 0와 끝 노드 1를 가리킨다.

2차 자질 사용

기존의 그래프 기반 의존 파싱 연구와 유사하게, 앞의 형식화는 2차 (second-order)자질을 고려하여 확장될 수 있다. 1차 (first-oder) 자질은 자질 벡터를 구성할 때 참조하는 에지가 1개인 경우를 말하며, 2차 자질은 참조하는 에지들이 연속된 두 개의 인접한 에지인 경우를 일컫는다. 결국, 2차 자질은 연속된 세 개의 형태소 노드로부터 추출된다. 2차 자질을 사용하여 식 (1)을 확장하면 다음과 같다.

$$s(\boldsymbol{y}) = \sum_{(i,j,k)\in \boldsymbol{y}} s(i,j,k) \qquad (3)$$

여기서, $s(i,j,k)$로 다음과 같이 형식화 된다

$$s(i,j,k) = \boldsymbol{w} \cdot \boldsymbol{f}(i,j,k) \quad (4)$$

$\boldsymbol{f}(i,j,k)$는 노드 i,j,k에서 추출된 2차 자질 벡터이다. [5]의 그래프 의존 파싱에서와 같이, 최종적으로 $s(i,j,k)$는 1차 자질의 점수와 통합되어 사용된다.

$$s'(i,j,k) = s(j,k) + s(i,j,k) \quad (5)$$

3.2 디코딩 알고리즘

식 (2)의 최적화 문제는 에지의 점수 $s(i,j)$를 $-s(i,j)$로 간주할 때, 그래프 상의 최단 경로를 찾는 문제로 Viterbi 알고리즘을 이용하여 풀 수 있다. 그림 2은 2차 자질을 사용할 때의 최적 경로를 찾는 Viterbi 알고리즘의 pseudo code을 보여준다.

```
T = { 0 }
while( T ≠ ∅ ){
k = T.pop()
visited[k] = 1
for each j∈prevs(k)
    S[j][k] = max_{i∈prevs(j)}(S[i][j] + s(i,j,k) + s(j,k))
  a[i][k] = argmax_{i∈prevs(j)}(S[i][j] + s(i,j,k) + s(j,k))
}

for each j∈nexts(k)
    if prevs(j)의 모든 노드가 이전에 방문되었으면,
        T.add(j)
}
```

그림 2. 래티스상 최적 경로를 찾는 Viterbi 알고리즘 (2차 자질을 사용하도록 확장된 버전)

3.3 자질 가중치 벡터 학습

자질 가중치 벡터 $\boldsymbol{w}$를 학습하기 위해서, 본 논문은 [1]의 *averaged perceptron*을 사용한다. 그림 3은 자질 가중치 벡터를 학습하기 위한 averaged perceptron알고리즘의 pseudo code을 보여준다.

```
Training data: T = {(x_t, y_t)}_{t=1}^{M}
w_0 = 0; v = 0; i = 0
for n : 1 ··· N
    for t : 1 ··· M
        ŷ_t = argmax_{y∈P(x_t)}(s(y))
        w_{i+1} = F(x_t, y_t) − F(x_t, ŷ_t)
                (where F(x,y) = Σ_{(i,j)∈y} f(i,j))
        v = v + w^{(i+1)}
        i = i + 1
w = v/(N • M)
```

그림 3. Averaged perceptron에 기반한 자질 가중치 벡터 학습 알고리즘

3.4 래티스 구성 (Lattice Construction)

래티스를 구성할 때, 일본어나 중국어와 달리, 한국어에서는 활용/변이로 인해 표층형과 사전형이 달라지는 문제를 고려해야 한다. 이른바 *표층형과 사전형 간의 불일치* 문제를 해결하기 위해, 본 논문에서는 별도로 음절 매핑 테이블 (syllable mapping table)을 참조하여 래티스를 구성한다[1]. 아래, 음절 매핑 테이블의 예들을 보여준다.

표층형	사전형
했	하았
게	거이, 것이
내	나의, 나어
왔	오았
와	오아

한국어에서 래티스 구성은 다중 입력문 (multiple input sentences)으로부터 각 입력문의 부분음절열에 대해 사전 참조하여 래티스를 구성하는 방식과 동일하다. 다시 말해, 주어진 음절열에 대해, 입력문내 각 음절마다 음절 매핑 테이블을 참조하여 사전형으로 치환하여 새로운 입력문을 만들어 낸다. 예를 들어, "처리했다"의 경우, "했"을 사전형 "하았"으로 치환하여 "처리하았다"라는 새로운 입력문을 파생시키는 것이다. 결국 최종 래티스는 가능한 모든 변이 입력문을 모아서 N개의 다중 입력문을 구성한 후에 각각의 입력문에 대해서 래티스를 생성 한 후, 이렇게 얻어진 N개의 래티스를 병합하여 얻어진다. 다중 입력문의 수 N는 매핑 테이블을 참조하여 음절 변이가 일어나는 음절의 갯수에 따라 지수적으로 증가하는데 (exponentially increasing), 이를 효율적으로 처리하기 위해 dynamic programming기법을 사용한다.

1) 음절 매핑 테이블은 학습코퍼스로부터 대부분 자동으로 획득된다.

CRF기반 미등록어 자동 추출

래티스 생성 단계는 어휘 사전에 등재된 형태소만을 대상으로 하기 때문에, 미등록어 문제를 해결하지 못한다. 미등록어 처리를 위해, 본 논문에서는 [11]의 linear-chain CRF를 입력문에 적용하여 얻어진 1-best 형태소 분석 결과를 어휘 사전에 추가하였다.

3.5 자질 벡터

본 논문에서 사용하는 자질은 1차 자질과 2차 자질으로 나뉜다. 사용되는 정보는 형태소 노드의 표층형 (surface form), 품사 (POS tag), 해당 표층형의 시작 음절 앞 또는 끝 음절 다음에 띄어쓰기가 있었는지 여부 (spacing information), 형태소 노드의 시작과 끝 음절이 변이를 통해 얻어졌는지 여부 등이다.

사용하는 자질 집합을 보다 구체적으로 기술하기 위해, 먼저, 노드 i에 대한 기본 자질 (basic features)을 다음과 같이 정의하도록 한다.

기본자질이름	정의
lspace(i)	현재 형태소 시작전에 띄어쓰기가 있는지 여부
rspace(i)	현재 형태소 다음에 바로 띄어쓰기가 있는지 여부
lopen(i)	현재 형태소의 시작 음절이, 음절 매핑테이블을 참조하여 변이되었는지 여부
ropen(i)	현재 형태소 마지막 음절이 음절매핑테이블을 참조하여 변이되었는지 여부
surface(i)	현재 형태소의 표층 문자열
tag(i)	형태소 i의 품사 태그
length(i)	형태소 i의 표층형 길이
surfaceext(i)	lopen(i), ropen(i)을 고려한 표층 문자열: surface(i), lopen(i), ropen(i)의 세 정보의 조합으로 구성된다.

여기서, lopen, ropen의 자질 값의 예를 들기 위해, “갔”으로부터 “갔=>가았”의 음절 매핑테이블을 참조하여, “가/VV”, “았/EP”의 두 가지 형태소 노드가 만들어졌다고 가정하자. 이렇게 얻어진 두 형태소 노드에 대한 lopen, ropen정보는 다음과 같다.

	lopen	ropen
가/VV	false	true
았/EP	true	false

반면, 주어진 형태소의 시작과 끝 음절이 모두 변이가 없이 얻어진 경우에는 lopen, ropen값이 모두 false가 된다.

1차 자질과 2차 자질은 위에서 정의된 기본 자질로부터 파생되는데, 간편한 기술을 위해, 다음과 같은 표기를 정의하도록 하자.

표기	내용
Pt	tag(i)
Ct	tag(j)
Nt	tag(k)
Plen	length(i)+lopen(i)+ropen(i)
Clen	length(j)+lopen(j)+ropen(j)
Pr	ropen(i)
Cr	ropen(j)
Pls	lspace(i)
Cls	lspace(j)
Pw	surface(i)
Cw	surface(j)
Nw	surface(k)
Pw2	surfaceext(i)
Cw2	surfaceext(j)
Nw2	surfaceext(k)

여기서 “+”는 여러개의 자질들의 조합 연산자이다.

최종적으로, 1차 자질과 2차 자질은 다음과 같다.

1) 1차 자질 $\boldsymbol{f}(i,j)$:

- CtClen, CtPlenClen, PtCtPlenClen, PtCtPr, CtPr, CtCr, CtCl, CtCr, CtClsCl, CtCrsCr, CtCw, CtCw2, PwCw, Pw2Cw, PtCtCw, PtCtCw2, PtCtCw, Pw2CtCw2, PtPwCw, Pw2CtCw2, PtPwCw, PtPw2Cw2, PtPwCt, PtPw2Ct, PtPwCtCw, PtPw2CtCtw2
- window기반 지역 문맥 정보 (local context): 각 형태소의 시작과 끝 음절을 기준으로 앞뒤 3음절까지의 부분음절열을 지역 정보로 사용한다.
- penality: 경로상 하나의 에지를 취할 때 드는 패널티로, 최장일치휴리스틱처럼, 경로의 노드의 수를 가능한한 적게 (또는 많게)하기 위해서 사용한다.

2) 2차 자질 $\boldsymbol{f}(i,j,k)$

- PtCwNt, PwCtNt, PtCtNw, PtCtNtNw, PtCtNtCwNw, PtCtNtPwCwNw, PtCtNtCw, PtCtNtPw, PtCtNtPwCw

1차 및 2차 자질 모두 자질명 (feature name)과 자질값 (feature value)으로 구성되며, penalty만 제외하고 나머지는 모두 이진 값(binary value)를 자질값으로 취한다.

4. 실험 결과

실험을 위해 ETRI품사 부착 대화체 말뭉치 약 10만 문장을 사용한다. 이중 90%를 학습용으로 나머지 10%는 테스트용으로 사용하였다. 참조하는 어휘 사전의 크기는 총 약 380만 엔트리로, 대부분 복합명사나 전문용어로 구성되었다. 최종 어휘 사전은 원래의 380만 어휘 사전 외에 학습데이터에 나타나는 어휘까지 추가하여 구성하였다 (대부분은 사전에 포함된다). 제안 방법을 포함 본 실험에서 평가 대상으로 하는 형태소 분석 방법들은 다음과 같다.

- **베이스라인 시스템 (CRF)**: [11, 25]의 linear-chain CRF에 기반한 방법을 확장한 것으로, [25]과 유사한 음절 태깅에 기반하여 형태소 분할과 태깅을 동시에 수행하는 결합 모델을 사용하였다. 복합형태소를 단위 형태소로 분해하기 위해 [12]의 방법을 사용한다. [12]에서는 복합형태소에 대한 기분석과 기능형태소에 대한 기분석의 두 가지 기분석 패턴을 사용하는데. 먼저, 복합형태소 전체에 대한 기분석을 적용하여 분해를 시도하고, 여기에 나타나지 않는 복합형태소에 대해서는 기능형태소에 대한 기분석에 참조하여 분해를 시도한다.
- **래티스 기반 구조화 분류 (Lattice)**: 제안하는 래티스 기반 구조화 분류를 적용한 방법으로, 별도의 미등록어 추출없이 원래의 어휘 사전만을 이용한 결과이다. 사용 자질 유형에 따라 두 가지 모델이 얻어지는데, **Lattice**는 first-order자질만 사용한 것을, **Lattice(2nd order)**는 second-order자질까지 함께 사용한 모델을 가리킨다.
- **래티스 기반 구조화 분류 + 미등록어 자동 추출 (Lattice+unkProc)**: 앞에서 설명한 제안 래티스 기반 구조화 분류 방법에 더하여, 3.4절에서 CRF기반 미등록어 자동 추출까지 적용한 방식을 가리킨다.

표 1. 제안 lattice기반 방법의 성능

방법	F-measure	EA	SA
CRF	0.9086	0.8649	0.5543
Lattice	0.9262	0.8886	0.6322
Lattice(2nd order)	0.9393	0.9103	0.6920
Lattice(2nd order)+ unkProc	**0.9407**	**0.9127**	**0.6938**

표 1은 네 가지 방법을 비교한 결과이다. F-measure는 형태소 단위의 F-measure를, EA (eojeol accuracy)는 어절 정확률을, SA는 문장 정확률 (sentence accuracy)을 가리킨다. 표 1에 보다시피, 제안 래티스 기반 방법은 CRF기반 방법대비 세가지 지표에서 모두 높은 성능을 보여주었다. 또한, 1차 자질에 더해 2차 자질까지 확장할 때, 성능이 더욱 향상됨을 알 수 있다. 미등록어 자동 추출을 적용하여 어휘 사전을 확장한 경우, 최종 성능에서 증가가 있었으나 그 차이는 크지 않았다. 이는 어휘 사전이 이미 충분히 있어, 미등록어 현상이 거의 발생하지 않았기 때문이다.

기존 연구 [11]과 비교하여, 어절 정확률이 낮은 이유는 상당부분 대부분 복합명사 분해 애매성으로부터 기인하는 것으로 보인다. 다시 말해, 복합 명사의 경우 형태소 단위가 코퍼스 구축자의 판단에 의존하는데, 분석 결과는 다른 관점에서 보면 올바르나, 정답은 한가지 경우만 기술되어 있어 나머지들이 모두 오류로 잡혀 정확률을 감소시킨 것이다. 향후, 복합명사 분해에 대해 보다 완화된 평가 방식을 고안하여, 더욱 정교하게 성능을 비교할 필요가 있다.

5. 결론

본 논문은 래티스 기반의 구조적 분류에 기반한 한국어 형태소 분석 및 품사 태깅 방법을 제안하였다. 실험결과, ETRI 품사 부착 말뭉치에 대해, 제안 래티스 기반 방법은 기존의 linear-chain CRF기반 방법에 비해 우수한 성능을 보여주었다. 물론, 본 실험에서 사용된 linear-chain CRF은 [11]의 특정 1차 자질 셋에 기반을 둔 것이므로, 래티스 기반 방법이 모델적으로 (일반적으로) linear-chain CRF보다 우수하다고 결론을 이끌어낼 수는 없다. 더욱 객관적인 결론 도출을 위해서는 보다 다양한 자질 집합에서 엄밀한 실험을 수행하여야 할 것이다.

향후, Sejong코퍼스에서 본 논문에서 제안 방법의 실험을 확장하여, 코퍼스가 대규모인 경우에 어떠한 차이를 보이는지 비교해 볼 계획이다. 또한, 래티스 구성 시, 형태/음운론적 제약과 접속 정보 등을 사용하여 제안 방법에 대한 상세한 실험을 수행할 것이다. 그리고, 미등록어 자동 추출시 CRF결과의 1-best결과 외에 n-best결과를 함께 이용하는 방법도 향후 연구 주제로서 흥미로울 것이다. 마지막으로, 본 논문에서는 자질 가중치 벡터를 학습하기 위해 averaged perceptron을 이용하였으나, SVMStruct등의 다른 학습 방법도 활용하여 비교하는 것도 흥미로운 연구 주제가 될 것이다.

참고문헌

[1] Collins, M. (2002). Discriminative Traning Methods for Hidden Markov Models: Theory and Experiments with Perceptron Algorithms. EMNLP, 1-8.

[2] Kudo, T., Yamamoto, K., & Matsumoto, Y. (2004). Applying Conditional Random Fields to Japanese Morphological Analysis. EMNLP, 230-237.

[3] Lee, D., & Rim, H. (2005). Probabilistic Models for Korean Morphological Analysis. IJCNLP (pp. 197-202).

[4] Lee, G. G., Cha, J., & Lee, J. (2002). Syllable-Pattern-Based Unknown- Morpheme Segmentation and Estimation for Hybrid Part-of-Speech Tagging of Korean. Computational Linguistics, 28(1).

[5] Mcdonald, R., Grammer, K., & Pereira, F. (2005). Online Large-Margin Training of Dependency Parsers. ACL, 91-98.

[6] Shim, K., & Yang, J. (2002). MACH - A Supersonic Korean Morphological Analyzer. COLING. 939-945.

[7] 강승식. (1994). 다층 형태론과 한국어 형태소 분석 모델. 한글 및 한국어정보처리 학술대회 (HCLT) 140-145.

[8] 강승식, & 장병탁. (1996). 음절 특성을 이용한 범용 한국어 형태소 분석기 및 맞춤법 검사기. 정보과학회

논문지, 23(5).
[9] 강승식. (2002). “한국어 형태소 분석과 정보 검색“, 홍릉과학출판사.
[10] 권오욱, 정유진, 김미영, 류동원, 이문기, & 이종혁. (1999). 음절단위 CYK알고리즘에 기반한 형태소 분석기 및 품사태거. MATEC ‘99 대회.
[11] 나승훈, 양성일, 김창현, 권오욱, 김영길. (2012). CRF에 기반한 한국어 형태소 분할 및 품사 태깅. 한글 및 한국어 정보처리 학술대회.
[12] 나승훈, 김창현, 김영길. (2013). CRF기반 한국어 형태소 분할 및 품사 태깅에서 두 단계 복합형태소 분해 방법. 한글 및 한국어 정보처리 학술대회.
[13] 박철제, 이종혁, & 이근배. (1997). 확률 접속표를 이용한 일본어 형태소 분석 결과의 후처리 기법. 정보과학회논문지, 24(1).
[14] 신준철, & 옥철영. (2012). 기분석 부분 어절 사전을 활용한 한국어 형태소 분석기. 한국정보과학회 논문지, 39(5), 415-424.
[15] 심광섭. (2013). 음절단위의 한국어 품사 태깅에서 원형 복원. 정보과학회논문지: 소프트웨어 및 응용, 40(3), 182-189.
[16] 심광섭. (2011). 형태소 분석기 사용을 배제한 음절 단위의 한국어 품사 태깅. 인지과학, 22(3), 327-345.
[17] 심광섭. (2011). CRF 를 이용한 한국어 자동 띄어쓰기. 인지과학, 22(2), 217-233.
[18] 심광섭, & 양재형. (2004). 인접 조건 검사에 의한 초고속 한국어 형태소 분석. 정보과학회논문지, 31(1), 89-99.
[19] 양성일, 홍문표, 김영길, 최승권. (2003). 띄어쓰기 정보를 이용한 한국어 복합 형태소 분석. 한국정보과학회 학술발표논문집, 612-616.
[20] 여상화, 김용호, 이학주, & 이정현. (1991). 다단계 필터링 능력을 갖는 형태소 분석기의 설계 및 구현. 한국정보과학회 가을 학술발표 논문집.
[21] 오진영, & 차정원. (2009). 엔트로피 지도 CRF를 이용한 한국어 어절 구문태그 예측. 한국정보과학회 논문지, 15(5), 395-399.
[22] 이성진, 김덕봉, 서정연, 최기선, & 김길창. (1992). Two-level 모델을 이용한 한국어 용언의 형태소 해석. 한국정보과학회 가을 학술발표논문집.
[23] 이은철, & 이종혁. (1992). 계층적 기호 접속정보를 이용한 한국어 형태소 분석기의 구현. 한글 및 한국어 정보처리 학술발표 논문집 (HCLT).
[24] 이재성. (2011). 한국어 형태소 분석을 위한 3단계 확률 모델. 정보과학회논문지, 38(5).
[25] 이창기. (2013). Structural SVM을 이용한 한국어 띄어쓰기 및 품사 태깅. 한국컴퓨터종합학술대회, 604-606.
[26] 최재혁, & 이상조. (1993). 양방향 최장일치법에 의한 한국어 형태소 분석기에서의 사전 검색 횟수 감소 방안. 한국정보과학회 논문지, 20(10).
[27] 홍진표, & 차정원. (2008). 어절패턴 사전을 이용한 새로운 한국어 형태소 분석기. 한국컴퓨터종합학술대회 논문집 (Vol. 35).

Semi-CRF or Linear-Chain CRF? 한국어 형태소 분할 및 품사 태깅을 위한 결합 모델 비교

나승훈°, 김창현, 김영길
한국전자통신연구원
nash@etri.re.kr, chkim@etri.re.kr, kimyk@etri.re.kr

Semi-CRF or Linear-chain CRF? A Comparative Study of Joint Models for Korean Morphological Analysis and POS Tagging

Seung-Hoon Na°, Chang-Hyun Kim, Young-Kil Kim
Natural Language Processing Laboratory
Electronics and Telecommunication Research Institute

요 약

본 논문에서는 한국어 형태소 분할 및 품사 태깅 방법을 위한 결합 모델로 Semi-CRF와 Linear-chain CRF에 대한 초기 비교 실험을 수행한다. Linear-chain방법은 출력 레이블을 형태소 분할 정보와 품사 태그를 조합함으로써 결합을 시도하는 방식이고, Semi-CRF는 출력의 구조가 분할과 태깅 정보를 동시에 포함하도록 표현함으로써, 디코딩 과정에서 분할과 태깅을 동시에 수행하는 방법이다. Sejong품사 부착 말뭉치에서 비교결과 Linear-chain방법이 Semi-CRF방법보다 우수한 성능을 보여주었다.

주제어: 성능 비교, Semi-CRF, Linear-chain CRF, 형태소 분석, 품사 태깅

1. 서론

한국어 형태소 분석을 위한 통계기반 방법은 대규모 품사 부착 말뭉치로부터 형태소 분할 및 품사 태깅에 필요한 규칙 및 확률 모델을 자동 또는 반자동으로 학습하는 방식으로 [3,4,10,12,13,14,15,16,17], 수작업이 거의 필요 없고, 성능이 우수하며, 타 도메인으로의 적용성이 높고, 기존의 어휘 사전과의 하이브리드가 가능하다는 점 등의 장점으로 인해, 현대의 대부분의 품사 태깅 연구가 이에 기반을 두고 있다.

통계 기반 방법 중, 최근 들어 입력문 전체에 대해 CRF나 structured SVM등과 같은 기계 학습 방법을 한국어 형태소 분석에 적용하는 연구들이 제안되었다 [10, 14,15,16]. 이들 연구 중, [10]에서는 한국어 분석을 중국어의 단어 분할 및 품사 태깅 방법을 그대로 적용할 수 있는 방법을 제안했다. 이 방법은 실험 결과 SEJONG코퍼스에서 높은 효과성을 보였으며, [16]에서는 음절 기반 결합 모델로 성능이 더욱 향상되었다.

본 연구에서는 형태소분할과 품사 태깅을 동시 수행하는 결합 모델로 linear-chain CRF [2]와 Semi-CRF [5]을 성능 평가를 수행하여 경험적으로 비교한다. 먼저, linear-chain CRF방식은 [16]의 음절 기반 결합 모델로, 음절의 태깅 출력 레이블을 형태소 분할 정보와 품사태그를 조합하여 구성함으로써, 결합을 수행하는 방식이다. Semi-CRF는 디코딩 과정에서 분할과 태깅을 동시에 수행할 수 있는 방법이다. Sejong품사 부착 말뭉치에서 비교 결과, 기본 자질 집합에서, linear-chain CRF가 Semi-CRF보다 더욱 높은 성능을 보여주었다.

2. 관련 연구

최근의 한국어 형태소 분석의 연구는 입력문 전체에 언어학적 형태소 분석이 없이 통계기반 모델을 수행하는 방식으로 연구가 진행되고 있다. 이들 연구는 표층형과 사전형 간의 불일치 문제를 해소하는 방법에 따라 다음의 두 가지 접근법으로 구분할 수 있다.

먼저, 표층형과 사전형 불일치를 *복합형태소* (compound morpheme)의 도입을 통해 해결하는 방식이다 [10,16]. [10,16]에서는 순차 태깅을 통해, 형태소 분할과 태깅을 수행, 입력문을 단위형태소 또는 복합형태소로 분해한다. 이후, 복합형태소는 후처리 단계에서 기분석 패턴을 참조하여 단위형태소로 상세 분해한다 [10,15]. 기분석 패턴에 나타나지 않은 미등록 복합형태소를 위해 [10]은 이에 더 나아가 래티스 (lattice)기반 HMM방식을 제안하였다.

다른 접근법은 [13,14]의 음절 단위의 접근법이다. 이 방법은 [12]과 기본적으로 동일하나, 표층형과 사전형간의 불일치를 해소하기 위해 복합형태소를 도입하는 대신, 복합 음절을 도입한다. 복합 음절은, 두 가지 서로 다른 태그의 부분음절들이 표층화 단계에서 한음절로 표현된 것으로, [13,14]은 복합 음절을 분해하기 위해, 별도의 음절 복원 기분석 사전에 기반한다. 복합 음절의 경우에는 음절 단위로 처리하므로, 미등록 복합 음절이 거의 발생하지 않는다는 장점이 있어, 별도의 미등록 복합 음절 처리 과정까지는 제안되지는 않았다.

3. 형태소 분할 및 태깅을 위한 결합 방법

본 논문에서 비교하고자하는 형태소 분할 및 품사 태깅의 기본 절차는 [10]의 다음 3단계로 이루어진 형태소

분석 과정이다.

1. 형태소 분할:
   예) <나, 는, 학교, 에, 갔, 다>
2. 품사 태깅:
   예) <나/NP, 는/JX, 학교/NNG, 에/JKB, 갔/VV~EP, 다/EF>
3. 복합형태소 분해:
   예) <나/NP, 는/JX, 학교/NNG, 에/JKB, 가/VV, 았/EP, 다/EF>

위 3단계 절차중 본 논문에서 결합하고자 하는 과정은 형태소 분할과 품사 태깅의 두 단계이다. 다음 절에 linear-chain CRF 및 Semi-CRF에 기반한 두 결합 방법을 기술한다.

3.1 Linear-chain CRF

Linear-chain CRF [2]에 기반한 결합 모델은 음절 기반 태깅으로, [16]와 유사하게, 분할과 품사 태그 정보를 조합하여 태깅의 출력 레이블로 사용한다. 예를 들어, 입력문 "학교에갔다"에 대해 출력 레이블<B-NNG,I-NNG, B-JKB, B-VV~EP, B-EF>와 같이 출력 결과를 구성하는 것이다. 레이블에서 B는 해당 음절에서 새로운 형태소가 시작한다는 것을, I는 해당 음절이 이전 음절의 형태소 단위에 포함된다는 것을 의미한다. {B,I} 태그의 의미에 따라 해석하면 다음의 분할/태깅 열을 얻는다.

<학교/NNG, 에/JKB, 갔/VV~EP, 다/EF>

여기서 "갔"은 복합형태소로, VV~EP는 *복합태그*가 된다, 복합태그는 [10]의 정의에 따르는데, VV~EP는 복합형태소의 시작형태소의 태그가 VV이고 마지막형태소의 태그가 EP를 취한다는 뜻이다. 복합형태소 분해 단계는 linear-chain CRF 및 Semi-CRF 공통으로 사용하는 것으로 3.4절에서 자세히 설명한다.

3.2 Semi-CRF

Linear-chain CRF에서는 품사 태그에 {B, I}의 분할 태그를 덧붙여서 음절의 레이블 집합을 구성함으로써, 분할과 태깅을 동시에 수행한다. 그러나, Semi-CRF [5]는 구조적 분류를 통해 순차열의 출력 구조를 분할과 태깅 정보를 동시에 내포하도록 구성함으로써, 디코딩 과정에서 동시에 분할과 태깅을 수행할 수 있도록 하는 방법이다. 형식적으로 기술하기 위해, 먼저 일반적인 구조적 분류 문제 (structured output problem)를 살펴보자. 주어진 입력문 $\boldsymbol{x}$에 대해 분할/태깅 결과가 $\boldsymbol{y}$라고 하면, 구조적 분류 문제는 다음과 같이 형식화 된다.

$$\boldsymbol{y}^* = \arg max_{\boldsymbol{y}}(\boldsymbol{w} \cdot \boldsymbol{\phi}(\boldsymbol{x},\boldsymbol{y})) \qquad (1)$$

Semi-CRF [5]는 구조적 분류 문제의 특수한 예인데, 형식적으로, 입력 $\boldsymbol{x}$는 순차열로 $x_1 \cdots x_n$로 기술되고. 이때 x_i는 주어진 입력의 i번째 단어 (또는 음절)을 n은 음절열의 길이로 $|\boldsymbol{x}|$로도 표현된다. $\boldsymbol{y}$는 세그먼트열로 $y_1 \cdots y_m$으로 기술되는데, 각 $y_{i=}(s_i,e_i,t_i)$로 s_i는 i번째 세그먼트의 시작 위치 (start position)를, e_i는 끝 위치 (end position)를, t_i는 레이블 (label)을 가리키며, m은 세그먼트열의 길이로 $|\boldsymbol{y}|$로도 표기된다.

예를 들어, 입력문 "학교에갔다"에 대한 세그먼트 결과 $\boldsymbol{y}$ = <(1,2,NNG), (3,3,JKB), (4,4,VV~EP), (5,5,EF)>는 linear-chain CRF의 출력 레이블 <B-NNG,I-NNG,B-JKB,B-VV~EP, B-EF>에 대응이 된다.

3.3 Semi-CRF와 linear-chain CRF의 관계

Semi-CRF와 linear-chain CRF는 다음의 관계를 갖는다.

1. Semi-CRF는 linear-chain CRF를 특수한 경우로 포함한다.

형식적으로 Semi-CRF는 linear-chain CRF의 일반화된 형태로, 각 세그먼트 y_i의 최대 길이를 1로 제한하면 Semi-CRf는 linear-chain CRF와 형식적으로 등가가 된다.

2. 분할/태깅 결합 모델의 1차 Linear-chain CRF에 대해, 디코딩 결과가 등가가 되는 Semi-CRF가 존재한다.

여기서 linear-chain CRF는 {B,I}에 기반한 분할/태깅의 결합 모델로 가정하고, 자질벡터를 다음 두 가지로 분류하자.

1. $\boldsymbol{\phi_i}(t,\boldsymbol{x})$: 위치 i의 레이블 t와 입력문 $\boldsymbol{x}$의 함수로 표현되는 모든 자질 벡터의 결합된 형태. 이에 대응되는 자질 벡터는 $\boldsymbol{w_i}(t,\boldsymbol{x})$로 기술한다.
2. $\phi(t,t')$: 레이블 t에서 t'로의 전이 자질로 scalar이다. 이에 대응되는 자질 가중치는 $w(t,t')$로 기술한다.

위의 자질을 사용하는 linear-chain CRF와 등가가 되는 Semi-CRF은 다음의 자질을 사용한다.

1. $\boldsymbol{g}(i,j,t,\boldsymbol{x})$: 시작 위치 i와 끝 위치 j에 대응되는 품사 태그 t에 대한 자질 벡터로, $[\boldsymbol{\phi_i}(B-t,\boldsymbol{x}) \cdots \boldsymbol{\phi_j}(I-t,\boldsymbol{x})]$로 표현된다 (여기서 ;은 여러 개의 벡터를 하나의 벡터로 결합하는 연산자이다). 여기서, $B-t$는 태그 t의 시작 음절을, $I-t$는 태그 t의 중간 또는 끝 음절을 의미하는 음절 단위의 레이블이다. 대응되는 자질 가중치 벡터는 $\boldsymbol{h}(i,j,t,\boldsymbol{x}) = [\boldsymbol{w_i}(B-t,\boldsymbol{x}) \cdots \boldsymbol{w_j}(I-t,\boldsymbol{x})]$로 정의한다.

2. $g(i,j,t,i',j',t')$: 세그먼트 (i,j,t)와 (i',j',t')의 전이 자질 벡터로, $|j-i+1|=1$이면 $\phi(B-t,B-t')$을, 그렇지 않으면 $\phi(I-t,B-t')$로 정의된다. 이에 대한 자질 가중치 벡터는 $h(i,j,t,i',j',t')$로 $|j-i+1|=1$이면 $w(B-t,B-t')$ 그렇지 않으면 $w(I-t,B-t')$ 가 된다.

위의 linear-chain CRF와 semi-CRF는 분할/품사 태깅 결과가 등가임을 쉽게 보일 수 있다. 즉, 이론적으로, 분할/태깅 문제에서 1차 semi-CRF는 1차 linear-chain CRF를 특수한 자질을 사용한 경우로 포함하고 있음을 알 수 있다.

3.4 복합 형태소 분해

Semi-CRF와 Linear-chain CRF의 복합형태소 분할은 [12]의 방법을 따른다. [12]에서는 복합형태소 처리는 기분석에 패턴에 기반을 두며, 기분석 패턴은 두 종류로, 복합형태소에 대한 기분석과 *복합기능형태소* (compound functional morpheme)으로 나뉜다. 예를 들면, 복합형태소의 입력으로 "살려줬/VV~EP "이 주어졌다고 하자. 먼저, "살려줬/VV~EP"이 복합형태소 기분석 패턴에 있는지를 파악한다. 만약, 복합형태소 기분석 패턴에 없다면, 다음 두 단계로 나뉘어 분할이 된다. 먼저, 첫 번째 단계에서는 "살리/VV"를 내용형태소로 분해하고, "어줬/EP"를 복합기능형태소 분해한다. 이때, EP는 복합기능형태소에 대한 복합태그로서, 가장 마지막 단위 형태소의 태그로 지정하였다. 두 번째 단계에서는, 복합기능형태소 "어줬/EP"에 대해서 기분석패턴에 기반하여 분해를 시도한다. 기분석을 참조하여, "어줬/EP: 어주/VX+었/EP"를 후보로 얻어내어, 최종적으로 해당 복합형태소에 대해 "살리/VV+어주/VX+었/EP"의 분해 결과를 얻게 된다. 위의 두 과정을 요약하면 다음과 같다.

살려줬/VV~EP ➔ (내용형태소와 기능형태소 분해)
살리/VV+어줬/EP ➔ (기능형태소 상세 분해)
살리/VV+어주/VX+었/EP

3.5 자질 벡터

앞에서 설명한 두 가지 결합 방법에 대한 자질 벡터를 정리하면 다음과 같다.

Linear-chain CRF에서 자질 벡터

[10]의 형태소 분할을 위한 자질벡터와 동일하며, 다음과 같이 기술된다.

표 1. Linear-chain CRF에서 사용하는 자질 유형

Feature symbol	Desciption
$C_{-2}, C_{-1}, C_0, C_1, C_2$	1음절 (uni-char) 정보
$C_{-1}C_0, C_0C_1, C_1C_2$	2음절 (bi-char) 정보
$C_{-2}C_{-1}C_0, C_{-1}C_0C_1,$ $C_0C_1C_2$	3음절 (tri-char) 정보
$S_{-2}, S_{-1}, S_0, S_1, S_2$	1음절 띄어쓰기 정보
$S_{-1}S_0, S_0S_1, S_1S_2$	2음절 띄어쓰기 정보
$S_{-2}S_{-1}S_0, S_{-1}S_0S_1,$ $S_0S_1S_2$	3음절 띄어쓰기 정보

Semi-CRF에서 자질 벡터

Semi-CRF에서 사용하는 자질 표기를 위해, C_i를 i번째 위치에서 표층음절을, C_i^j는 $C_i \cdots C_j$를, S_i를 i번째 위치 바로 이전에 띄어쓰기가 있었는지 여부를, S_i^j는 $S_i \cdots S_j$를 가리키는 표기라고 하자. 세그먼트 (i,j,t)에 대해서 Semi-CRF에서 사용하는 자질은 다음과 같다.

표 2. Semi-CRF에서 사용하는 자질 유형

Feature symbol	Desciption
$C_i \cdots C_j$	세그먼트에 대한 표층정보
$C_{i-2}C_{i-1}C_i^j, C_{i-1}C_i^j,$ $C_i^jC_{j+1}, C_i^jC_{j+1}C_{j+2}$	세그먼트와 주변 1-2음절에 대한 공기정보
$S_i \cdots S_j$	세그먼트에 대한 띄어쓰기 정보
$C_{i-2}, C_{i-1}, C_i,$ C_j, C_{j+1}, C_{j+2}	1음절 (uni-char) 정보
$C_{i-2}C_{i-1}, C_{i-1}C_i, C_iC_{i+1},$ $C_{j-1}C_j, C_jC_{j+1}, C_{j+1}C_{j+2}$	2음절 (bi-char) 정보
$C_{i-2}C_{i-1}C_i, C_{i-1}C_iC_{i+1},$ $C_iC_{i+1}C_{i+2}, C_{j-2}C_{j-1}C_j,$ $C_{j-1}C_jC_{j+1}, C_jC_{j+1}C_{j+2}$	3음절 (tri-char) 정보
$S_{i-2}, S_{i-1}, S_i, S_j, S_{j+1}, S_{j+2}$	1음절 띄어쓰기 정보
$S_{i-2}S_{i-1}, S_{i-1}S_i, S_iS_{i+1},$ $S_{j-1}S_j, S_jS_{j+1}, S_{j+1}S_{j+2}$	2음절 띄어쓰기 정보
$S_{i-2}S_{i-1}S_i, S_{i-1}S_iS_{i+1},$ $S_iS_{i+1}S_{i+2}, S_{j-2}S_{j-1}S_j,$ $S_{j-1}S_jS_{j+1}, S_jS_{j+1}S_{j+2}$	3음절 띄어쓰기 정보

4. 실험 결과

두 가지 결합 모델의 형태소 분할 및 품사 태깅 성능 평가를 위해, 세종 품사 부착 말뭉치를 이용하였는데 이는 총 253,884개의 문장, 1,008,925개의 어절로 구성된다. 성능 평가 측도로 형태소 단위의 F-measure, 어절 정확률 (EA)를 사용하였으며, 이중 80%를 학습데이터로 20%는 테스트데이터로 사용했다. Linear-chain CRF와 Semi-CRF의 가질 가중치 벡터는 모두 [1]의 Averaged perceptron을 사용하여 학습하였다. 전체 학습데이터에서 최대 반복횟수를 총 30회로 제한하였다.

표 3 Linear-chain CRF와 Semi-CRF의 성능 비교

방법	평가 단계	F-measure	EA
CRF	분할만	98.78%	98.42%
	분할+태깅	97.61%	96.14%
	형태소 분석 전과정	**97.23%**	**95.21%**
Semi-CRF	분할만	97.84%	98.16%
	분할+태깅	96.59%	95.73%
	형태소 분석 전과정	96.83%	94.51%

표 3은 Sejong데이터에서 사용한 성능 결과를 보여준다. 표 3에서 CRF에 대한 결과는 linear-chain CRF의 성능 추치를 보여준다. 표 3에서 보다시피, 3.5에서 사용된 자질 집합에 대해서, linear-chian CRF가 Semi-CRF보다 전 과정에서 다소 높은 성능을 보여주었다. Semi-CRF가 이론적으로 linear-chain CRF를 포괄함에도 불구하고, 본 연구에서는 linear-chain CRF의 성능을 뛰어넘지 못했다. 3.5의 자질의 형태로는 Semi-CRF의 성능이 한계를 보여준다는 것을 의미한다.

Linear-chain CRF의 경우 상태 전이의 개수가 일정하는 점이 세그먼트의 개수에 따른 선호나 편향성 없이 학습될 수 있도록 하는 장점이 된 것으로 보인다. 물론, 이러한 결과는 본 연구에서 사용되는 자질 집합으로 국한된 것이므로, 두 방법에 대한 비교의 최종 결론을 도출하기 위해서는 보다 다양한 자질 집합에서 엄밀한 성능 비교를 해야 할 것이다.

5. 결론

본 논문은 형태소 분할과 품사 태깅을 위한 결합 모델로 linear-chain CRF와 Semi-CRF를 비교했다. 실험 결과, Semi-CRF가 이론적으로 linear-chain CRF를 포괄함에도 불구하고, 본 연구에서는 linear-chain CRF의 성능을 뛰어넘지 못했다. 본 연구는 Semi-CRF의 적용의 초기 연구로, 향후 개선의 여지가 남아있음을 물론이다. 후속 정교한 비교 연구를 통해, 위의 이슈에 대해 보다 엄밀한 실험을 수행할 예정이다. 학습 단계에서, Linear-chain CRF와 등가가 되는 고유 자질을 고정시키고, Semi-CRF에서 추가로 사용되는 자질만을 선택적으로 가중치 학습하는 것도 Semi-CRF의 성능 개선을 위한 한 방법이 될 것이다.

참고문헌

[1] Collins, M. (2002). Discriminative Traning Methods for Hidden Markov Models: Theory and Experiments with Perceptron Algorithms. In EMNLP (pp. 1-8).

[2] Lafferty, J., Mccallum, A., & Pereira, F. C. N. (2001). Conditional Random Fields : Probabilistic Models for Segmenting and Labeling Sequence Data. In ICML (Vol. 2001).

[3] Lee, D., & Rim, H. (2005). Probabilistic Models for Korean Morphological Analysis. IJCNLP (pp. 197-202).

[4] Lee, G. G., Cha, J., & Lee, J. (2002). Syllable-Pattern-Based Unknown- Morpheme Segmentation and Estimation for Hybrid Part-of-Speech Tagging of Korean. Computational Linguistics, 28(1).

[5] Sarawagi, S., & Cohen, W. W. (2004). Semi-Markov Conditional Random Fields for Information Extraction. In NIPS.

[6] Shim, K., & Yang, J. (2002). MACH - A Supersonic Korean Morphological Analyzer. COLING (pp. 939-945).

[7] 강승식. (1994). 다층 형태론과 한국어 형태소 분석 모델. 한글 및 한국어정보처리 학술대회 (HCLT) (pp. 140-145).

[8] 강승식, & 장병탁. (1996). 음절 특성을 이용한 범용 한국어 형태소 분석기 및 맞춤법 검사기. 정보과학회 논문지, 23(5).

[9] 강승식. (2002). "한국어 형태소 분석과 정보 검색", 홍릉과학출판사

[10] 나승훈, 양성일, 김창현, 권오욱, 김영길. (2012). CRF에 기반한 한국어 형태소 분할 및 품사 태깅. 한글 및 한국어 정보처리 학술대회

[11] 나승훈, 김창현, 김영길. (2013). CRF기반 한국어 형태소 분할 및 품사 태깅에서 두 단계 복합형태소 분해 방법. 한글 및 한국어 정보처리 학술대회

[12] 신준철, & 옥철영. (2012). 기분석 부분 어절 사전을 활용한 한국어 형태소 분석기. 한국정보과학회 논문지, 39(5), 415-424.

[13] 심광섭. (2013). 음절단위의 한국어 품사 태깅에서 원형 복원. 정보과학회논문지: 소프트웨어 및 응용, 40(3), 182-189.

[14] 심광섭. (2011). 형태소 분석기 사용을 배제한 음절 단위의 한국어 품사 태깅. 인지과학, 22(3), 327-345.

[15] 이재성. (2011). 한국어 형태소 분석을 위한 3단계 확률 모델. 정보과학회논문지, 38(5).

[16] 이창기. (2013). Structural SVM을 이용한 한국어 띄어쓰기 및 품사 태깅. In 한국컴퓨터종합학술대회 (pp. 604-606).

[17] 홍진표, & 차정원. (2008). 어절패턴 사전을 이용한 새로운 한국어 형태소 분석기. 한국컴퓨터종합학술대회 논문집 (Vol. 35)

CRF기반 한국어 형태소 분할 및 품사 태깅에서 두 단계 복합형태소 분해 방법

나승훈○ 김창현, 김영길
한국전자통신연구원
nash@etri.re.kr, chkim@etri.re.kr ,kimyk@etri.re.kr

Two-Stage Compound Morpheme Segmentation in CRF-based Korean Morphological Analysis

Seung-Hoon Na○, Chang-Hyun Kim, Young-Kil Kim
Natural Language Processing Laboratory
Electronics and Telecommunication Research Institute

요 약

본 논문은 CRF기반 한국어 형태소 분석 및 품사 태깅 과정에서 발생하는 미등록 복합형태소를 분해하기 위한 단순하고 효과적인 방법을 제안한다. 제안 방법은 1) 복합형태소를 내용형태소와 복합기능형태소로 분리하는 단계, 2) 복합기능형태소를 분해하는 두 단계로 구성된다. 실험 결과, 제안 알고리즘은 Sejong데이터에 대해, 기존의 lattice HMM 대비 높은 복합형태소 분해 정확률 및 두드러진 속도 개선을 보여준다.

주제어: 형태소분석, 복합형태소 분해, 복합기능형태소, 내용형태소, 최장일치

1. 서론

최근의 한국어 형태소 분석 연구는 입력문 전체에 대해 통계기반 모델을 수행하는 방법이 주된 축을 이루고 있다 [1,7,9,10,11]. 이 중, [7]은 한국어 형태소 분석을 형태소 분할 및 품사 태깅으로 나누어 각 단계에 CRF를 적용하는 방법을 제안했으며, [11]는 이를 확장하여 띄어쓰기, 형태소 분할 및 품사 태깅 단계를 모두 통합하는 joint모델을 제안했다.

그러나, 이들 연구에서는 복합형태소 처리 과정이 별도로 필요하다. 복합형태소 (compound morpheme)란 복수개의 단위형태소 (atomic morpheme)로 구성되는 형태소를 일컫는데, 표층형이 입력문과 일치하는 특징을 가지고 있다. 복합 형태소를 단위형태소로 분할하는 간단한 방법은 학습데이터에 나타나는 기분석 패턴을 이용하는 것이다 [7,11], 그러나, 복합형태소는 용언류의 활용형이 많아 대부분 개방어이므로 미등록어가 필연적으로 발생하게 된다. 미등록 복합형태소 처리를 위한 일반적인 모델로, [7]은 lattice HMM기반 방식을 제안하였다. 그러나, 이 방법은 높은 시간 복잡도를 지니며, 결과적으로 CRF와 이질적인 생성 모델의 도입으로 형태소 전 분석 과정을 한 가지 모델로 통일시키지 못했다.

본 논문은 미등록 복합형태소 처리를 위해 보다 단순하고 효율적인 방법을 제안한다. 제안 방법의 핵심은 기능형태소만을 기분석화하는 것으로, 다음의 두 단계로 이루어진다. 1) 복합형태소를 내용형태소와 복합기능형태소로 분리하는 단계, 2) 복합기능형태소를 분해하는 단계이다. Sejong 품사 부착 코퍼스에서 실험 결과, 제안 방법은 기존 lattice HMM방식에 비해 복합형태소 분해 정확률을 향상시킬 수 있을 뿐 아니라 두드러진 속도 향상을 보여주었다.

본 논문의 구성은 다음과 같다. 2장에서는 관련 연구를 다루고, 3장에서는 제안하는 두 단계 복합형태소 분해 방법을 소개하고, 4장에서는 실험을, 5장에서는 결론 및 향후 계획에 대해서 언급한다.

2. 관련 연구

최근의 한국어 형태소 분석의 연구는 입력문 전체에 별도의 언어학적 지식 사용없이 통계기반 모델을 수행하는 방법이 주된 축을 이룬다 [1,7,9,10,11]. 그러나, 한국어는 축약/불규칙 변이로 인해, 사전형과 표층형과 일치하지 않은 문제가 있어, 중국어의 경우와 달리 단어분리 (word segmentation) 및 품사 태깅 (POS tagging) 방법을 직접 적용할 수 없다. 이른바 *표층형과 사전형 불일치 문제*를 해소하는 방식에 따라, 최근 통계 기반 한국어 형태소 분석 연구는 다음의 두 가지 방식으로 나눌 수 있다.

먼저, 형태소 단위의 접근법으로, 표층형과 사전형 불일치를 복합형태소의 도입을 통해 해결하는 방식이다 [7,11]. 서론에서 언급했듯이, 복합형태소 분해의 대표적인 방법은 학습 코퍼스에 발생하는 복합형태소와 분해 결과를 기분석 패턴을 사용하는 것이다. 미등록 복합형태소의 일반적 처리를 위해, [7]는 lattice HMM의 생성 모델의 사용을 제안했다.

두 번째 방법은 [9,10]의 음절 단위의 접근법이다. 표층형으로부터 사전형으로 원형 복원을 위해 *음절 복원 사전*을 사용하는데, 이는 일종의 *음절 레벨에서의 기분석패턴*으로, 해당 음절/태그에 대해 원래 사전형에 대응되는 음절열/태그열을 저장해놓은 것이다. 가령, 축약의

경우, *복합음절*을 도입하여, 복합음절/음절복합태그에 대해 원래의 사전형의 음절열/태그열을 얻어낼 수 있도록 음절 복원 사전을 구성한다 (가/VVEP=>가/VV+았/EP).

3. 제안 방법

3.1. 개요

제안 방법은 다음의 두 단계로 구성된다.

1) 내용형태소와 복합기능형태소로 분할 (음절테이블 활용)
2) 복합기능형태소 분할 (기분석 활용)

그림 1에 제안 방법의 도식도를 보여준다. 첫 번째 단계에서는 복합형태소를 *내용형태소* (content morpheme)와 *기능형태소* (functional morpheme)로 분할한다. 내용형태소는 단위형태소로 구성, 기능형태소는 여러 개의 단위형태소로 이루어진 *복합기능형태소* (compound functional morpheme)이다. 두 번째 단계에서는, 복합기능형태소를 기분석패턴을 이용하여 단위기능형태소로 분해하는 방식으로 [7]과 동일한 방법을 사용한다.

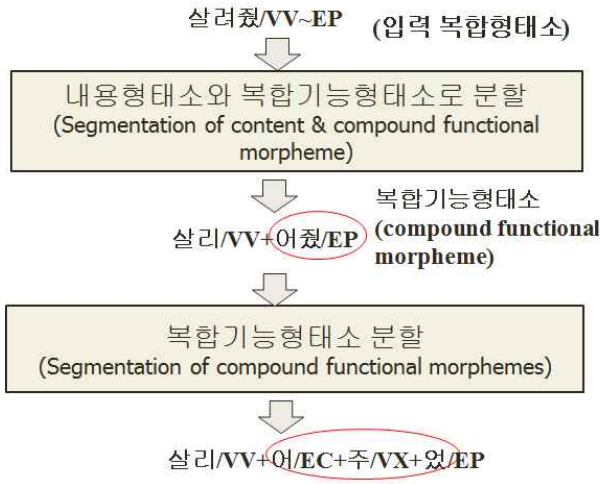


그림 4 제안 복합형태소 분해 절차

이해를 높기 위해, 아래, 그림 1의 제시된 예를 보다 상세히 기술한다. 복합형태소의 입력 예는 "살려줬/VV~EP "이며, VV~EP는 *복합태그*이다. 우선 "살려줬/VV~EP"이 복합형태소 기분석 패턴에 있는지를 조사한다. 기분석 패턴에 없을 경우 제안하는 2단계 과정을 통해 분해가 시도된다.

첫 번째 단계에서는 주어진 복합형태소가 *내용형태소*인 "살리/VV"와 *복합기능형태소*인 "어줬/EP"으로 분해해된다. 여기서, EP는 *복합기능형태소의 복합태그*로서 3.2절에서 보다 명확히 정의된다. 다음으로, 두 번째 단계에서는, 복합기능형태소 "어줬/EP"에 대한 기분석패턴에 기반하여 분해를 시도한다. 기분석 패턴으로부터 "어줬/EP: 어주/VX+었/EP"을 얻어내, 최종적으로 "살리/VV+어주/VX+었/EP"의 분해 결과를 얻는다.

제안 분해 방법은 본래 복합형태소단위로 기분석 패턴 중, 기능형태소에 대한 기분석 리스트만 유지하면 되기 때문에 원칙적으로 기분석 패턴의 개수를 대폭 감소시킬 수 있게 된다. 무엇보다 복합기능형태소는 1개의 단일형태소로 이루어지고, 2개 이상이 많지 않다는 점, 기능형태소는 용언류보다는 closed word에 해당되어 대부분은 사전에 등재 가능하는 점으로 인하여, 복합기능형태소의 기분석 패턴의 사용은 복합형태소 기분석패턴 기반 방법보다 현실성이 높다고 할 수 있다.

두 단계 복합형태소 분해를 위한 지식/사전은 모두 학습데이터로부터 자동으로 얻어진다. 다음 절부터, 분해 지식/사전을 어떻게 학습하는지 상세히 기술하도록 한다.

3.2. 복합형태소 분해 지식 학습

제안 복합형태소 분해를 수행하기 위해서는 다음의 세 가지 정보가 필요하다.

- **음절 매핑 테이블**, ST (*syllable mapping table*): 입력 표층형 s에 대해서 ST(s)는 원래 사전형과 해당되는 빈도수의 리스트를 리턴한다. ST의 예는 다음과 같다.

- ST(했)= {하았}
  ST(려)= {리어}
  ST(줬)= {주었}
  ST(써)= {쓰어}
  ST(겨)= {기어}
  (표기의 단순화를 위해 빈도수는 생략하였다)

  본 논문에서는 축약 문제로 한정, 입력음절열을 한음절, 출력음절로 2음절로 제한하여 설명한다. 물론, 제안 방법은 불규칙 활용 등을 포괄하는 형태로 알고리즘을 손쉽게 확장될 수 있다.
- **복합기능형태소 기분석 사전**, FLEX (*preanalyzed patterns for compound functional morpheme*): 입력 형태소 m, t에 대해서 FLEX(m,t)는 단위기능형태소 분해 결과 및 빈도수의 리스트를 리턴한다.
- **내용형태소 사전**, CLEX (*lexicon of content morphemes*): 입력 형태소 m, t에 대해서 CLEX(m,t)는 내용형태소 분해 결과 및 빈도수의 리스트를 리턴한다.

위의 세 가지 정보 중 음절 매핑 테이블 (ST)와 복합형태소 기분석 사전 (FLEX)은 복합형태소의 기분석 리스트로부터, 내용형태소 사전 (CLEX)은 학습코퍼스의 단위형태소로부터 얻어진다. ST, FLEX, CLEX모두 리스트는 빈도수로 정렬(sorting)하여, 가장 빈도수가 높은 엔트리가 리스트의 첫 번째에 위치하도록 배치한다.

보다 명확한 설명을 위해, 아래, ST, FLEX, CLEX의 학습 과정을 형식적으로 기술한다.

분해 지식 학습을 위한 기본 정보 추출

복합형태소에 대한 기분석 엔트리가 $m/t : m_1/t_1, m_2/t_2, \cdots, m_k/t_k$로 주어졌다고 하자. l_i는 m_i의 음절 갯수를, $m_i[j]$는 m_i의 j번째 음절을, $m_i[j \cdots k]$

는 m_i의 j부터 k까지의 음절열을, $m_i[j\cdots]$는 m_i의 j부터 마지막음절까지의 음절열을 가리킨다고 하자. "+"는 음절열의 결합 (concatenation) 연산자라고 정의하자.

학습의 핵심은 주어진 복합형태소 음절열 m과 이를 구성하는 단위형태소들로부터, *head, tail, compressed* (축약음절), *decompressed1* (분해요소음절1), *decompressed2* (분해요소음절2) 의 정보를 추출하는 것으로, 각각에 대한 정의는 다음과 같다.

정의 1. *head*는 처음 단위형태소 m_1의 l_1-1번째까지의 음절열로 정의한다

head = $m_1[1\cdots l_1-1]$

정의 2. tail은 복합형태소의 m의 l_1+1부터 마지막까지의 음절열로 정의한다.

tail = $m[l_1+1\cdots l]$

여기서 l은 m의 길이이다.

정의 3. *compressed*는 복합형태소의 m의 l_1-1번째 음절을 축약된 음절로 정의한다.

compressed = $m[l_1-1]$

정의 4. 축약된 음절 *compressed*에 대응되는 사전형 두음절은 *decompressed1*와 *decompressed2* 다음과 같이 정의된다.

decompressed1 = $m_1[l_1]$
decompressed2 = $m_2[1]$

여기서, 두 개의 분해요소음절, *decompressed1, decompressed2*은 축약음절 *compressed*에 대응되는 정보이다.

예를 들어, "구겨진/VV~ETM: 구기/VV+어/EC+지/VX+ㄴ/ETM"의 경우, 위의 정의에 따라, *head, tail, compressed, decompressed1, decompressed2* 정보는 다음과 같이 주어진다.

head	구
tail	진
compressed	겨
decompressed1	기
decompressed2	어

정의 5. 위의 정의로부터, 내용형태소의 사전형 *cm*, 복합기능형태소 표층형 *fm*은 각각 다음과 같이 정의된다.

cm = m_1
fm = *decomposed2* + *tail*

앞서의 예에서, *cm*과 *fm*은 다음과 같이 정의된다.

cm	구기
fm	어진

위의 정보에 기반, ST, FLEX는 아래와 같이 학습된다.

음절매핑테이블 (ST) 구축: 축약음절 *compressed*에 대해 분해요소음절 *decompressed1*+*decompressed1*에 *대한 매핑을* 음절매핑테이블의 한 엔트리로 구성한다.

복합기능형태소 (FLEX) 기분석 사전 구축: 기능형태소 *fm*에 대한 단위형태소들은 $m_2/t_2,\cdots,m_k/t_k$임로, 이들은 복합기능형태소 기분석 패턴의 엔트리로 추가한다.

여기서, 복합기능형태소의 복합태그 *ft*는 t와 $t_1\cdots t_k$에 의존하는데, 다음은 몇 가지 예시들이다.

1. 마지막태그만 사용: t_k
   예) 해진/ETM (행해진/VV~ETM: 행하/VV+아/EC+지/VX+ㄴ/ETM)
2. 시작태그만 사용: F-t_1
   예) 해진/F-VV (행해진/VV: 행하/VV+아/EC+지/VX+ㄴ/ETM)
3. 시작 및 마지막 태그: t_2-t_k
   예) 해진/EC~ETM (행해진/VV~ETM: 행하/VV+아/EC+지/VX+ㄴ/ETM)
4. 시작부터 마지막 태그: = $t_2\cdots t_k$
   예) 해진/EC~VX~ETM (행해진/VV~ETM: 행하/VV+아/EC+지/VX+ㄴ/ETM)

본 논문에서는, [7]의 복합형태소의 복합태그의 정의를 참조하여 복합기능형태소의 복합 태그를 다음과 같이 정의한다. 즉, 용언류에 대해서는 시작과 마지막 태그가 주어지므로, 방법 1을 사용, 나머지형태소에 대해서는 모두 방법2를 사용한다. 즉, 방법3-4는 복합형태소가 복합태그가 해당 정보를 제공해줄 수 있을 때에만 활용될 수 있는 것으로 본 논문에서는 고려되지 않는다.

내용형태소 (CLEX) 사전 학습: 위의 두 정보 ST, FLEX와 달리, 단위형태소 사전은 품사 부착 학습코퍼스 전체로부터 얻어진다. 사전의 엔트리는 단위형태소의 사전형과 품사태그 그리고 빈도수로 구성된다.

3.3. 최장 일치 기반 복합형태소 분해 알고리즘

위의 세 가지 지식, ST, FLEX, CLEX를 바탕으로, 본 논문에서 복합 형태소 분해 알고리즘은 최장일치법에 기반한다. 분해 알고리즘의 핵심적인 가정은 *복합형태소내의 축약음절은 최대 단 한 개만이 존재한다*는 것이다. 즉, 복합형태소내의 어느 위치에서 축약이 발생했는지를 판단하는 것이 알고리즘의 메인이 된다.

분해 알고리즘은 *복합형태소 표층형의 마지막 음절부터 순차적으로 해당 음절이 축약음절인지를 조사*한다. 축약여부는 음절매핑테이블을 참조하여 이를 2음절로 분해하고 분해된 음절 앞부분은 내용형태소로, 뒷부분은 기능형태소로 포함된다고 가정하고, 이들이 실제 CLEX, FLEX사전에 있는지 확인한다. 만약, 모두 사전에 있으면 바로 해당 음절위치에서 축약이 이루어진 것이다. 그렇

지 않으면, 음절 위치를 앞으로 한칸 이동하여, 다시 축약음절인지 아닌지 조사한다.

상세한 분해 알고리즘은 다음과 같다.

```
input: m/t   ; 복합형태소

ct = getCTag(t); 내용형태소 태그
ft = getFTag(t); 기능형태소 태그

for (i=l;i ≥ 1;i--)
    syllSet = ST(m[i]); 분해요소음절 획득
    if syllSet == NULL:
       continue
    for each decomps in syllSet:
       cm = m[1 ··· i−1] + decomps[1] ; 내용형태소
       fm = decomps[2] + m[i+1 ··· l]; 기능형태소

       if CLEX(cm,ct) is not empty
         && FLEX(fm,ft) is not emptry:
           return 해당 분해 결과
return NULL
```

여기서, getCTag(t)와 getFTag(t)는 복합 태그 t로부터 각각 내용형태소의 태그 및 복합기능형태소의 태그를 얻어내는 함수이다. 위의 알고리즘에서 CLEX, FLEX에 모든 복수개의 엔트리가 있는 경우에는 기분석 결과를 제시할 때, 애매성이 발생하게 된다. 애매성이 발생하는 경우, 가장 높은 빈도수를 갖는 엔트리를 취했다.

4. 실험

실험을 위해, 세종코퍼스 약 25여만문장중 80%는 학습데이터로, 20%는 테스트데이터로 사용하였다. 형태소 분할 및 품사 태깅 모델로 [11]과 유사하게, 형태소 분할과 태깅을 동시에 수행하는 음절 기반 결합 모델을 적용하였다. 이때 사용되는 자질은 [7]의 형태소 분할을 위해 사용되는 자질과 동일하다.

표 1은 [7]의 CRF기반 형태소 분할/품사 태깅 과정에서, 제안 방법과 기존 방법의 복합형태소 분해 정확도를 비교한 결과이다. 표 1에서 보다시피 제안 방법은 기존 lattice HMM기반 방법보다 높은 분해 정확률을 보여주고 있다.

표 1. 복합형태소 분해에서 제안 두 단계 방법과 lattice HMM 방법 비교 (복합형태소만을 대상으로 평가)

방법	성능 (Accuracy)
기분석+lattice HMM [7]	92.66%
기분석+제안방법	**93.84%**

표 2는 전체 형태소 분석에서 [7]의 CRF기반 방법의 복합형태소 분해 단계가 끝난 후 최종 성능을 비교한 것이다. 표 2에서 보다시피 최종 성능에서도, 제안 방법은 기존 CRF기반 방법대비 다소 높은 성능을 보여준다.

표 3는 전체 테스트 문장에 대해 복합형태소 분해 소요시간에 대해 기존 방법과 제안 방법을 비교한 것이다. 표에서 확인되는 것처럼, 제안 방법은 기존 방법과 비교하여 훨씬 적은 시간을 소요하였다.

요약 하면, 제안 방법은 기존방법 대비, 분해 정확률을 높일 뿐 아니라, 시간 복잡도를 두드러지게 개선할 수 있음을 보여준다.

표 2. 음절 기반 형태소분할/태깅에서 분석 결과에서 제안 복합형태소 분해 방법과 기존 방법과의 비교 (전체 형태소 대상)

방법	성능 (F-measure)
CRF+기분석	97.08%
CRF+기분석+lattice HMM [7]	97.21%
CRF+기분석+제안방법	**97.23%**

표 3. 제안 방법과 기존 방법의 복합형태소 분해의 시간복잡도 비교

방법	소요 시간
lattice HMM [7]	65.76 sec
제안 방법	**4.77 sec**

5. 결론

본 논문은 미등록 복합형태소 분해를 위해 단순하고 효율적인 두 단계 방법을 제안하였다. 제안 방법은 두 단계로 구성되는데 먼저 복합형태소를 내용형태소와 복합기능형태소로 분해하고, 다음으로 복합기능형태소를 기분석패턴으로 처리한다. 결국, 제안 방법은 복합기능형태소에 대해서만 기분석 패턴을 유지하기 때문에 기존의 방식에 비해서 기분석 패턴의 수를 크게 줄일 수 있는 장점이 있다. 실험 결과, 제안 방법은 기존의 lattice HMM기반 분해보다 정확도와 시간 복잡도를 모두 개선하였다.

본 논문에서는 음절매핑테이블을 구성할 때 축약음절만을 고려하였으나, 일반적으로는 불규칙 변형 등 보다 다양한 유형을 포괄하도록 확장하는 것이 필요하다. 또한, 제안 분해 알고리즘은 최장일치 방식에 기반을 두고 있으나, 전체적으로 가장 개연성이 높은 분해 결과를 찾는 글로벌 분해 방법을 도입하여 분해 알고리즘을 보다 정교화 하는 것도 흥미로운 주제가 될 것이다.

참고문헌

[1] Lee, D., & Rim, H. (2005). Probabilistic Models for Korean Morphological Analysis. IJCNLP (pp. 197-202).

[2] Lee, G. G., Cha, J., & Lee, J. (2002). Syllable-Pattern-Based Unknown- Morpheme Segmentation and Estimation for Hybrid Part-of-Speech Tagging of Korean. Computational Linguistics, 28(1).

[3] Shim, K., & Yang, J. (2002). MACH - A Supersonic Korean Morphological Analyzer. COLING (pp. 939-945).

[4] 강승식. (1994). 다층 형태론과 한국어 형태소 분석 모델. 한글 및 한국어정보처리 학술대회 (HCLT) (pp. 140-145).

[5] 강승식, & 장병탁. (1996). 음절 특성을 이용한 범용

한국어 형태소 분석기 및 맞춤법 검사기. 정보과학회 논문지, 23(5).
[6] 강승식. (2002). “한국어 형태소 분석과 정보 검색“, 홍릉과학출판사
[7] 나승훈, 양성일, 김창현, 권오욱, 김영길. (2012). CRF에 기반한 한국어 형태소 분할 및 품사 태깅. In 한글 및 한국어 정보처리.
[8] 신준철, & 옥철영. (2012). 기분석 부분 어절 사전을 활용한 한국어 형태소 분석기. 한국정보과학회 논문지, 39(5), 415-424.
[9] 심광섭. (2011). 형태소 분석기 사용을 배제한 음절 단위의 한국어 품사 태깅. 인지과학, 22(3), 327-345.
[10] 심광섭. (2013). 음절단위의 한국어 품사 태깅에서 원형 복원. 정보과학회논문지: 소프트웨어 및 응용, 40(3), 182-189.
[11] 이창기. (2013). Structural SVM을 이용한 한국어 띄어쓰기 및 품사 태깅. In 한국컴퓨터종합학술대회 (pp. 604-606).

대화형 개인 비서 시스템을 위한 하이브리드 방식의 개체명 및 문장목적 동시 인식기술

이창수○, 고영중
동아대학교
blue772001@gmail.com, youngjoong.ko@gmail.com

A Simultaneous Recognition Technology of Named Entities and Objects for a Dialogue Based Private Secretary Software

ChangSu Lee○, YoungJoong Ko
Donga University, Computer Engineering

요 약

기존 대화시스템과 달리 대화형 개인 비서 시스템은 사용자에게 정보를 제공하기 위해 앱(APP)을 구동하는 방법을 사용한다. 사용자가 앱을 통해 정보를 얻고자 할 때, 사용자가 필요로 하는 정보를 제공해주기 위해서는 사용자의 목적을 정확하게 인식하는 작업이 필요하다. 그 작업 중 중요한 두 요소는 개체명 인식과 문장목적 인식이다. 문장목적 인식이란, 사용자의 문장을 분석해 하나의 앱에 존재하는 여러 정보 중 사용자가 원하는 정보(문장의 목적)가 무엇인지 찾아주는 인식작업이다. 이러한 인식시스템을 구축하는 방법 중 대표적인 방법은 사전규칙방법과 기계학습방법이다. 사전규칙은 사전정보와 규칙을 적용하는 방법으로, 시간이 지남에 따라 새로운 규칙을 추가해야하는 문제가 있으며, 규칙이 일반화되지 않을 경우 오류가 증가하는 문제가 있다. 또 두 인식작업을 파이프라인 방식으로 적용 할 경우, 개체명 인식단계에서의 오류를 가지고 문장목적 인식단계로 넘어가기 때문에 두 단계에 걸친 성능저하와 속도저하를 초래할 수 있다. 이러한 문제점을 해결하기 위해 우리는 통계기반의 기계학습방법인 Conditional Random Fields(CRF)를 사용한다. 또한 사전정보를 CRF와 결합함으로써, 단독으로 수행하는 CRF방식의 성능을 개선시킨다. 개체명과 문장목적인식의 구조를 분석한 결과, 비슷한 자질을 사용할 수 있다고 판단하여, 두 작업을 동시에 수행하는 방법을 제안한다. 실험결과, 사전규칙방법보다 제안한 방법이 문장단위 2.67% 성능개선을 보였다.

주제어: 개체명 인식, 하이브리드, Conditional Random Fields, 문장목적인식, 대화시스템

1. 서론

모바일기술의 발달로, 대화시스템이 기존의 오프라인 대화시스템으로부터, 실시간 정보획득 및 질의응답을 목적으로 한 온라인 대화시스템으로 변화되고 있다. 온라인 대화시스템의 하나인 대화형 개인 비서 시스템의 가장 큰 장점은 앱(APP)을 구동할 수 있는데 있다. 기존의 대화시스템은 규칙이나 학습을 통해 정해진 질문에 정해진 대답만을 할 수 있는데 반해, 이 시스템은 앱을 통해 사용자의 질문에 따라 사용자가 원하는 화면을 보여주며, 실시간으로 정보를 제공한다.

사용자가 앱을 통해 정보를 얻고자 할 때, 시스템에서 사용자가 필요로 하는 정보를 제공해주기 위해서는 사용자의 목적을 정확히 인식하는 작업이 중요하다. 이러한 작업 중 중요한 두 가지 요소는 개체명 인식과 문장목적 인식이다.

개체명 인식은 질의응답시스템과 정보검색 분야에서 유용하게 사용되고 있는 정보추출의 한 단계이다. 개체명은 인명, 지명, 조직명, 시간, 날짜, 화폐 등의 고유명사이며, 개체명인식은 문장에서 개체명을 식별하고 식별된 개체명의 종류를 결정하는 작업이다. 이 작업은 문장에서 중요한 핵심어를 추출해 문장의 의미를 파악하는데 도움을 준다.

개체명 인식 방법은 크게 세 가지 방법으로 나눌 수 있다. 첫 번째 방법은 규칙기반 개체명 인식[1][2][3]이며, 규칙기반 개체명 인식은 사전정보나 규칙만을 사용하기 때문에 다음과 같은 문제점을 가지고 있다.

1. 사전정보나 규칙이 포함되어 있지 않은 문장 인식 문제.
2. 시간이 지남에 따라, 규칙을 지속적으로 추가해줘야 하는 문제.
3. 규칙이 일반화되지 않을 경우 많은 오류를 유발 하는 문제.

두 번째 방법은 통계기반의 기계학습 방법이다[4][5][6][7]. 통계기반의 방법을 사용하기 위해선 양질의 말뭉치가 필요하며, 자질을 색출하는 작업을 거쳐야 한다. 마지막으로 규칙기반과 기계학습을 결합한 방법도 있다[8][9].

문장목적인식이란, 앱 구동을 통해 실행된 하나의 앱

본 연구는 산업자원통상부 및 한국산업기술평가관리원의 산업융합원천기술개발사업(정보통신)의 일환으로 수행하였음.
[10041678, 다중영역 정보서비스를 위한 대화형 개인 비서 소프트웨어 원천 기술 개발]

에서는 여러 정보가 있을 수 있는데, 그 정보들 중 사용자가 필요로 하는 정보를 제공하기 위해 문장의 목적을 찾는 작업이다. 예를 들어, 날씨 앱을 구동 했을 때, 사용자는 오늘 비가 올 것인지, 기온이 어떻게 되는지, 날씨가 맑은지, 바람이 많이 부는지 등 여러 정보 중 원하는 하나의 정보가 있을 것이다. 문장목적 인식은 사용자의 문장을 분석해, 위와 같은 여러 정보 중 사용자가 필요로 하는 정보가 무엇인지 찾아 주는 인식작업이다. 기존의 대화시스템에서는 이러한 문장목적을 인식하는 시스템이 존재하지 않았다. 하지만 앱을 구동하는 대화형 개인 비서 시스템에서는 문장목적을 인식하는 것은 사용자가 원하는 정보를 앱에서 찾아 사용자에게 정확히 제공해주기 위해 반드시 필요한 인식작업이다.

대화형 개인 비서 시스템에선 개체명 인식, 문장목적 인식을 통해 앱에서 사용자에게 제공해야 할 정보가 무엇인지 파악한다. 예를 들어, 사용자가 “오늘 기온”에 대한 정보를 얻고자 할 때, 사용자는 “오늘 기온이 어때?” 라는 질의를 시스템에게 보내고, 개체명 인식 작업을 통해 “오늘”이라는 개체명을 인식한 후, 문장목적 인식 작업을 통해 문장의 목적이 “기온”에 대한 정보라는 것을 파악한다. 이러한 과정을 거쳐 시스템은 날씨앱을 실행시켜, 오늘의 기온에 대한 정보를 앱에서 찾아 사용자에게 제공해주는 것이다.

본 논문에서는 개체명 및 문장목적 인식을 위해 통계기반의 기계학습기법인 CRF와 사전정보를 함께 사용하는 하이브리드 방식을 제안한다. 또한 개체명과 문장목적을 인식하는 방법의 구조를 분석한 결과, 서로 비슷한 자질로 분류가 가능하다고 판단하여 서로 다른 두 개의 인식 시스템을 단 한 번의 기계학습을 통해 동시에 인식하는 방법을 제안한다. 2장에서는 관련연구에 대해 살펴보고, 3장에서는 개체명 인식 시스템, 문장목적 인식 시스템, 개체명 및 문장목적 동시 인식 시스템에 대해 각각 사전규칙, CRF, 하이브리드방식으로 나누어 설명하고, 4장에서는 3장에서 제안한 하이브리드 시스템을 사전규칙 시스템과 CRF만을 사용한 시스템과 비교하여 하이브리드 시스템의 유용성을 살펴보며, 본 논문에서 제안한 하이브리드 기반의 개체명 및 문장목적 동시 인식기술이 사전규칙방식이나 단독CRF방식을 사용해 두 인식을 수행한 것보다 더 합리적인 것을 살펴본다. 5장에서는 결론 및 향후 연구과제에 대하여 기술한다.

2. 관련 연구

개체명 인식은 질의응답시스템과 정보검색 분야에서 유용하게 사용되고 있는 정보 추출의 한 분야로서 문서나 문장 내에서 개체명을 추출하고 추출된 개체명의 종류를 식별 하는 작업을 말한다. 개체명 인식에 관한 연구는 1990년대에 정보추출(Information Extraction)의 목적으로 개최되었던 Message Understanding Conference (MUC)에서 본격적으로 연구되기 시작해, MUC 이후 개체명에 대한 연구가 꾸준히 진행 되어왔으며, Conference on Computational Natural Language Learning 2002(CoNLL 2002)와 CoNLL2003을 통해서 더욱 많은 발전이 있었다[10]. 개체명 인식은 크게 3가지 방법으로 연구되었다. 첫 번째는 규칙 기반 방법이며 이 방법에서는 주로 정규표현식[3]이나 자연어 특징을 이용한 규칙과 사전정보[2]를 사용했다. 두 번째로 통계기반의 기계학습 방법이며, 대표적인 방법으로 Hidden Markov Model, Maximun Entropy Model, Conditional Random Fields, Decision Tree 등이 있다[4][5][6][7]. 마지막으로 규칙기반과 기계학습을 함께 사용한 하이브리드 방법도 연구되었다[8][9]. 개체명 인식 연구가 시작된 1990년대에는 대부분 영어만을 대상으로 개체명 인식 연구가 이루어졌지만 최근에서 영어뿐만 아니라 한국어[5], 일본어, 중국어 등 다양한 언어에 대해서 개체명 인식 시스템이 연구되고 있다.

3. 개체명 및 문장목적 인식

우리는 사전정보와 CRF를 결합하는 하이브리드 방법을 통해 개체명 및 문장목적 인식의 성능을 높이는 방법을 시도하며, 개체명과 문장목적 인식이라는 각각의 작업을 동시에 수행하는 방법을 제안한다.

대화형 개인 비서 시스템에서 사용자의 목적을 파악해 원하는 정보를 실시간으로 제공 하기위해서는 개체명과 문장목적을 인식하는 작업이 필요하다. 이에 따라 본 논문은 다음의 3가지 인식작업을 나눠서 살펴 본 후, 개체명과 문장목적을 동시에 인식하는 방법이 합리적인 접근방법임을 보인다.

1.개체명 인식
2.문장목적 인식
3.개체명과 문장목적의 동시 인식

대화형 개인 비서 시스템은 6개의 도메인을 가지며, 이 6개의 도메인에서 출현하는 개체명과 문장목적을 인식 대상으로 한다.

표1. 도메인 종류

교통	날씨	시계	알람	일정	환율

3.1 개체명 인식

개체명의 종류는 표2와 같이 8가지로 분류 된다.

표2. 개체명 종류

인명 (Person)	지명 (Location)	날짜 (Date)	시간 (Time)
반복 (Cycle)	타이틀 (Title)	통화 (Currency)	숫자 (Number)

개체명 종류에서 반복은 “매년, 매주, 매월” 같은 단어이며, “논문 계획, 미팅 일정, 점심 약속” 같은 단어를 타이틀이라고 명명한다. 개체명 인식 시스템은 비교를 위해 사전규칙, CRF, 하이브리드 방식으로 각각 시스템을 구축한다. 먼저, 사전규칙기반은 각 도메인별로 사전과 규칙을 다르게 적용하여, 도메인별로 사전이나 규칙이 중복되어 오류를 유발할 수 있는 부분을 최대한 줄였다. 다음은 사전 정보와 규칙적용 방식을 보여준다.

그림1. 사전매치방법(왼쪽)과 규칙적용방법(오른쪽)

규칙은 여러 가지 타입에 유연하게 적용할 수 있도록, 어휘정보 및 형태소정보, 정규표현 등 5가지의 타입규칙을 사용하였다. 또한 사전정보는 최장길이일치법을 적용했다. 하이브리드 방식에서는 개체명 인식 시스템의 성능 향상을 위해 CRF에 사전정보를 결합함으로써 개체명 인식 시스템의 성능을 높이는 방법을 시도했다. 또한 사전정보와 CRF를 각각 단독으로 사용하는 시스템을 구축해 하이브리드 방식이 합리적인지 평가했다. 그림2는 하이브리드 기반의 개체명 인식시스템의 구조도이다.

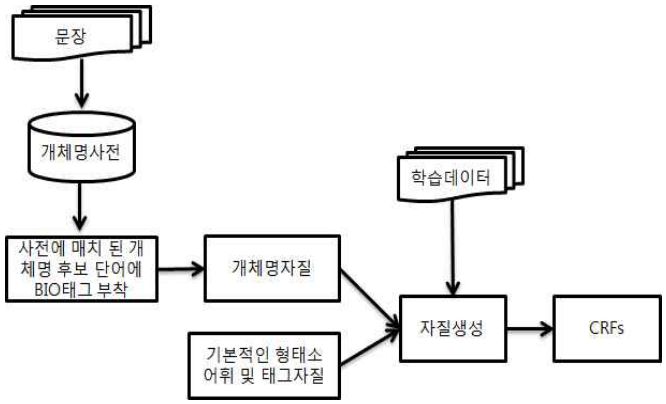


그림2. 하이브리드기반의 개체명 인식시스템 구조도

사전정보를 이용해 사전에 매치된 단어는 개체명 후보로 인식하며, 개체명인식 중 날짜와 시간은 정해진 템플릿이 존재하기 때문에 기존의 규칙시스템에서 사용한 방법을 그대로 사용한다. 그리고 사전에 매치된 단어는 그림3과 같이 BIO태그를 부착해 CRF를 사용할 때 개체명사전자질을 사용할 수 있도록 구성했다.

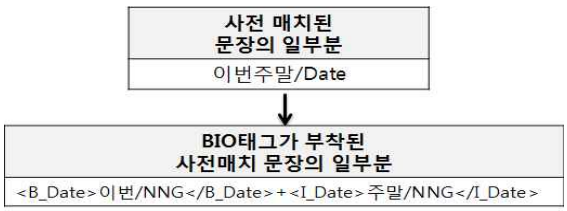


그림3. BIO태그가 부착된 사전정보

3.2 문장목적 인식

문장목적의 종류는 6개의 도메인에서 28개의 세부목적이 존재한다. 대표적인 문장목적 종류는 알람(Alarm), 지역시간(LocationTime), 버스스케줄(busSchedule), 날씨정보(weatherInfo)등이 있다.

문장목적 인식에서는 문장의 목적을 파악하기 위해 다음과 같은 두 가지 학습방법을 사용할 수 있다.

1.문장 전체를 학습시켜 목적을 찾아내는 방법

2.문장에서 중요한 실마리가 되는 부분의 구간을 정해 그 구간이 존재하는지 확인해 목적을 찾아내는 방법.

문장전체를 학습시키는 방법은 개체명과 문장목적 인식을 수행하기 위해 서로 다른 기계학습 기법을 사용해야 하는 단점이 있다. 그 이유는, 인식 단위의 차이 때문이다. 즉, 개체명 인식에서는 형태소별, 목적인식에서는 문장별로 분류하게 된다. 실시간 시스템에서는 속도가 매우 중요하기 때문에, 우리는 두 가지 인식을 동시에 수행하기 위한 기반을 마련하기 위해 실마리구간 탐색방법을 사용했다.

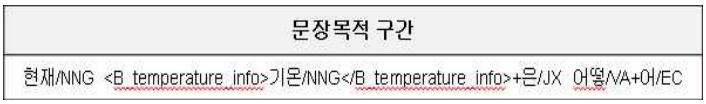


그림4. 문장의 실마리 구간

그림4에서는 문장의 목적이 기온정보인 경우 문장에서 기온정보의 실마리 구간을 “기온”라는 단어를 실마리라고 정해 이 문장이 기온정보에 대한 목적을 가진 문장인지를 식별하게 해준다.

문장목적 인식 사전규칙시스템은 개체명 인식과 같은 방법으로 각 도메인별 사전과 규칙을 적용시켜 오류를 최소화 시켰다. 통계기반의 문장목적 인식시스템도 마찬가지로 사전정보와 CRF를 결합하는 하이브리드 방식을 통해 성능을 높이는 방법을 시도했다. 또한 사전정보와 CRF를 각각 단독으로 사용하는 시스템을 구축해 하이브리드 방식이 합리적인지 평가했다. 그림5는 문장목적인식의 구조도이다.

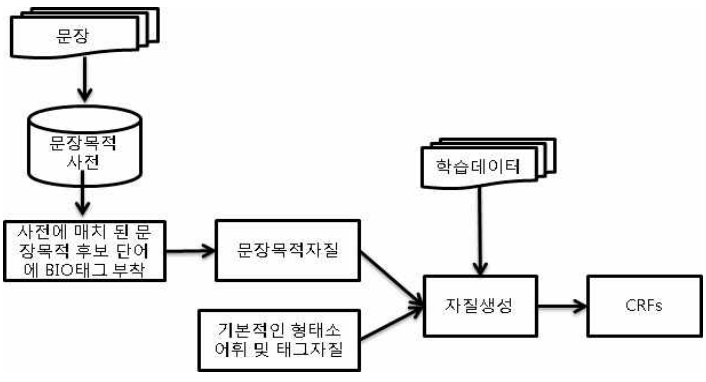


그림5. 하이브리드기반의 문장목적 인식시스템 구조도

3.1절의 개체명 인식과 같은 단계를 거쳐 목적구간을 찾아 문장의 목적 인식작업을 수행하도록 구축했다.

3.3 개체명 문장목적 동시 인식

3.1과 3.2절에서 개체명과 문장목적을 인식하는 구조

도를 보였다. 두 시스템의 비교결과, 문장목적인식에 구간정보를 사용함으로써, 개체명 인식과 같은 시스템구조를 사용할 수 있으며, 따라서 우리는 두 인식 시스템이 서로 비슷한 자질로 분류가 가능할 것이라고 판단했다. 이에 따라 개체명과 문장목적을 동시에 인식하는 시스템을 구현 하는 것은 합리적인 방법일 것이라고 생각했다. 우리가 생각한 방법이 올바른지 판단하기 위해 사전규칙, CRF, 하이브리드기반 시스템을 각각 구현해 비교했다.

사전규칙기반 시스템은 개체명과 문장목적인식을 파이프라인 방식으로 수행한다. 그림6은 사전규칙기반 시스템의 구조도이다.

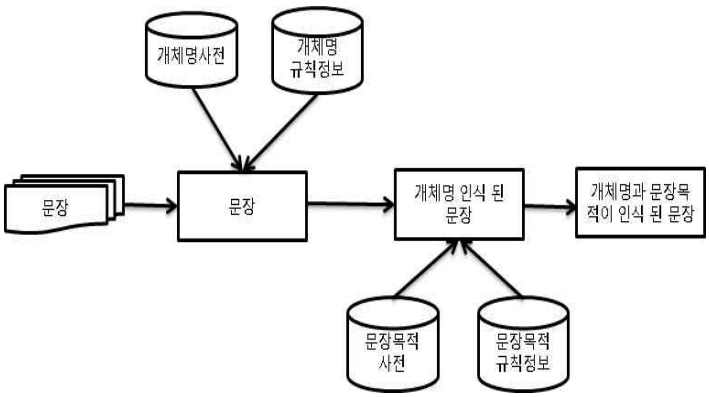


그림6. 개체명과 문장목적 인식을 파이프라인 방식으로 수행하는 사전규칙기반 시스템의 구조도

사전규칙기반은 파이프라인 방식을 수행함에 있어, 다음과 같은 문제점을 가지게 된다.

1.개체명 단계에서 생긴 오류를 가지고 문장목적 단계로 넘어가기 때문에 두 단계에 걸친 성능저하.

2.두 단계의 순차적 수행으로 인한 속도저하.

반면, 본 논문에서 제안한 시스템은 두 인식 시스템이 같은 구조를 갖는다는 것을 바탕으로 개체명과 문장목적을 동시에 수행함으로써 위의 문제점을 해결한다. 동시인식 시스템 또한 사전정보와 CRF를 결합하는 하이브리드 방식을 통해 성능을 높이는 방법을 시도했다. 또 CRF를 단독으로 사용하는 시스템을 구축해 제안한 방법이 합리적인지 평가했다. 그림7은 제안한 하이브리드기반 시스템의 구조도이다.

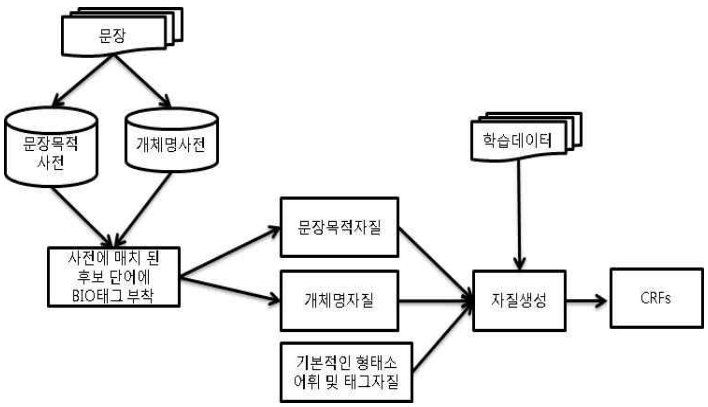


그림7. 개체명과 문장목적을 동시에 인식하는 하이브리드기반 시스템의 구조도

하이브리드기반 시스템은 사전정보를 활용해, 사전에 매치된 개체명후보와 문장목적 구간후보를 인식해, CRF를 사용할 때 사전자질을 사용할 수 있도록 구성했다.

3.4 자질집합

자질 집합은 형태소어휘, 형태소태그, 어절 내 자질등 기본적으로 사용하는 자질집합과 함께, 인식 성능을 높이기 위해 본 논문에서 구축한 사전자질, 개체명자질, 문장목적 자질을 추가하였다. 표3는 전체 자질집합을 보여준다.

표3. 시스템에 사용된 자질집합

형태소어휘태그 자질	1.형태소어휘/태그
형태소어휘자질	2.w_0/w_1 3.w_{-1}/w_0 4.$w_{-1}/w_0/w_1$, 5.$w_0/w_1/w_2$ 6.$w_{-2}/w_{-1}/w_0$ 7.$w_{-1}/w_0/w_1/w_2$ 8.$w_{-2}/w_{-1}/w_0/w_1$ 9.$w_{-2}/w_{-1}/w_0/w_1/w_2$
형태소태그자질	10.p_0/p_1 11.p_{-1}/p_0 12.$p_{-1}/p_0/p_1$ 13.$p_0/p_1/p_2$ 14.$p_{-2}/p_{-1}/p_0$ 15.$p_{-1}/p_0/p_1/p_2$ 16.$p_{-2}/p_{-1}/p_0/p_1$ 17.$p_{-2}/p_{-1}/p_0/p_1/p_2$
어절 내 자질	18.형태소 어절 내 위치 19.형태소태그/어절길이
사전 자질	20.형태소어휘/태그 + 레이블정보
개체명자질	21.현재 위치의 앞에 나온 모든 개체명 정보 22.현재 위치의 앞에 나온 모든 개체명의 시퀀스정보(NULL 포함) 23.형태소어휘/태그와 문장에 존재하는 모든 개체명정보 24.형태소어휘/태그와 문장에 존재하는 모든 개체명의 시퀀스정보 (NULL포함)
문장목적자질	25.현재 위치의 앞에 나온 모든 문장목적 정보 26.현재 위치의 앞에 나온 모든 문장목적의 시퀀스정보(NULL 포함) 27.형태소어휘/태그와 문장에 존재하는 모든 문장목적정보 28.형태소어휘/태그와 문장에 존재하는 모든 문장목적의 시퀀스정보 (NULL포함)

시스템에 사용된 자질집합은 총 28개로 구성된다. 형태소어휘자질은 현재형태소어휘(w_0)를 중심으로 이전/이후의 연속된 형태소어휘이다. 형태소태그자질도 마찬가지로 현재형태소태그(p_0)를 중심으로 이전/이후의 연속된 형태소태그이다. 형태소의 어절 내 위치자질은 형태소가 어절의 시작, 중간, 끝을 나타내는 정보이며, 형태소태그/어절길이자질은 형태소태그와 형태소를 포함하는 어절의 길이정보이다. 마지막으로 사전자질은 형태소어휘/태그와 형태소의 레이블정보이다. 레이블은 사전매치를 통해 후보가 된 개체명이나 문장목적에 BIO태그가 부착된 정보이다.

4. 실험결과

모든 실험결과는 정확도(Accuracy)를 측정해 성능을 평가하며, 개체명과 문장목적을 동시에 인식하는 시스템은 문장 내에 모든 개체명과 함께 문장의 목적을 정확히 인식했을 경우에만 맞는 문장이라고 간주한다. CRF 및 하이브리드기반 시스템에 사용한 자질 중 실험결과 가장 좋은 성능을 보이는 자질집합을 각 실험결과의 하위에 기재했다. 또한 모든 자질은 표3에 기재한 자질만을 사용했으며, 각 실험에서 사용한 자질을 쉽게 보여주기 위해 자질 별로 번호를 부여해, 자질번호 나열을 통해 사용한 자질을 보여주는 방식으로 구성했다.

4.1 실험데이터

개체명과 문장목적 인식을 위해 ETRI개체명사전을 사용했으며, 학습데이터를 통해 문장목적구간사전을 구축했다. 또한 말뭉치는 전문적으로 태깅을 하는 전문가를 통해 6개의 도메인에서 2972개의 문장을 태깅한 것을 사용했다. 표4는 6개의 도메인에 대한 문장의 분포이다.

표4. 도메인에 따른 말뭉치 분포

교통	날씨	시계	알람	일정	환율
355	603	246	658	895	215

본 논문에서는 학습데이터에 1925개의 문장을 사용했고, 테스트데이터는 1047개의 문장을 사용했다.

4.2 개체명 인식 결과

개체명 인식은 4개의 버전으로 성능을 비교했다.
표5는 개체명 인식에 대한 실험결과이다.

표5. 개체명 인식 성능

개체명인식 방법	문장 단위 정확도(Accuracy)
사전	86.72%
사전+규칙	93.50%
CRF	91.88%
하이브리드 (사전+CRF)	93.50%

표5에서 개체명 인식 성능은 사전규칙기반 시스템과 하이브리드기반 시스템이 높은 것을 확인할 수 있다. 하이브리드기반 시스템에서는 CRF에 개체명자질과 사전자질을 추가함으로써, CRF를 단독으로 사용한 방법보다 높은 성능을 낼 수 있음을 보였다. 서로 비슷한 의미를 지닌 문장은 같은 개체명 종류가 나올 것이라는 가정을 통해 구축한 개체명사전자질의 타당성을 실험을 통해 입증했으며, 하이브리드 방식에서 개체명인식 성능을 개선하는데 기여했음을 알 수 있다.

개체명 인식 방법 중 CRF자질은 형태소어휘/태그자질(1), 형태소어휘자질(2,3,4,9), 형태소태그자질(10,11,12,13,14,17), 어절 내 자질(18,19)을 사용했다.

하이브리드 방법은 형태소 어휘/태그자질(1), 형태소어휘자질(2,3,4,5,6,7,8,9), 형태소태그자질(10,11,12,13,14,15,16,17), 어절 내 자질(18,19), 사전자질(20), 개체명자질(21,22,23,24)을 사용해 성능을 평가했다.

4.3 문장목적 인식 결과

문장목적 인식 또한 4개의 버전으로 성능을 비교했다. 표6은 문장목적인식에 대한 결과이다.

표6. 문장목적 인식 성능

문장목적 인식방법	문장 단위 정확도(Accuracy)
사전	80.80%
사전+규칙	95.41%
CRF	99.71%
하이브리드 (사전+CRF)	99.31%

문장목적 인식은 사전규칙기반 시스템보다 CRF와 하이브리드기반 시스템을 수행한 결과가 더 높은 성능을 보이는 것을 결과를 통해 확인 할 수 있었다. 그리고 문장목적 인식은 문장목적사전자질을 추가했을 때보다 기본적인 자질만을 사용했을 때, 더 높은 성능을 보이는 것을 확인했다. 그 이유는 문장목적인식은 구간을 통해 문장의 목적을 식별하기 때문에, 완전히 같은 단어와 단어집합, 태그정보가 반복적으로 구간에 나타나는 경향이 많았다. 그런 이유로, 기본적인 자질을 사용했을 때 성능이 더 높은 것으로 분석되었다.

문장목적 인식 방법 중 CRF자질은 형태소어휘/태그자질(1), 형태소어휘자질(2,3,4,5,6,7,8,9), 형태소태그자질(10,11,12,13,14,15,16,17), 어절 내 자질(18,19)을 사용했다.

하이브리드 방법에서는 사전자질 및 문장목적자질을 추가 할 때 마다 인식성능이 떨어지는 것이 확인되었다. 그러므로 CRF에서 사용한 자질 중 사전자질(20)만을 추가한 결과를 사용해 성능을 평가했다.

4.4 개체명 문장목적 동시 인식 결과

개체명 문장목적 동시 인식방법은 3가지 버전으로 성능을 비교했다. 표7은 개체명 문장목적 동시인식에 대한 결과이다.

표7. 개체명 및 문장목적 동시 인식 성능

개체명 및 문장목적 인식방법	문장 단위 정확도(Accuracy)
사전+규칙	88.92%
CRF	91.50% (개체명 91.69%, 문장목적 98.80%)
하이브리드 (사전+CRF)	91.59% (개체명 93.12%, 문장목적 98.09%)

사전규칙기반 시스템은 파이프라인을 통해 수행되기

때문에 개체명과 문장목적 인식성능을 따로 기재하지 않고, 전체 성능만을 기재했다. 사전규칙기반 시스템은 오류가 축적됨에 따라 개체명 인식, 문장목적 인식을 각각 수행할 때 보다 문장 단위 정확도가 많이 떨어진 것을 확인할 수 있었다.

하이브리드기반 시스템에서는 기본적인자질과 사전자질을 사용해, 동시에 두 개의 작업을 수행함으로써, 사전규칙기반 시스템보다 2.67%의 성능이 향상된 것을 확인할 수 있었다. 개체명 문장목적 동시 인식 실험에서 확인할 수 있었던 것은 문장목적은 기본적인자질(시퀀스자질)이 높은 성능을 보이는데 반해, 개체명은 기본적인자질과 함께 개체명사전자질을 사용했을 때 더 높은 성능을 보였다. CRF를 단독으로 사용한 시스템에서는 개체명인식 성능은 상대적으로 낮은 성능을 보였지만, 문장목적 인식결과가 상당히 높은 성능을 보여, 문장 단위 성능이 높게 나온 것을 확인했다.

동시 인식 방법 중 CRF자질은 형태소어휘/태그자질(1), 형태소어휘자질(2,3,4,5,6,7,8,9), 형태소태그자질(10,11,12,13,14,15,16,17), 어절 내 자질(18,19)을 사용했다.

하이브리드 방법은 형태소어휘/태그자질(1), 형태소어휘자질(2,3,4,5,6,7,8,9),형태소태그자질(10,11,12,13,14,15,16,17), 어절 내 자질(18,19), 사전자질(20), 개체명자질(21,22,23,24)을 사용해 성능을 평가했다.

5. 결론

본 논문에서는 대화형 개인 비서 시스템에서 중요한 작업인 개체명 인식과 문장목적 인식을 동시에 수행하는 방법을 제안했다. 6개의 도메인에서 8개의 개체명과 28개의 문장목적을 대상으로 개체명과 문장목적 분류를 수행한 결과, 사전규칙기반의 파이프라인방식을 통해 두 인식을 수행한 성능보다 CRF와 사전자질을 이용해 하이브리드방식으로 두 인식을 동시에 수행한 결과가 문장단위 2.67%의 성능이 향상된 것을 확인 할 수 있었다.

향후 연구로는 문장목적구간이 학습데이터에서는 문장당 한 구간만 존재했지만, 서로 다른 목적을 가진 구간이 한 문장에 2개 이상 출현했을 경우 여러 목적구간 중 가장 확률이 높은 것을 선택하는 방법을 연구할 것이며, 대화형 개인 비서 시스템에서는 개체명인식과 문장목적 인식 이외에 형태소분석, 화행분석, 도메인인식 등 여러 절차가 존재하는데, 그 절차 중 도메인인식은 본 논문에서 제안한 방법과 결합이 가능 할 것으로 생각된다. 그러므로 개체명, 문장목적, 도메인 인식을 동시에 수행하는 방법을 연구할 것이다.

참고문헌

[1] Krupka, G.R. and Hausman, K. "Description of the netowl text extraction system as using for MUC-7." In proceedings of the Seventh Message Understanding Conference(MUC-7) 1998.

[2] 이경희, 이주호, 최명석, 김길창, “한국어 문서에서 개체명 인식에 관한 연구” 한국정보과학회 언어공학연구회 학술발표 논문집16 pp.40-45, 2004

[3] Mesfar, S. "Named Entity Recognition for Arabic Using Syntactic Grammars." 12th International Conference on Application of Natural Language to Information Systems, pp. 305-316, 2007.

[4] Nadeau, D. and Sekine, S. "A Survey of Named Entity Recognition and Classfication" Lingvisicae Investigationes, 30(1), pp.3-26, 2007.

[5] 이창기, 황인규, 오효정, 임수종, 허정, 이충희, 김현진, 왕지현, 장명길 “Conditional Random Fields를 이용한 세부 분류 개체명 인식” 한국정보과학회 언어공학연구회 학술발표 논문집, pp.268-272, 2006.

[6] Lafferty, J. McCallum, A.Pereira, F., "Conditional random fields : Probabilistic models for segmenting and labeling sequence data", ICML, pp.282-2289, 2001.

[7] Ratnaparkhi, A., "A Simple Introduction to Maximum Entropy Models for Natural Language Processing." University of Pennsylvania Institute for Research in Cognitive Science Technical Report No. IRCS-97-08, 1997.

[8] Petasis, G., Vichot, F., Wolinskim, F., Paliouras, G., Karkaletsis, V. and Spyropoulos, C. D. "Using Machine Learning to Maintain Rule-based Named-Entity Recognition and Classification Systems." Proceeding Conference of Association for Computational Linguistics, pp.426-433, 2001.

[9] Mai Mohamed Oudah, Khaled Shaalan "A Pipeline Aribic Named Entity Recognition Using a Hybrid Approach" COLING, pp.2159-2176, 2012.

[10] Kim Sang, E. F. T., de meulder, F., "Introduction to the CoNLL-2003 shared task : Language-independent named entity recognition" CoNLL 2003.

제25회 한글 및 한국어 정보처리 학술대회
25th Annual Conference on Human and Cognitive Language Technology

구두발표 1B

중간언어와 단어정렬을 통한 이중언어 사전의 자동 추출에 대한 성능 개선

권홍석○, 서형원, 김재훈
한국해양대학교, 컴퓨터공학과
hong8c@naver.com, wonn24@gmail.com, jhoon@kmou.ac.kr

Performance Improvement of Bilingual Lexicon Extraction via Pivot Language and Word Alignment Tool

Hong-Seok Kwon○, Hyeung-Won Seo, Jae-Hoon Kim
Korea Maritime and Ocean University, Department of Computer Engineering

요 약

본 논문은 잘 알려지지 않은 언어 쌍에 대해서 병렬말뭉치(parallel corpus)로부터 자동으로 이중언어 사전을 추출하는 방법을 제안하였다. 이 방법은 중간언어(pivot language)를 매개로 하고 문맥 벡터를 생성하기 위해 공개된 단어 정렬 도구인 Anymalign을 사용하였다. 그 결과로 초기사전(seed dictionary)을 사용한 문맥벡터의 번역 과정이 필요 없으며 통계적 방법의 약점인 낮은 빈도수를 가지는 어휘에 대한 번역 정확도를 높였다. 또한 문맥벡터의 요소 값으로 특정 임계값 이상을 가지는 양방향 번역 확률 정보를 사용하여 상위 5위 이내의 번역 정확도를 크게 높였다. 본 논문은 두 개의 서로 다른 언어 쌍 한국어-스페인어 그리고 한국어-프랑스어 양방향에 대해서 각각 이중언어 사전을 추출하는 실험을 하였다. 높은 빈도수를 가지는 어휘에 대한 번역 정확도는 이전 연구에서 보인 실험 결과에 비해 최소 3.41% 최대 67.91%의 성능 향상을 보였고 낮은 빈도수를 가지는 어휘에 대한 번역 정확도는 최소 5.06%, 최대 990%의 성능 향상을 보였다.

주제어 : Word alignment, Pivot language, Bilingual Lexicon Extraction, Parallel corpus

1. 서론

이중언어 사전은 기계 번역, 교차언어 정보검색 등 분야에서 중요한 자원으로 사용되고 있다. 이중언어 사전의 추출에 있어 가장 직접적인 방법으로는 병렬 말뭉치로부터 대역쌍(translation equivalence)을 추출하는 것이다[1]. 그러나 잘 알려지지 않은 언어 쌍에 대해서 병렬말뭉치를 모으는 일은 쉽지 않으며 특정 도메인에 제한되어 있다. 이러한 이유들 때문에 비교말뭉치(comparable corpus)로부터 이중언어 사전을 추출하는 연구[2][3][4]가 많이 이루어지고 있다. 그러나 잘 알려지지 않은 언어 쌍에 대해서는 비교말뭉치 또한 구축이 쉽지 않다. 이와 같은 문제를 해결하기 위해 중간언어를 매개로 이중언어 사전을 추출하는 연구들[5][6][7]이 있다.

다른 한편으로는 이중언어 사전 추출에 정보 검색 기법을 도입한 연구[8][9][10]도 있다. 정보 검색 기법을 사용한 방법은 간략하게 다음과 같다. 서로 다른 두 언어 L_1과 L_2 각각에 대해서 모든 어휘 단위들을 수집한다. 그리고 수집된 모든 어휘 단위에 대해서 문맥벡터 S와 T를 각각 생성하고 L_1과 L_2의 초기사전을 이용하여 각 문맥벡터 S와 T를 번역한다. 이때 초기사전은 사람에 의해서 수동으로 구축되어진 사전이며 그 용량이 클수록 정확한 번역이 가능해진다. 마지막으로 문맥벡터 S와 T의 유사도를 서로 비교하여 최종 번역 쌍을 추출한다.

본 논문에서는 잘 알려지지 않은 언어 쌍에 대해서 중간언어를 사용하고 정보검색 기법을 기반으로 이중언어 사전을 자동으로 추출하는 간단하고 효과적인 방법을 제안한다. 중간언어는 원시언어(source language)와 대상언어(target language)의 문맥벡터를 표현하는데 사용되고 정보검색 기법은 중간언어로 표현되어진 원시언어 문맥벡터와 대상언어 문맥벡터 사이의 유사도를 비교하는데 사용된다. 기존 연구와는 다르게 우리는 두 개의 병렬 말뭉치(예: 한국어-영어, 영어-스페인어)를 사용한다. 여기서 영어가 중간언어로서 사용된다. 그리고 우리는 문맥벡터를 쉽게 생성하기 위해 공개되어진 단어 정렬 도구인 Anymalign[11]을 사용한다.

우리가 제안한 방법은 다음과 같은 장점들이 있다. 첫째로 영어와 같은 중간언어를 사용하여 잘 알려지지 않은 언어 쌍에 대해서도 쉽게 적용이 가능하다. 둘째로 복합단어로 표현된 어휘에도 쉽게 확장이 용이하며 마지막으로 큰 규모의 초기사전을 수동으로 구축해야하는 수고스러움이 없다.

본 논문의 구성은 다음과 같다. 2장에서는 우리가 제안한 방법의 전체구성과 각 단계에 대해서 설명하고, 3장에서는 실험과 그 결과에 대해서 설명한다. 마지막으로 4장에서는 결론 및 향후연구에 대해서 논의한다.

2. 제안 방법

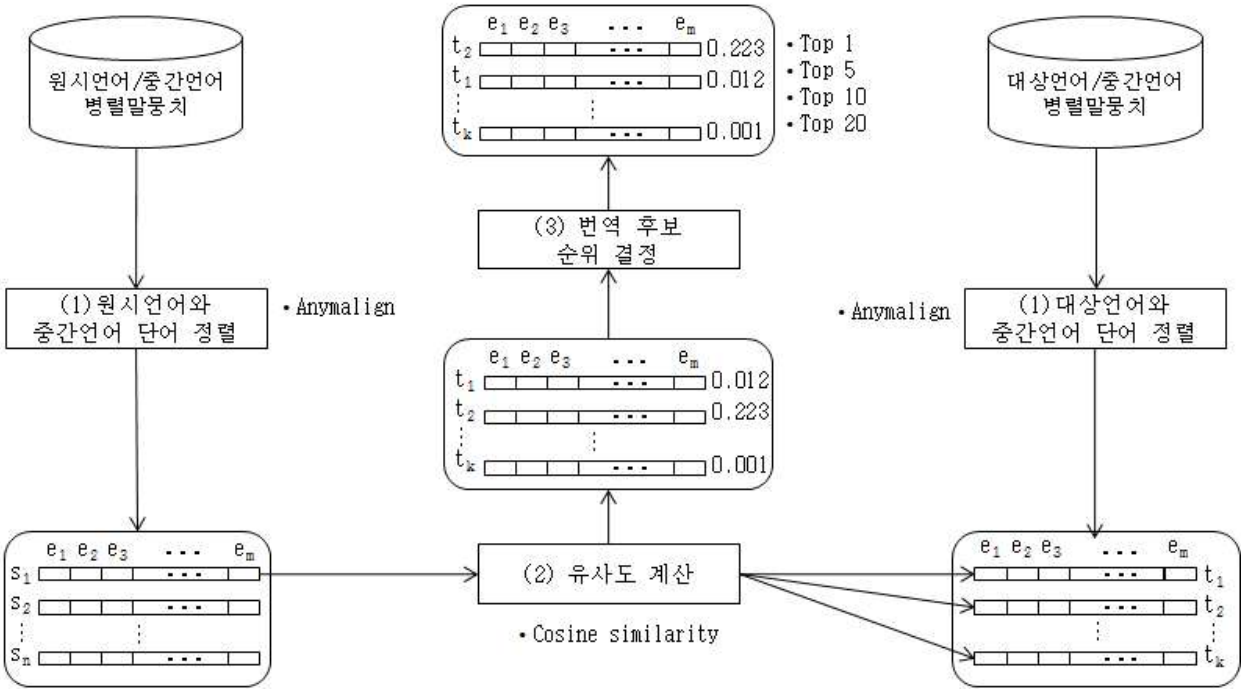


그림 1. 이중언어 사전 구축 전체 시스템 다이어그램

본 논문에서는 잘 알려지지 않은 언어 쌍에 대해서 중간언어와 정보검색 기법을 사용하여 간단하고 효과적으로 이중언어 사전을 추출하는 방법을 제안한다. 우리는 더 정확한 어휘 정렬 정보를 얻기 위해서 비교말뭉치가 아닌 병렬말뭉치를 사용한다. 그러나 잘 알려지지 않은 언어 쌍들에 대해서 병렬말뭉치를 모으는 일은 쉽지 않다. 이러한 이유들 때문에 우리는 잘 알려진 영어를 중간언어로써 사용한다.

중간언어는 원시언어와 대상언어의 문맥벡터를 표현하기 위해 사용된다. 비교말뭉치를 사용하는 기존 연구와는 다르게 우리는 두 종류의 한국어-영어 그리고 영어-스페인어 병렬말뭉치를 사용하며 중간언어로 표현되어진 원시언어 문맥벡터와 대상언어 문맥벡터의 유사도 비교에 정보검색 기법을 사용한다.

기존 연구에서는 큰 용량의 초기사전을 사용해 문맥벡터들을 번역하는 과정이 필요했다. 그러나 본 논문에서는 문맥벡터들을 번역하는 과정은 더 이상 필요하지 않다. 게다가 공개된 단어 정렬 도구를 사용하여 문맥벡터들을 생성했다. 제안한 방법의 전체적인 구조는 그림 1과 같다. 그리고 제안한 방법은 크게 세 단계로 설명 할 수 있다.

(1) 원시언어와 중간언어 단어 정렬

두 종류의 독립적인 병렬말뭉치(한국어-영어, 영어-스페인어)에서 원시언어(예. KR)와 대상언어(예. ES) 각각의 단어들에 대해 원시언어 문맥벡터와 대상언어 문맥벡터를 각각 생성한다. 여기서 문맥벡터의 요소가 되는 모든 단어(영어)들은 단어 정렬 도구인 Anymalign에 의해 가중치가 결정된다. Anymalign이 제공하는 정보의 예는 표1과 같다. $T(t|s)$는 원시단어가 주어졌을 때 대상단어로 번역될 확률이며 $T(s|t)$는 대상단어가 주어졌을 때 원시단어로 번역될 확률을 의미한다. 본 논문에서는 문맥벡터의 요소로써 Anymalign이 제공하는 모든 단어 정렬 정보를 사용하지 않고, 양방향 번역확률 $T(t|s)$, $T(s|t) \geq x$ 특정 임계값($0 \leq x \leq 1$) 이상이고, 양방향 번역확률의 차 $|T(t|s) - T(s|t)| \geq 0.5$일 경우 올바른 단어 정렬 정보라고 판단하고 $\frac{T(t|s)+T(s|t)}{2}$를 가중치로 사용하였다. 이러한 이유는 문맥벡터의 잡음을 줄이기 위해서이다. 아래 표 1의 예를 보면 올바른 단어 정렬 정보는 양방향 번역확률의 차이가 크지 않고 번역확률 또한 높은 것을 확인할 수 있다. 그러나 잘못된 정렬 정보의 경우 양방향 번역확률의 차이가 크고 어느 한쪽의 번역확률이 낮거나 양방향 번역확률이 모두 낮은 것을 볼 수 있다.

표 1. Anymalign을 통한 단어 정렬 결과

원시단어	대상단어	$T(t\|s)$	$T(s\|t)$	빈도수
경찰	police	0.890421	0.877882	389447
경찰	scenario	1.000000	0.000167	8
정부	balance	0.000004	0.000506	11

(2) 유사도 계산

모든 원시언어 문맥벡터와 모든 대상언어 문맥벡터들 사이의 유사도를 계산한다. 여기서 유사도 비교에 코사인 유사도(cosine similarity)를 사용한다.

(3) 번역 후보 순위 결정

계산된 코사인 유사도의 값에 따라 번역 후보 중 상위 k

개의 단어 쌍을 추출한다.

본 논문에서는 문맥벡터를 생성하기 위해 중간언어(영어)를 공유하는 두 개의 병렬말뭉치를 사용했다. 그 이유는 공개된 한국어-스페인어의 병렬말뭉치가 없었기 때문이다. 그리고 공개 단어 정렬도구인 Anymalign은 문맥벡터의 모든 단어들에 가중치를 결정하는데 사용되었다. Anymalign은 무작위 샘플링과 문자열 차이를 기반으로 단어들을 정렬하며, 낮은 빈도수를 가지는 단어들에 대해서 정확한 번역 후보들을 추출하는데 높은 정확도를 보였다.

앞서 언급했듯이 기존 연구에서는 원시언어와 대상언어의 문맥벡터들을 초기사전을 이용하여 번역하는 과정이 필요했으나 본 논문이 제안하는 방법에서는 더 이상 필요하지 않다. 또한 양방향 번역정보를 활용하고 이전 연구[12]에서 보인 불용어들이 번역후보로 추출되는 문제점을 품사태깅 후 제거함으로서 해결하였다. 그리고 문맥벡터들이 생성되고 나면 모든 원시언어 문맥벡터들과 모든 대상언어 문맥벡터들 사이 코사인 유사도를 통해 두 문맥벡터 간의 유사도가 측정된다. 마지막으로 유사도가 큰 순으로 정렬하고 상위 k개의 단어 쌍을 번역후보로 추출한다.

3. 실험 및 결과

본 논문에서는 두 개의 서로 다른 언어 쌍인 한국어(KR)-스페인어(ES) 그리고 한국어-프랑스어(FR)에서 명사들을 대상으로 임계값 x에 따른 이중언어 사전 추출 실험을 하였고 이전 연구(양방향 번역확률, 양방향 번역확률의 차를 고려하지 않은 방법)와 그 성능을 비교 실험하였다.

3.1. 실험 설정

3.1.1. 병렬 말뭉치

본 논문에서 사용된 병렬말뭉치로는 뉴스기사로부터 수집된 433,151 문장으로 구성된 한국어-영어[13] 병렬말뭉치와 Europarl[14] 병렬말뭉치로부터 무작위로 500,000 문장을 추출한 스페인어-영어 그리고 프랑스어-영어 병렬말뭉치이다. 각 언어에서 하나의 문장이 구성하고 있는 단어의 수는 표 2와 같다. 표 2를 보면 ES-EN과 FR-EN에서 각 언어의 문장 당 단어의 개수는 비슷하나 KR-EN의 경우는 조금 차이가 있다는 것을 볼 수 있다. 그 이유는 영어와는 다르게 한국어의 경우 하나의 단어가 한 개 또는 그 이상의 형태소로 구성되기 때문이다. 만약 형태소의 개수가 아니라 단어의 개수를 세면 한 문장 당 단어의 수는 비슷할 것이다.

표 2. 문장 당 평균 단어의 수

KR-EN		ES-EN		FR-EN	
KR	EN	ES	EN	FR	EN
31	19.2	26.4	25.4	29.7	27.1

3.1.2. 데이터 전처리

각 언어들은 다음과 같은 도구에 의해서 토큰이 분리되었다. 한국어의 경우 창원대학교 형태소분석기 Espresso[15]를 사용하여 토큰을 분리하고 품사 태깅을 하였다. 영어, 스페인어 그리고 프랑스어는 Tree-Tagger[16]를 사용하여 토큰이 분리되고 원형 복원되었다. 또한 모든 언어에서 명사를 제외한 품사들은 문장에서 제거되었다. 그 이유는 앞서 언급했듯이 Anymalign은 단어 정렬 시에 문맥을 고려하는 것이 아니라 무작위 샘플링과 문자열의 차이를 이용하기 때문에 본 논문에서의 이중언어 사전 구축대상인 명사를 제외한 품사들은 단어 정렬 시에 잡음으로 작용할 수 있기 때문이다.

3.1.3. 평가사전의 구축

본 논문이 제안하는 방법의 성능을 평가하기 위해 우리는 네 집합(KR-ES, KR-FR, ES-KR, FR-KR)의 평가사전을 인터넷 사전[1]으로부터 수동으로 구축하였다. 각 평가사전은 단방향으로 구축되어있으며, 즉 한 언어에서 다른 언어로의 번역을 말하며, 말뭉치 내에서 가장 빈도수가 높은 100개의 명사(HIGH)와 가장 빈도수가 낮은 100개의 명사(LOW)들로 구성되어 있다. 표 3는 각 평가사전에서 하나의 명사에 대한 평균 번역 수를 나타낸 것이다. 여기서 평균 번역 수는 그 명사의 모호성 정도를 의미한다고 볼 수 있다.

표 3. 평가사전에서의 평균 번역단어의 수

평가사전	HIGH	LOW
KR-FR	5.79	2.26
KR-ES	7.36	3.12
ES-KR	10.31	5.49
FR-KR	10.42	6.32

3.1.4. 평가 방법

제안한 방법의 이중언어 사전 구축 성능을 평가하기 위해서 정보검색의 평가방법과 유사하게 정확도(accuracy)와 MRR(Mean Reciprocal Rank)[17]을 이용하였다. 정확도는 모든 평가 단어에 대하여 평가기준 상위 k개 이내에 정답이 적어도 한 개 있는 평가 단어들의 수의 조화평균을 구한 것이며, MRR은 모든 평가 단어에 대해 평가기준 상위 k개 이내에서 처음으로 나온 정답순위의 역순을 구하고 그것들의 조화평균을 구한 것이며 번역후보결과의 순위화 성능을 평가하기 위해 이용되었다.

3.2. 결과

본 논문에서는 임계값 x를 변경해 가면서 이중언어 사전 추출 실험을 하였고, 표 5는 각 언어쌍에서 이중언어 사전을

1) dic.naver.com

표 4. 평가사전 HIGH와 LOW에 대한 정확도 결과

HIGH								
	accu.Top 1		*accu.Top 5*		*accu.Top 10*		*accu.Top 20*	
	이전연구	제안방법	이전연구	제안방법	이전연구	제안방법	이전연구	제안방법
KR-ES	0.577	0.683(↑18.37%)	0.718	0.704(↓1.94%)	0.758	0.714(↓5.80%)	0.778	0.724(↓6.94%)
KR-FR	0.487	0.707(↑45.17%)	0.703	0.727(↑3.41%)	0.738	0.727(↓1.49%)	0.763	0.727(↓4.71%)
ES-KR	0.293	0.492(↑67.91%)	0.751	0.743(↓1.06%)	0.804	0.779(↓3.10%)	0.872	0.825(↓5.38%)
FR-KR	0.282	0.459(↑62.76%)	0.736	0.709(↓3.66%)	0.789	0.775(↓1.77%)	0.842	0.790(↓6.17%)

LOW								
	accu.Top 1		*accu.Top 5*		*accu.Top 10*		*accu.Top 20*	
	이전연구	제안방법	이전연구	제안방법	이전연구	제안방법	이전연구	제안방법
KR-ES	0.244	0.324(↑032.78%)	0.449	0.492(↑09.57%)	0.469	0.555(↑18.33%)	0.489	0.575(↑17.58%)
KR-FR	0.164	0.269(↑064.02%)	0.299	0.543(↑81.60%)	0.412	0.624(↑51.45%)	0.463	0.680(↑46.86%)
ES-KR	0.030	0.297(↑990.00%)	0.299	0.459(↑53.51%)	0.391	0.459(↑17.39%)	0.433	0.459(↑06.00%)
FR-KR	0.062	0.251(↑404.83%)	0.260	0.415(↑59.61%)	0.395	0.415(↑05.06%)	0.489	0.415(↓15.13%)

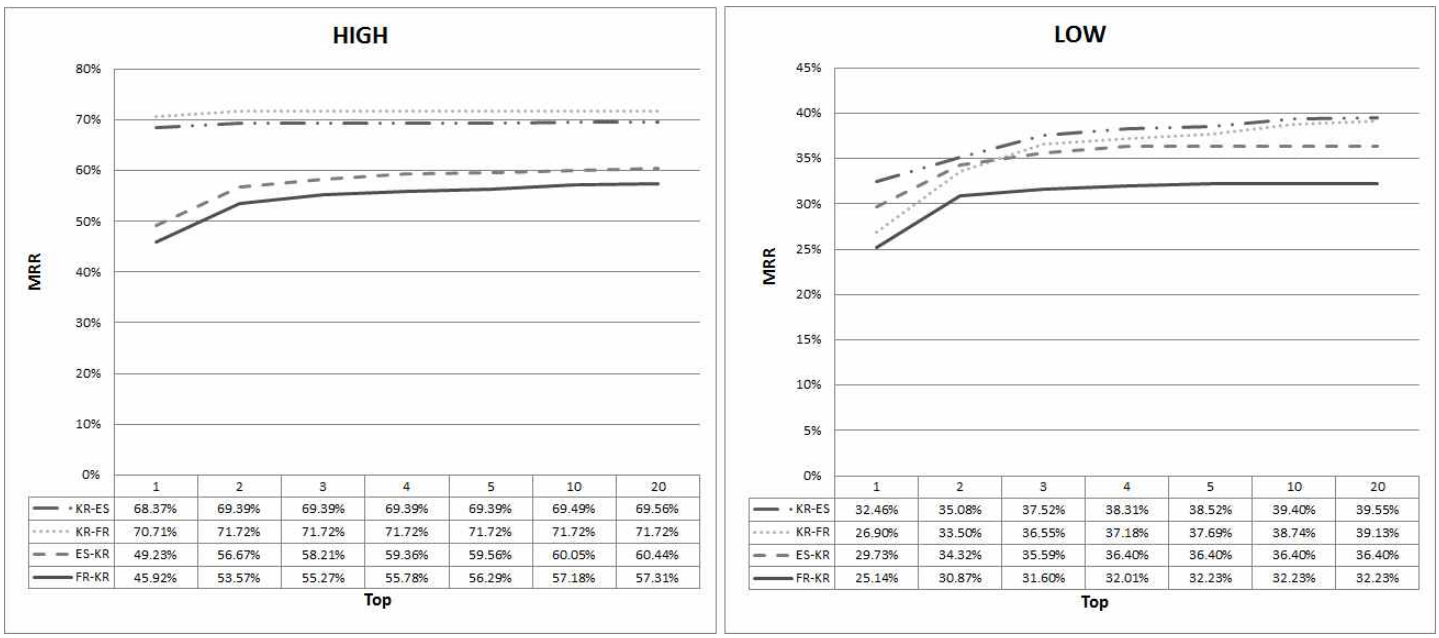


그림 2. 평가사전 HIGH와 LOW에 대한 MRR 결과

추출할 때 가장 좋은 성능을 보여준 임계값 x를 보여준다.

표 5. 각 평가사전에서 사용한 임계값 x

평가사전	HIGH	LOW
KR-FR	0.1	0.00005
KR-ES	0.1	0.00005
ES-KR	0.01	0.05
FR-KR	0.01	0.05

평가단어 HIGH와 LOW에 대한 정확도는 표 4와 같다. 양방향 번역확률을 고려한 결과 평가사전 HIGH에서는 최상위에서 최대 67.91%의 성능 향상을 얻을 수 있었다. 그러나 상위 5위를 넘어서면서 상위 20위에서 최대 6.94%의 성능 하락이 있었다. 그 이유는 상위 5위 이내의 성능을 향상시키기 위해 임계값 x를 높게 설정하면서 문맥벡터의 요소가 되는 단어 정렬정보들이 많이 소실되었기 때문이다. 같은 실험으로 평가사전 LOW에서는 최상위에서 최대 990%의 성능 향상이 있었고 상위 10위에서 최소 5.06%의 성능 향상을 얻을 수 있었다.

표에서도 볼 수 있듯이 최상위에서 KR에서 ES와 FR로 번역되는 정확도에 비해 그 역방향인 ES-KR과 FR-KR의 번역 정확도가 낮은 것을 볼 수 있다. 그 이유를 분석해 보면 한국어의 경우 다른 언어들과는 다르게 형태소 분석과정이 필요하고 또한 앞서 표 2에서 보았듯이 한국어의 문장 당 단어의 수가 영어의 문장 당 단어 수보다 많다는 것에 있다. 즉 다른 언어에서 한국어로의 번역이 한국어에서 다른 언어로의 번역보다 더 모호하다는 것이다.

표 6는 'exercice(운동)'의 상위 5위의 번역후보를 보여주는 예이다. 이전 연구에서의 문제점으로 불용어들이 최종 번역후보로 추출되는 문제점들은 앞서 언급했듯이 데이터 전처리 과정을 통해 해결되었다. 그러나 최상위와 3위를 보면 형태소분석과 품사태깅의 오류로 인한 잘못된

번역 후보들이 추출되는 문제점들을 발견할 수 있었다.

표 6. 'exercice(훈련)'의 상위 5위 이내의 번역 후보의 예

Rank	원시언어	대상언어	유사도
1	exercice	훈련에	0.350
2	exercice	운동	0.281
3	exercice	중요하다	0.265
4	exercice	합동	0.256
5	exercice	군사훈련	0.145

평가단어 HIGH와 LOW에 대한 MRR은 그림 2와 같다. 그림에서 볼 수 있듯이 HIGH에서의 MRR은 상위 2위까지 급격히 증가하다가 점차 증가하는 것을 볼 수 있으며 LOW에서의 MRR은 상위 3위까지 급격히 증가하다가 그 이후로는 점차 증가하는 것을 확인 할 수 있다. 이는 본 논문이 제안하는 방법을 사용했을 때 HIGH에서는 상위 2위 이내에서 대부분의 올바른 번역후보들이 추출된다는 것을 의미하며 LOW에서는 상위 3위 이내에서 대부분의 올바른 번역후보들이 추출된다는 것을 의미한다.

4. 결론

본 논문에서는 정보 검색 기법을 기반으로 중간언어를 사용하여 이중언어 사전을 병렬말뭉치로부터 자동 구축하는 방법을 제안하였다. 우리는 이전 연구에서 보인 문제점을 데이터 전처리를 통해 해결하였고 양방향 번역확률 정보를 고려하여 상위 5위 이내의 정확도를 크게 향상시켰다. 향후 연구로는 스페인어와 프랑스어 이외에 다른 언어에 대해서도 이중언어 사전을 자동으로 구축해 보는 것과 복합단어(Multi-word expression)으로 확장해 나가는 연구를 해볼 것이다.

감사의 글

본 연구는 미래창조과학부 및 한국산업기술평가관리원의 산업융합원천기술개발사업(정보통신)의 일환으로 수행하였음. [10041807, 지식학습 기반의 다국어 확장이 용이한 관광/국제행사 통역률 90%급 자동 통번역 소프트웨어 원천 기술 개발]

참고문헌

[1] D. Wu and X. Xia. 1994. Learning an English-Chinese lexicon from a parallel corpus. In Proceedings of the First Conference of the Association for Machine Translation in the Americas (AMTA 1994, Columbia, Maryland, USA, October), pages 206-213.

[2] P. Fung. 1995. Compiling bilingual lexicon entries from a non-parallel English-Chinese corpus. In Proceedings of the Third Workshop on Very Large Corpora (VLC' 95), pages 173-183.

[3] K. Yu and J. Tsujii. 2009. Bilingual dictionary extraction from Wikipedia. In Proceedings of the 12th Machine Translation Summit (MTS 2009), Ottawa, Ontario, Canada.

[4] A. Ismail and S. Manandhar. 2010. Bilingual lexicon extraction from comparable corpora using in-domain terms. In Proceedings of the International Conference on Computational Linguistics, pages 481-489.

[5] K. Tanaka and K. Umemura. 1994. Construction of a Bilingual Dictionary Intermediated by a Third Language. In Proceedings of the 15th International Conference on Computational Linguistics (Coling' 94), Kyoto, Japan, August, pages 297-303.

[6] H. Wu and H. Wang. 2007. Pivot Language Approach for Phrase-Based Statistical Machine Translation. In Proceedings of 45th Annual Meeting of the Association for Computational Linguistics, pages 856-863.

[7] T. Tsunakawa, N. Okazaki, and J. Tsujii. 2008. Build-ing Bilingual Lexicons Using Lexical Translation Probabilities via Pivot Languages. In Proceedings of the International Conference on Computational Linguistics, Posters Proceedings, pages 18-22.

[8] P. Fung. 1998. A statistical view on bilingual lexicon extraction: from parallel corpora to non-parallel corpora. In Proceedings of the Parallel Text Processing, pages 1-16.

[9] E. Gaussier, J.-M. Renders, I. Matveeva, C. Goutte and H. Dejean. 2004. A geometric view on bilingual lexcion extraction from comparable corpora. In Proceedings of the 42th Annual Meeting of the Association for Computational Linguistics, Barcelona, Spain, pages 527-534.

[10] A. Hazem and E. Morin. 2012. Adaptive dictionary for bilingual lexicon extraction from comparable corpora. In Proceedings of the 8th International Conference on Language Resources and Evaluation (LREC'12), pages 288-292.

[11] A. Lardilleux, Y. Lepage, and F. Yvon. 2011. The contribution of low frequencies to multilingual sub-sentential alignment: a differential associative approach. International Journal of Advanced Intelligence, 3(2):189-217.

[12] H. Kwon, H. Seo and J. Kim, 2013. Bilingual Lexicon Extraction via Pivot Language and Word Alignment Tool. In Proceedings of the Sixth Workshop on Building and Using Comparable Corpora, Sofia, Bulgaria, August 8, 2013, pages 11-15.

[13] H. Seo, H. Kim, H. Cho, J. Kim and S. Yang, 2006. Automatically constructing English-Korean parallel corpus from web documents. Korea

Information Proceedings Society, 13(2):161-164.
[14] P. Koehn. 2005. EuroParl: A parallel corpus for statistical machine translation. In proceedings of the Conference on the Machine Translation Summit, page 79-86.
[15] J. Hong and J. Cha, 2008. A New Korean Morphological Analyzer using the Eojeol Pattern Dictionary. In Proceedings of the Korea Computer Congress, pages 279-283.
[16] H Schmid. 1994. Probabilistic part-of-speech tagging using decision trees. In Proceedings of International Conference on New Methods in Language Processing, Manchester, UK, pages 44-49.
[17] E. Voorhees. 1999. The TREC-8 Question Answering Track Report. In 8th Text Retrieval Conference (TREC-8), pages 77-82.

준지도학습을 통한 세부감성 어휘 구축

조요한O, 오효정, 이충희, 김현기
한국전자통신연구원
yohan.jo@etri.re.kr, ohj@etri.re.kr, forever@etri.re.kr, hkk@etri.re.kr

Fine-grained Sentiment Lexicon Construction via Semi-supervised Learning

Yo-han JoO, Hyo-Jung Oh, Chung-Hee Lee, Hyun-ki Kim
Electronics and Telecommunications Research Institute

요 약

소셜미디어를 통한 여론분석과 브랜드 모니터링에 대한 요구가 증가하면서, 빅데이터로부터 감성을 분석하는 기술에 대한 필요가 늘고 있다. 이를 위해, 본 논문에서는 단순 긍/부정 감성이 아닌 20종류의 세분화된 감성을 분석하기 위한 감성어휘 구축 알고리즘을 제시한다. 감성어휘 구축을 위해서는 준지도학습을 사용하였으며, 도메인에 특화되지 않은 일반 감성어휘를 구축하도록 학습되었다. 학습된 감성어휘를 인물, 스마트기기, 정책 등 다양한 도메인의 트위터 데이터에 적용하여 세부감성을 분석한 결과, 알고리즘의 특성상 재현율이 낮다는 한계를 가지고 있었으나, 대부분의 감성에 대해 높은 정확도를 지닌 감성어휘를 구축할 수 있었고, 감성을 직간접적으로 나타내는 표현들을 학습할 수 있었다.

주제어: 감성 분석, 오피니언 마이닝, 감성 어휘

1. 서론

트위터, 페이스북, 블로그 등의 소셜미디어 사용량이 증가하면서, 이런 빅데이터로부터 자동으로 오피니언 정보를 분석하려는 시도가 활발히 이루어지고 있다. 가령, 기업은 자사 제품이나 서비스가 소셜미디어 상에서 어떤 평가를 받고 있는지 파악함으로써 마케팅 전략에 참고할 수 있고, 정책 기관에서는 정책에 대한 여론분석을 통해 정책 수정방향 및 홍보방식 등을 결정할 수 있다. 이러한 필요가 대두되면서 빅데이터 기반 오피니언 마이닝을 전문으로 하는 브랜드 모니터링 서비스 사업도 활발해지고 있다.

본 논문에서는, 빅데이터 기반의 브랜드 모니터링 서비스를 위한 감성분석을 위해 감성어휘를 구축하는 기법을 제시한다. 그동안 학계에서 제시되어 온 감성분석 기법들은 각기 장단점을 가지고 있으며, 사용 목적에 따라 적절한 기법을 선택할 필요가 있다. 본 논문에서 초점을 맞추고 있는 브랜드 모니터링 서비스를 위해서는 다음과 같은 조건을 만족시키는 감성분석 기법이 필요하다.

먼저 단순 극성(긍정/부정)감성이 아닌 더 세분화된 세부감성을 제공할 수 있어야 한다. 기존의 감성분석 기법들이 주로 다루는 극성감성은 정보력에 한계가 있다. 가령, 어떤 기업의 제품에 대해 '걱정'하는 여론이 주를 이룬다면 기업에서는 소비자를 안심시킬 수 있는 홍보를 기획하는 반면, '불만' 여론이 주를 이룬다면 소비자들이 제품에 대해서 불만을 갖는 부분을 개선하는 방향으로 전략을 세울 수 있을 것이다. 하지만 '걱정'과 '불만'은 모두 부정적인 여론이므로, 극성감성 분석만 가지고는 이와 같은 구체적인 전략을 세우기가 어렵다.

또한, 감성분석을 통해 텍스트에서 찾아낸 감성에 대해, 사람이 납득할 만한 단서를 제시할 수 있어야 한다. 그 이유는, 브랜드 모니터링 서비스를 이용하는 사용자들은 서비스의 신뢰도 파악을 위해 서비스에서 제공하는 감성분석 결과에 대한 판단 근거를 요구하기 때문이다. 단서를 제공하기 위해서는 어휘(lexicon) 기반의 감성분석이 적합하다. 가령, 지지벡터기계(SVM: Support Vector Machine)[1]처럼 여러 자질들의 조합을 통해 감성을 분류하는 방식으로는 감성 판단의 근거를 사용자들에게 납득시키기가 쉽지 않다. 반면 어휘를 기반으로 하는 경우, 감성 판단의 근거가 되는 단어 혹은 자질을 제시함으로써 사용자들을 이해시키기가 비교적 용이하다.

다음으로, 높은 정확도(precision)가 높은 재현율(recall)보다 우선순위에 있어야 한다. 서비스 사용자들은 감성분석 된 결과량이 많지 않을지언정 틀린 결과를 보고 싶어 하지 않는다. 게다가 재현율이 낮은 단점은 빅데이터 분석 결과를 활용함으로써 보상 가능하다. 일반적으로 어휘 기반의 감성분석 방식은 다른 기계학습 방식에 비해 재현율이 낮다는 단점을 가지고 있는데, 왜냐하면 기본적으로 어휘에 들어가는 자질은 사용자에게 보여지기 위한 것이고, 따라서 정확도가 높은 자질들 위주로 들어가기 때문이다.

마지막으로, 너무 많은 양의 학습데이터 구축을 필요로 해서는 안된다. 브랜드 모니터링 서비스 사용자들이 요구하는 다양한 도메인을 반영하는 어휘가 모두 포함되도록 학습데이터를 구축하기에는 너무 많은 비용과 노력이 수반된다. 따라서 본 논문에서는 준지도학습(semi-supervised learning) 방식을 취함으로써, 태깅된 데이터와 태깅되지 않은 데이터를 모두 사용하여 학습한다.

이후 장들에서는 위의 네 가지 조건을 전제로 하여, 한국어 트윗으로부터 세부감성 분석을 위한 어휘를 구축하는 알고리즘을 제시하고 평가 결과를 제시한다. 2장에서는 관련 연구를 살펴보고, 3장에서는 구체적인 알고리즘을 제시하며, 4장에서는 평가데이터 및 평가척도에 대해 명시하고, 5장에서는 평가 결과를 분석하여, 6장에서 이 논문의 결론을 지으며 마무리 할 것이다.

2. 관련 연구

세부감성을 분석하는 연구는 크게 벡터표현 기반, 매트릭스 기반, 그래프 기반의 연구가 이루어졌다. 먼저 벡터표현 기반의 연구[2]에서는 ExperienceProject.com에 있는 고백에 관한 글들을 대상으로 다섯 종류의 감성을 분석한다. 모든 단어와 감성은 벡터로 표현되고, 문서의 감성은 문서에 들어있는 단어벡터와 감성벡터의 내적으로 정의된다. 이 방식에서는 지도학습을 통해 단어벡터와 감성벡터를 계

산하며, 태깅된 데이터에 존재하지 않는 단어에 대해서는 단어벡터를 계산할 수 없다. 매트릭스 기반의 연구[3]에서는 뉴스 헤드라인에 대하여 네 종류의 감성을 분석한다. 문서와 감성을 매트릭스로 표현하고 매트릭스 분해(matrix factorization)를 통해 감성을 분석하는데, 역시 태깅된 데이터에 존재하지 않는 단어에 대해서는 계산이 불가능하다. 그래프 기반의 연구[4]는 본 논문과 가장 유사한 접근 방식을 취한다. 단어들을 꼭지점으로 나타내고 단어들 간의 관계를 변으로 나타낸 뒤, 감성을 지닌 단어들로부터 감성이 알려지지 않은 단어들로 감성을 확장시킨다. 감성을 모르는 대량의 단어들에 대해서 학습할 수 있다는 장점이 있으나, 원래 논문에서는 긍정/부정 두 종류의 극성감성만을 사용하였다. 그밖에도 일반적으로 널리 사용되는 Structural SVM[5]등을 사용해 세부감성을 분류할 수 있으나, 역시 태깅된 데이터로만 학습할 수 있다는 한계가 있다.

감성	포함 감성 예	감성	포함 감성 예
자신감	자신감	두려움	공포 무서움 걱정 불안
감동	감탄 존경 경의	화남	격노 분노 억울함 짜증
감사	감사	싫어함	미움 증오 혐오 지루함 심심함 싫증
기대감	설렘 희망 기대됨	슬픔	연민 애처로움 쓸쓸함 참담함 비참함 안타까움
좋아함	호감 사랑	실망	체념 아쉬움 그리움 허무 어이없음 황당함 후회
기쁨	즐거움 흥미 행복함 반가움 만족	수치심	창피 무안 부끄러움 민망
안심	위안 안심 안도	곤란	서먹 어색 거북함 곤란 난처
신뢰	신뢰 믿음	미안함	미안함
선의	인사말 축하 응원 격려	부러움	질투 시기
의심	불신	반대	비동의 비납득

표 1 세부감성 및 포함되는 감성의 예

3. 방법

3.1 감성 분류 체계

본 논문에서는 세부감성을 위해, [6]에서 제안하는 감성분류체계를 기반으로 한다. 이 감성분류체계는 기존의 심리학에서 사용되는 감성체계[7]를 소셜 웹 미디어에 적합하도록 수정한 19개의 카테고리로 구성되어 있다. 본 논문에서는 기본적으로 이 분류체계를 사용하되, 서비스에 적절하도록 다음과 같은 수정을 하였다.

- 세부감성이 애매한 경우를 위해 만들어 놓았던 '나쁨(bad)', '좋음(good)' 제거
- 사전적 정의에 따라 '두려움'과 '걱정' 병합
- 소셜미디어에 많은 '의심', '수치심', '곤란' 감성 추가
- 출현 빈도가 높은 '감사'는 '감동'으로부터 분리
- 쓰임새의 특성상 '안심'을 '만족'으로부터 분리하고, '만족'은 '기쁨'에 병합
- '인정'에서 감성이 불분명한 '이해', '납득', '동감'을 제거

이렇게 수정된 세부감성 분류체계가 표 1에 있다.

3.2 감성어휘 구축 모델

본 논문에서 제시하는 모델은 [4]의 모델을 응용한다. 이는 그래프 기반의 알고리즘으로서, 원래 논문에서는 극성 감성어휘를 구축하기 위해 사용되었다. 감성어휘에 들어갈 후보들이 꼭지점이 되고, 후보들 간의 상호정보량(mutual information)이 변이 되며, 극성이 이미 태깅된 꼭지점이 몇 개 있다. 기본적인 아이디어는 태깅된 꼭지점과의 상호정보량이 큰 꼭지점들은 비슷한 강도의 극성을 갖고, 반면 상호정보량이 작은 꼭지점들은 낮은 강도의 극성을 갖도록 그래프 상에서 극성값을 확산시키는 것이다.

구체적인 알고리즘을 설명하기에 앞서, 본 논문에서 특별히 그래프 확산 알고리즘을 선택한 이유는 다음과 같다. 대부분의 지도학습은 태깅된 학습데이터에 나타나는 자질 외에는 학습이 어려운데, 빅데이터에서 모든 중요한 자질을 포함하는 태깅 데이터를 구축하기는 쉽지 않다. 즉, 트위터와 같은 빅데이터를 충분히 수용할 수 있을 정도의 감성어휘를 위한 학습데이터를 구축하는 것은 현실적으로 어렵기 때문에 준지도학습 방식이 적합하다. 또한, 지도학습은 하나의 학습데이터 포인트에 대해 카테고리 수만큼의 true/false가 태깅되어 있어야 하는 것에 반해, 그래프 확산 알고리즘에서는 하나의 데이터 포인트에 대해 false 정보를 태깅할 필요가 없어서 학습데이터 구축이 용이하다. 특히 본 논문에서처럼 20 종류의 감성을 사용하는 경우, 지도학습 방식에서처럼 하나의 트윗에 대해 20개의 true/false를 태깅하는 것보다, 각 감성별로 해당하는 단어를 독립적으로 수집하는 것이 훨씬 효율적이다.

3.2.1 그래프 구축

본 알고리즘에서 그래프 $G=(V,E)$를 구축하고 감성어휘를 구축하는 방법을 구체적으로 설명하도록 한다. 먼저 감성집합 $S=\{s_1,\cdots,s_M\}$이 있고, M은 감성 종류의 개수이다. 각 감성에 대해서는 학습을 위한 기본감성어휘 $L_k^S, 1 \leq k \leq M$가 있다. 임의의 학습데이터 트윗 d는 다음과 같은 자질 집합으로 나타낼 수 있다.

$$d = F_d \cup \{LABEL(s_k) : s_k\text{는 } d\text{에 들어있는 감성}\}$$

F_d는 트윗 d에 있는 형태소나 단어 등 학습을 통해 감성어휘에 들어가게 될 자질들의 집합이다. 어떤 트윗 d가 L_k^S에 속하는 자질을 적어도 하나 이상 포함하는 경우, s_k가 d에 들어있다고 보고, 감성 s_k의 감성값 $LABEL(s_k)$을 하나의 자질로 간주한다. 한편, 준지도학습에서는 태깅이 되어있지 않은 트윗들도 학습에 함께 사용된다. 이런 트윗들에 대해서는 레이블 자질이 존재하지 않는다.

G의 꼭지점 집합 $V = V^F \cup V^L$에 대해서, 꼭지점 $v \in V^F$는 감성어휘에 들어갈 후보 자질을 나타내고 V^L은 $\{LABEL(s_k) : 1 \leq k \leq M\}$을 나타낸다. 임의의 꼭지점 v_i가 나타내는 자질을 f_i라고 할 때, 꼭지점 $v_i, v_j \in V$를 연결하는 변 $e_{ij} \in E$에 대해 PMI h_{ij}를 다음과 같이 정의한다.

$$\begin{aligned} h_{ij} &= PMI(f_i, f_j) \\ &= \log \frac{p(f_i, f_j)}{p(f_i)p(f_j)} \\ &= \log \frac{n(f_i, f_j)/N}{n(f_i)/N \cdot n(f_j)/N} \\ &= \log \frac{n(f_i, f_j)N}{n(f_i)n(f_j)} \end{aligned}$$

여기에서 $n(f_i, f_j)$는 f_i와 f_j가 모두 등장하는 트윗의 개수, $n(f_i)$는 f_i가 등장하는 트윗의 개수, N은 모든 트윗 개수이고, h_{ij}의 범위는 다음과 같다.

$$-\infty \leq h_{ij} \leq \min\{\log(N/n(f_i)), \log(N/n(f_j))\}$$

e_{ij}의 무게값에 대해서는, f_i와 f_j의 PMI 값이 0보다 작을 경우 무게값을 0으로 설정하고, 0 이상일 경우 h_{ij}를 [0,1]로 정규화시킨 값을 무게값으로 사용한다. 즉, e_{ij}의 무게값 w_{ij}는 다음과 같

이 정의된다.

$$w_{ij} = \begin{cases} 0 & \text{if } h_{ij} < 0 \\ h_{ij}/\log(N) & \text{otherwise} \end{cases}$$

그래프 상에서 감성이 확산되는 방식은 다음과 같다. 임의의 꼭지점 $v \in V$는 모든 감성에 대해 감성강도 $I_k, 1 \leq k \leq M$를 가지고 있다. 감성 레이블 꼭지점 $v \in V^L$는 그 레이블이 의미하는 감성 s_k에 대해 $I_k = 1$의 값을 갖는다. 임의의 꼭지점 $v_i \in V$와 인접한 꼭지점 $v_j \in V$에 대해 v_i의 감성강도 I_{ik}는 다음과 같이 계산된다.

$$I_{ik} = \max_j \{I_{jk} \cdot w_{ij}\}$$

각 $v \in V_L$에 대해, v와 연결된 모든 꼭지점들에 대해 I를 계산하고, 그 꼭지점들과 연결된 꼭지점들에 대해서 또 I를 계산을 하는 식으로 확산해 나간다. 초기에는 감성 레이블 꼭지점들에 대해서만 I의 값이 0보다 크지만, 감성이 확산됨에 따라 감성 레이블과 직접적(같은 트윗에 들어있는 자질들), 간접적으로 연결된 자질들도 0보다 큰 I를 갖게 된다. 이것이 준지도학습의 핵심이다. 한편 레이블 꼭지점이 두 개 이상인 감성의 경우, 감성강도의 정의에 따라 그래프의 각 꼭지점은 감성강도가 가장 큰 경로만을 취한다. 따라서 임의의 꼭지점의 I는 증가하는 방향으로만 업데이트 되므로 모든 감성 레이블 꼭지점들에 대하여 순차적으로 업데이트가 완료되면 I는 실제 감성강도 값으로 수렴한다. 또한 그래프에 싸이클이 존재하는 경우에도, w_{ij}은 [0,1]의 값을 가지므로 문제없이 감성강도를 계산할 수 있다.

한편, 모든 감성에 대해 높은 감성강도를 갖는 자질을 처리하기 위해서, 감성강도의 평균값을 빼서 조정하도록 한다. 즉, 조정된 감성강도 I_{ik}'는 다음과 같이 계산된다.

$$I_{ik}' = I_{ik} - \frac{1}{M-1} \sum_{\substack{p \neq k \\ 1 \leq p \leq M}} I_{ip}$$

이렇게 계산된 감성강도에 대해서, 각 감성별로 적절한 역치(threshold)를 선정하여 감성어휘에 포함시킨다.

3.2.2 트윗의 RT 처리

하나의 트윗에 리트윗한 내용이 포함되어 있을 수 있다. 예를 들어, "아후~!! 속 터져~!! RT @asdfjkl: 그리고 어제부터 쇼고객센터 로그인이 안되는데ㅠㅠ"라는 트윗은 아이디가 asdfjkl인 사용자가 쓴 트윗에 자신의 생각을 덧붙여 작성한 것이고, 이 경우 asdfjkl의 트윗이 리트윗되었다고 한다. 이런 경우에 트윗 하나의 범위를 어디까지 할 것인지 결정해야 한다. 인기가 많아서 리트윗이 많이 되는 트윗은 데이터에 자주 중복되어 나타나게 되고 이렇게 빈도가 비정상적으로 높아진 트윗은 자질 간의 상호정보량을 계산하는 데에 혼란을 줄 수 있다. 따라서 본 논문에서는 트윗의 범위를 다음의 세 가지 경우로 구분하여 실험하였다.

- RTLevel: 어떤 트윗이 리트윗을 포함할 경우, 각 리트윗을 별개의 트윗으로 간주한다. 가령, 어떤 트윗에 리트윗이 두 개 포함되어 있을 경우, 세 개의 트윗(직접 작성한 내용 + 두 개의 리트윗)으로 간주한다.
- TweetLevel: 어떤 트윗이 리트윗을 포함하고 있더라도 모두 합쳐서 하나의 트윗으로 간주한다.
- TweetLevel+: TweetLevel와 비슷한 방식이나, 리트윗 내용은 자질 간 상호정보량을 계산하지 않는다. 이는 TweetLevel에서 빈도가 높은 리트윗은 자질 간 상호정보량이 중복 계산되어 결과가 왜곡되는 것을 막기 위함이다.

4. 평가 준비

본 논문에서 제안하는 알고리즘을 통해 구축된 감성어휘를 평가하기 위해서, 구축된 감성어휘를 이용해 트윗의 감성을 분석하는 성능을 평가하기로 한다. 이 장에서는 구체적인 평가 방법, 기본감성어휘를 구축하는 방법, 평가데이터를 구축하는 방법, 선택한 자질들에 대해서 설명하도록 한다.

4.1 평가 방법 및 척도

본 논문에서 제시하는 알고리즘의 목적은 트위터 데이터 같은 빅데이터의 감성을 분석하여 여론을 파악하는 것이다. 이를 위해, 알고리즘을 통해 구축된 감성어휘를 사용하여 각 트윗별로 20종류의 감성을 분석하는 성능을 측정하도록 한다. 하나의 트윗이 다양한 감성을 표현할 수 있기 때문에, 이 실험에서는 하나의 트윗을 하나의 세부감성으로 분류하지 않고 대신 트윗이 가지고 있는 모든 감성을 찾도록 한다. 이를 위해, 구축된 감성어휘에 포함된 자질이 들어있는 트윗은 해당 감성을 지니고 있다고 판단하였다. 한편, 각 감성에 대해서 감성강도가 얼마인 자질들까지 감성어휘에 포함시킬지 그 역치를 정해야 한다. 본 실험에서는 0부터 1 사이를 0.02 간격으로 역치를 선정하여 총 50가지 경우에 대해 모두 평가해보고 성능이 가장 좋은 역치를 선택하였다.

일반적으로 이러한 정보추출의 성능을 평가하기 위한 척도로서 정확도와 재현율, 그리고 둘의 조화평균인 F1 score가 많이 사용된다. 각 감성별로 감성어휘의 역치에 따른 F1 score를 계산한 후 바로 성능 척도로 사용하는 것이 직관적인 방법이기는 하나, 본 논문에서 성능 척도로 사용하기에는 곤란한 점이 있다. 지도학습이 아닌 준지도학습을 사용한다는 점, 어휘 기반으로 감성을 분석한다는 점, 그리고 트윗의 절반 이상은 감성이 없다는 점 때문에, 재현율이 정확도에 비해 매우 낮다. 다시 말해, F1 score가 재현율에 지배적이게 되며, 재현율을 높이기 위해서는 정확도가 크게 희생된다. 이는 본 논문에서 목적을 두고 있는 브랜드 모니터링 서비스에서 정확도를 우선시 하는 것과 상충한다.

따라서 본 실험에서는 기본적으로 사용자들이 납득할 만한 수준의 정확도를 실험적으로 70%로 잡고, 정확도가 70% 이상인 감성어휘는 70% 미만인 감성어휘보다 무조건 우위에 있도록 한다. 정확도가 70% 이상인 감성어휘들 간에는 F1 score를 이용하여 성능을 비교하고, 정확도가 70% 미만인 감성어휘들 간에는 정확도를 기준으로 성능을 비교한다. 즉, 감성어휘 L의 성능 $PERF(L)$은 다음과 같이 정의된다.

$$PERF(L) = \begin{cases} Fscore(L) + 1 & \text{if } Prec(L) \geq 0.70 \\ Prec(L) & \text{otherwise} \end{cases}$$

4.2 기본감성어휘 구축

영어로 된 감성어휘 중에서 널리 사용되는 어휘로는 WordNet-Affect[8], LIWC[9], SentiWordNet[10] 등이 있으나, 한국어를 위해 사용할 만한 감성어휘는 구하기 어려운 실정이다. 세부감성을 이용해 노래 가사의 감성을 분석한 비교적 최근 논문[11]에서는 감성어휘를 수작업으로 구축하여 사용하였다. 따라서 본 논문에서는 간단한 방식을 사용하여 다음 항목들로 구성된 기본감성어휘를 구축하였다.

(ㄱ) 세부감성의 이름 (예: "감동", "슬픔")
(ㄴ) 세부감성에 포함되는 감성의 이름(표 1) (예: "감탄", "존경", "연민", "애처로움")
(ㄷ) 세부감성의 동의어 (예: "감동"의 동의어 "감명")

도메인	키워드	트윗 개수
인물	김태촌, 곽노현, 박근혜, 나경원, 박원순, 이명박, 한명숙, 문재인, 안철수, 나꼼수, 정봉주	1700
스마트 기기	아이폰, 스마트폰, 배터리, 킨들, 블랙베리, 베가, 갤럭시탭, 애플TV, LTE, 모바일OS, 옵티머스, 넥서스폰, 아이팟, 갤럭시폰, 아이패드, 구글폰	2000
정책	민주당, 전자팔찌, 종합편성채널, 셧다운제, 천안함, 반값등록금, 학생인권조례, 새누리당, 실명제, 무상급식, 요일제, 한미FTA	2000

표 2 평가데이터 키워드

(ㄷ)에서 너무 일반적인 동의어는 제외하였다. 예를 들어, "감동"의 동의어 중에 "느낌"이 있는데, "느낌"은 일반적으로 "감동"의 의미보다 더 포괄적으로 사용되므로 제외하였다. 또한 감성 이름 중에 명사형으로밖에 사용될 수 없는 경우는 동사로 사용될 수 있도록 형태를 확장하였다. 예를 들어, "두려움"과 같은 감성은 "두렵다", "두려워" 등도 포함할 수 있도록 형태를 확장하였다. 이렇게 하여, (ㄱ)과 (ㄴ)을 통해 구축된 용어는 140개, (ㄷ)을 통해 형태 확장된 용어는 60개가 되어 총 200개의 용어를 가진 기본감성어휘를 구축하였다.

4.3 학습데이터

학습데이터는 다음과 같은 방식으로 트윗을 샘플링하여 구축하였다. 본 논문에서 전제로 하는 서비스의 특성상 특정 도메인에 특화되지 않은 범용화 된 어휘사전을 구축하는 것이 목표이므로, 트위터 상의 모든 트윗을 대상으로 샘플링하였다. 그러나 트위터 상의 트윗 중 85% 이상은 감성이 없기 때문에[6], 완전히 임의로 샘플링을 할 경우 감성과 잠재적으로 관련 있는 자질을 충분히 확보하기 위해서 필요한 샘플의 크기가 매우 커지게 되고 자연히 학습에 필요한 메모리와 시간도 크게 증가한다. 따라서 감성과 관련 있는 자질을 확보하면서 샘플의 크기를 줄이기 위해서 "기분", "가슴이", "느낌"이 들어간 트윗들을 샘플링하였다. 이는 한 연구[12]에서 "We feel"이라는 텍스트가 들어있는 문장을 분석하여 블로그 상의 감성을 분석한 데에서 아이디어를 얻은 것이다. 이런 방식으로, 2010년 11월부터 2011년 3월 사이에 작성된 트윗을 대상으로 2백만 개의 트윗을 샘플링하여 학습데이터로 사용하였다. 그중 기본감성어휘에 포함된 용어가 들어있는 트윗은 태깅 데이터로, 나머지 트윗들은 태깅되지 않은 데이터로 사용되어 준지도학습을 하게 된다. 2백만 개의 트윗 중에서 감성이 태깅된 데이터는 419,456개이고, 태깅이 되지 않은 데이터는 1,580,544개이다. 학습데이터에서 기본감성어휘를 통해 태깅된 세부감성 분포가 그림 1에 나와 있다. '기쁨'과 '좋아함' 감성이 많이 태깅되었는데, 이는 두 감성의 기본감성어휘 중 "기쁘다", "즐겁다", "사랑", "좋아하다" 등의 표현이 텍스트 상에 많이 나타났기 때문이다.

4.4 평가데이터

평가데이터를 구축하기 위해서, 2011년에 작성된 트윗 중에서 다양한 이슈와 관련된 트윗들을 샘플링하고 여기에 20종류의 세부감성을 태깅하였다. 고려한 이슈는 인물, 스마트기기, 정책 중 하나에 속하며, 그 리스트가 표 2에 나열되어 있다.

태깅은 각 트윗에 대하여 트윗에 들어있는 모든 세부감성을 태깅하였다. 하나의 트윗에 대해 세 명이 상의하여 둘 이상의 동의를 얻은 감성만 태깅하였고, 그런 감성이 없는 트윗은 '중립'으로 태깅하였다. 총 5700개의 트윗 중 63%의 트윗이 '중립'으로 판단되었다. 감성이 존재하는 트윗에 대해서 세부감성의 비율이 그림 2에 있다. '실망', '싫어함', '화남', '슬픔'과 같은 부정적인 감성이 많은 비율을 차지하고 있고, '자신감', '미안함', '곤란' 등의 감성은 그 비율이 매우 낮은 것을 알 수 있다.

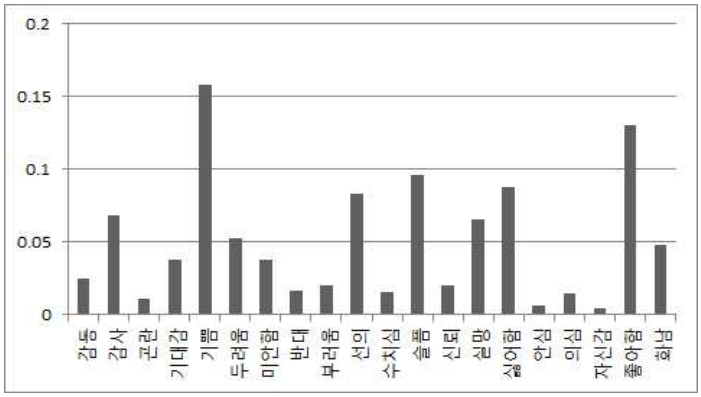


그림 1 학습데이터 내 태깅된 세부감성 분포

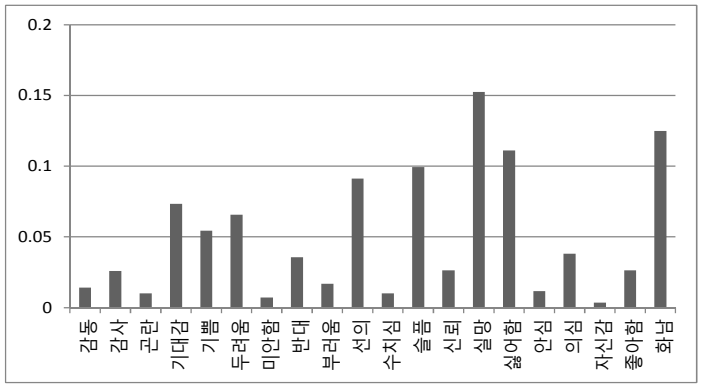


그림 2 평가데이터 세부감성 분포

평가데이터에 포함된 트윗의 예는 다음과 같다.

- 실망: #MLB_ 9호선은 DMB가 안 되는군요... 급행 빼고는 좋은 게 없는 듯... 4칸 짜리 객차로 어쩌려고...
- 기쁨: 수요일은 차량요일제로 전철 이용하는 날이 많습니다. 전철에서 간만에 Podcast 골라 듣고 회사 입구에서 커피 한잔 사들고 오는 맛도 괜찮군요!

4.5 자질

감성어휘에 들어갈 자질로 다음 여섯 개를 사용했다.

- MRP: 명사, 동사, 형용사 형태소 (예: 따분하/pa)
- MRPN: 명사, 동사, 형용사, 부사, 의존명사 형태소의 바이그램 (예: 우울하/pa 기분/pv)
- WD: 단어 (예: 감사합니다!, 디자인이)
- WDN: 단어 바이그램 (예: 뭔가 어색한, 본의 아니게)
- PRD: 명사 형태소 + 용언(동사, 형용사, 명사) 형태소 (예: 걱정/nc 많/pa, 희망/nc 없/pa)
- RM: 문장의 마지막 용언 형태소 (예: 방심하/pv)

형태소와 관련된 자질에 관해서는, 자질을 추출하기 전에 몇 개의 정규표현을 사용해 부정어 처리를 하였다. 예를 들어, "기분/nc 이/jc 좋/pa 지/ec 않/px 다/ef"의 경우 "기분/nc 이/jc 안_좋/pa 다/ef"와 같이 변환하였다. 학습데이터에 대해, 빈도가 낮은 자질은

MRP	MRPN	WD	WDN	PRD	RM
13700	12536	18553	14892	10807	3975

표 3 자질별 빈도

필터링하였고, 최종적으로 남은 자질의 개수가 표 3에 나와 있다.

5. 평가 결과

트윗을 구분하는 세 가지 방법(RTLevel, TweetLevel, TweetLevel+)과 여섯 종류의 자질(MRP, MRPN, WD, WDN, PRD, RM)을 조합하여 세부감성 분석 성능을 측정하였다. 각각의 트윗 구분 방법에 대해, 최대 두 개의 자질을 조합하였고, 감성별로 가장 높은 성능을 보이는 감성어휘를 선택하였다.

5.1 트윗 구분 방법별 성능

먼저 기본감성어휘만을 이용한 방법과 트윗을 구분하는 세 가지 방법 RTLevel, TweetLevel, TweetLevel+에 대해 성능을 측정하였고, 그 결과가 표 4에 나와 있다. 세 방법 중에서 가장 높은 성능을 보인 결과를 볼드체 및 음영으로 표시하였다.

전체적으로 몇 개의 감성을 제외하고는 재현율이 매우 낮은 것을 알 수 있다. 이는 도메인에 특화된 처리를 전혀 하지 않은 상태에서 중립 트윗이 절반 이상을 차지하는 평가데이터에 대해 높은 정확도를 얻으려고 하다 보니 맞닥뜨리게 되는 한계로 보인다.

기본감성어휘가 들어있는지 여부로 감성을 판단한 결과는, 구축된 감성어휘를 사용한 결과에 비해 전반적으로 정확도가 떨어졌다. 주된 이유는 기본감성어휘에 있는 "만족", "사랑" 등의 짧은 표현들이 감성을 부정확하게 판단했기 때문이다. 다만 '부러움'과 '슬픔' 감성에 대해서는 구축된 감성어휘에 비해서 높은 성능을 나타내었다. '부러움'의 경우 "부럽다", "질투", "탐나다"의 표현만으로 높은 정확도와 재현율을 내었고, 이를 통해 트위터 상에서 '부러움' 감성이 이 표현들을 통해 주로 나타남을 알 수 있다. 반면 '슬픔'에 해당하는 기본감성어휘에는 "슬프다", "연민", "애처롭다" 등 약 15 종류의 표현이 들어있다. 이 표현들의 존재를 통해 감성을 정확하게 판단할 수 있었으며, 또한 표현의 다양성으로 인해 재현율도 높았다.

RTLevel, TweetLevel, TweetLevel+의 성능을 비교해보면 눈에 띄는 차이는 보이지 않는다. 다만 '감동'의 경우에는 TweetLevel와 TweetLevel+에서 정확도 50% 이상인 감성어휘를 얻을 수 없었던 반면에 RTLevel에서는 얻을 수 있었다. 결과 어휘를 자세히 분석한 결과, RTLevel에서는 '감동'을 직접적으로 표현하는 RM 자질(예: 감탄스럽/pa, 감동적/nc)이 상위를 차지하고 있었던 반면, RTLevel+는 주로 명사(예: 감동/nc)가 상위를 차지하고 있었다. 이는 트윗에서 직접 작성한 내용과 리트윗을 함께 고려할 때 서로 영향을 주어 좀 더 직접적인 감성표현을 얻지 못하는 것으로 보인다. '미안함' 감정도 비슷하게 RTLevel에서 더 높은 정확도와 재현율을 얻을 수 있었다. 결과 어휘를 분석한 결과, 이번에는 '감동'과 반대로 RTLevel에서는 '미안함'의 이유가 되는 MRPN 자질들(예: "기분/nc 상하/pa", "기분/nc 나쁘/pa")이 상위를 차지한 반면, TweetLevel에서는 '미안함'을 직접적으로 표현하는 RM 자질들(예: 죄송하/pa, 미안하/pa)이 상위를 차지했음을 알 수 있었다. 이는 트윗에서 '미안함'을 표현할 경우, 주로 리트윗과 상관없이 미안함을 표현하기 때문이다.

한편, TweetLevel와 TweetLevel+를 비교하면, TweetLevel이 TweetLevel+에 비해 다소 높은 성능을 내는 것 같으나 차이가 그리 크지는 않았다.

	기본감성어휘			RTLevel			TweetLevel			TweetLevel+		
	Prec	Recl	PERF	Prec	Recl	PERF	Prec	Recl	PERF	Prec	Recl	PERF
감동	0.31	0.14	0.31	**1.00**	**0.03**	**1.07**	0.50	0.03	0.50	0.50	0.07	0.50
감사	0.65	0.53	0.65	**0.74**	**0.53**	**1.62**	**0.74**	**0.53**	**1.62**	**0.76**	**0.53**	**1.62**
곤란	0.50	0.21	0.50	**1.00**	**0.05**	**1.10**	**1.00**	**0.05**	**1.10**	**1.00**	**0.05**	**1.10**
기대감	0.38	0.06	0.38	**0.71**	**0.03**	**1.06**	**0.71**	**0.03**	**1.06**	**0.71**	**0.03**	**1.06**
기쁨	0.22	0.10	0.22	0.22	0.04	0.22	**0.24**	**0.04**	**0.24**	**0.24**	**0.04**	**0.24**
두려움	0.43	0.16	0.43	**0.88**	**0.06**	**1.10**	0.86	0.05	1.09	0.86	0.05	1.09
미안함	0.44	0.57	0.44	**0.86**	**0.43**	**1.57**	0.80	0.29	1.42	0.80	0.29	1.42
반대	0.18	0.16	0.18	**1.00**	**0.01**	**1.03**	**1.00**	**0.01**	**1.03**	**1.00**	**0.01**	**1.03**
부러움	**0.79**	**0.40**	**1.53**	0.78	0.37	1.50	0.78	0.37	1.50	0.78	0.37	1.50
선의	0.52	0.23	0.52	**0.75**	**0.03**	**1.07**	0.83	0.03	1.05	0.83	0.03	1.05
수치심	0.29	0.20	0.29	**1.00**	**0.05**	**1.10**	**1.00**	**0.05**	**1.10**	**1.00**	**0.05**	**1.10**
슬픔	**0.87**	**0.14**	**1.24**	0.88	0.04	1.07	0.88	0.04	1.07	0.88	0.04	1.07
신뢰	0.03	0.02	0.03	**0.33**	**0.02**	**0.33**	**0.33**	**0.02**	**0.33**	**0.33**	**0.02**	**0.33**
실망	0.60	0.07	0.60	0.70	0.05	1.08	**0.71**	**0.06**	**1.10**	**0.71**	**0.06**	**1.10**
싫어함	0.55	0.14	0.55	0.78	0.03	1.06	**0.80**	**0.05**	**1.10**	0.77	0.05	1.09
안심	0.01	0.04	0.01	**1.00**	**0.04**	**1.08**	**1.00**	**0.04**	**1.08**	**1.00**	**0.04**	**1.08**
의심	-	0.00	-	0.11	0.01	0.11	**0.13**	**0.01**	**0.13**	0.07	0.01	0.07
자신감	0.00	0.00	0.00	0.01	0.29	0.01	0.01	0.14	0.01	**0.02**	**0.57**	**0.02**
좋아함	0.22	0.14	0.22	**1.00**	**0.02**	**1.04**	**1.00**	**0.02**	**1.04**	**1.00**	**0.02**	**1.04**
화남	0.54	0.05	0.54	**1.00**	**0.01**	**1.02**	**1.00**	**0.01**	**1.02**	**1.00**	**0.01**	**1.02**

표 4 RTLevel, TweetLevel, TweetLevel+ 성능

	PERF	자질들	예
감동	1.07	PRD+RM	RM:감탄스럽/pa RM:감격하/pa
감사	1.62	MRP+WD	WD:감사감사 MRP:감사합니당/nc
곤란	1.10	PRD+RM	RM:안_어색하/pa PRD0:느낌/nc 어색하/pa
기대감	1.06	MRPN+WD	MRP2:가슴/nc 설레/pv WD:희망찬
두려움	1.10	MRPN+RM	MRP2:불안하/pv 기분/nc RM:섬뜩하/pa
미안함	1.57	MRP+MRPN	MRP2:ㅠㅠ기분/nc 나쁘/pa MRP2:혹시/mag 기분/nc
반대	1.03	MRPN+RM	RM:반대하/pv RM:거랑같/nc
부러움	1.50	MRP+MRPN	MRP:탐나/pv MRP:질투/nc
선의	1.07	RM+WDN	RM:ㅎㅎ화이팅/nc WD2:힘내!!!
수치심	1.10	MRPN+RM	RM:치욕스럽/pa MRP2:부끄러움/nc 느끼/pv
슬픔	1.07	MRPN+RM	RM:우울증/nc MRP2:우울하/pv 날/nc
실망	1.10	MRP+MRPN	MRP:아쉽/pa MRP2:아쉬움/nc 남/pv
싫어함	1.10	MRP+WD	WD:싫어 MRP:지루하/pa WD:불쾌한
안심	1.08	MRPN+PRD	MRP2:한숨/nc 쉬/pv PRD0:스트레스/nc 날리/pv
좋아함	1.04	RM	RM:호감/nc RM:사랑/nc
화남	1.02	RM+WDN	RM:일투성이/nc RM:삭감하/nc

표 5 감성별 어휘 자질

5.2 감성별 어휘 분석

다음으로, 트윗 구분 방법에 관계없이 감성별로 가장 좋은 성능을 얻은 감성어휘들을 분석하였다. $PERF$ 값이 1 이상인 감성들에 대해 상위를 차지하는 자질들의 예가 표 5에 나와 있다.

대부분의 감성에서 감성을 직접적으로 표현하는 자질들이 상위를 차지하고 있음을 알 수 있다. 하지만 '미안함'의 경우 "기분이 나빴다면"이나 "혹시 기분이 상하셨다면"과 같은 표현을 학습했음을 알

	통합 전			통합 후			향상 성능		
	Prec	Recl	PERF	Prec	Recl	PERF	Prec	Recl	PERF
감동	1.00	0.03	1.07	1.00	0.07	1.13	-	+0.04	+0.06
미안함	0.86	0.43	1.57	0.70	0.50	1.58	-0.16	+0.07	+0.01
선의	0.75	0.03	1.07	0.83	0.06	1.11	+0.08	+0.02	+0.04
슬픔	0.88	0.04	1.07	0.90	0.05	1.09	+0.03	+0.01	+0.02
싫어함	0.80	0.05	1.10	0.79	0.07	1.12	-0.01	+0.01	+0.02
좋아함	1.00	0.02	1.04	1.00	0.04	1.07	-	+0.02	+0.03
화남	1.00	0.01	1.02	0.80	0.02	1.03	-0.20	+0.00	+0.01

표 6 어휘 통합 전후 성능 비교

수 있고, '안심'의 경우 "한숨을 쉬다" 혹은 "스트레스를 날리다"와 같은 표현을 학습했음을 알 수 있다. 또한 '화남'의 경우에는 "일투성이"나 "임금을 삭감"한다는 표현을 학습하였다. 특정 상품이나 정책에 관한 트윗들을 대상으로 학습을 하였다면 그 도메인에 특화된 간접적인 감성 표현들(예를 들어, "너무 비싸다")이 많이 추출될 수 있겠지만, 본 실험에서는 일반적인 트윗을 사용해 학습하였기 때문에 감성을 직접적으로 나타내는 표현들의 비율이 상대적으로 높다. 그럼에도 불구하고 감성을 직접적으로 표현하는 기본감성어휘만 가지고 위와 같은 간접적인 감성 표현들을 학습해 낸 것을 통하여, 본 알고리즘에서 사용한 준지도학습의 유용성을 알 수 있다.

한편, '곤란'에서는 "안_어색하/pa"와 같은 자질이 상위로 잘못 추출되었다. 이는 학습데이터에 기본감성어휘를 적용할 때, 부정어 처리를 하지 않았기 때문에 발생한 문제이다. 기본감성어휘와 대량의 데이터에 대해서 정교한 부정어 처리를 할 경우에 성능이 향상될 것으로 예상된다.

정확도가 높은 감성어휘들을 통합하여 사용할 경우, 정확도를 크게 희생하지 않으면서 낮은 재현율을 보충할 수 있을 것이라 예상할 수 있다. 따라서 트윗 구분 방법과 자질 조합에 관계없이 감성별로 정확도가 70% 이상인 감성어휘들을 통합하여 성능을 측정하였고, 성능 향상을 보인 감성들이 표 6에 나와 있다. '감동', '미안함', '선의', '슬픔', '싫어함', '좋아함', '화남'에 대해 PERF가 향상되었다. 자질들을 통합함으로써 정확도가 떨어지는 경우가 많으나, 재현율이 올라가면서 전체적인 성능은 향상되었다. 이는 서비스 사용자들이 납득할 만한 정확도 범위 내에서 재현율이 상승되었음을 의미한다. '감동'과 '미안함'은 재현율이 큰 폭으로 상승하였고, '선의', '슬픔', '좋아함'은 정확도의 손실 없이 재현율이 향상되었다.

6. 결론

본 논문에서는 한국어로 된 트윗으로부터 20종류의 세부감성을 분석하기 위한 감성어휘를 구축하는 알고리즘을 제시하고 평가 결과를 제시하였다. 그래프를 기반으로 한 준지도학습을 이용하여, 적은 양의 기본감성어휘로부터 감성을 확장하는 알고리즘을 제시하였다.

도메인에 특화되지 않은 일반적인 트윗과, 직접적인 감성 표현들로 이루어진 소수의 기본감성어휘만 가지고 준지도학습을 통하여 감성어휘를 구축할 수 있었다. 트윗을 구분하는 방법에 따라 성능을 측정한 결과, 하나의 트윗 내에서 직접 작성한 부분과 리트윗 부분을 모두 하나의 트윗으로 간주한 경우와 모든 리트윗을 별개의 트윗으로 간주한 경우 성능 차이가 크지는 않았으나, 몇 감성에 대해서 후자의 경우 성능이 크게 증가하는 것을 확인하였다. 또한 학습된 감성어휘를 분석한 결과, 대부분 직접적인 감성표현이 상위를 차지하였으나 일부 감성들에서는 간접적인 감성표현들이 상위를 차지하는 것을 확인하였다.

본 논문은 세부감성 분석에 대한 기초 연구로서 발전 가능성이 크다. 먼저 본 논문에서 사용한 기본감성어휘는 주로 명사형이고 부정어 처리를 하지 않았기 때문에 감성을 태깅하는 데에 한계가 있다. 기본감성어휘를 좀 더 정교하게 처리함으로써 성능을 향상 시킬 수 있을 것이다. 또한 본 논문에서는 구축된 감성어휘의 일반성을 위해 도메인에 특화된 어떠한 정보나 데이터도 사용하지 않았다. 특정 도메인에 해당되는 학습데이터와 메타정보들을 활용한다면 더 높은 성능을 얻을 수 있을 것이다.

참고문헌

[1] Cortes, C. & Vapnik, V., Support-vector networks. *Machine learning*, 20(3), 273-297, 1995.

[1] Cortes, C. and Vapnik, V. N., Support-Vector Networks, *Machine Learning*, Springer, 1995.

[2] Maas, A. L., Ng, A. Y., and Potts, C., *M*ulti-Dimensional Sentiment Analysis with Learned Representations. 2011.

[3] Kim, S. M., Valitutti, A., and Calvo, R. A., Evaluation of unsupervised emotion models to textual affect recognition. In *Proceedings of the NAACL HLT 2010 Workshop on Computational Approaches to Analysis and Generation of Emotion in Text*, pp. 62-70., 2010.

[4] Velikovich, L., Blair-Goldensohn, S., Hannan, K. and McDonald, R., The viability of web-derived polarity lexicons. In *the 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pp. 777-785., 2010.

[5] Shawe-Taylor, C. A., and Schölkopf, S., The Support Vector Machine, 2000.

[6] 장문수, 심리학적 감정과 소셜 웹 자료를 이용한 감성의 실증적 분류. *한국지능시스템학회 논문지* 제22권 제5호, 2012.10, 563-569, 2012.

[7] Plutchik, R. and Kellerman, H., Emotion: Theory, Research, and Experience. Vol.5, Academic Press, 1990.

[8] Strapparava, C. and Valitutti, A., WordNet-Affect: an affective extension of WordNet. In *Proc. of 4th International Conference on Language Resources and Evaluation*, 2004.

[9] Pennebaker, J. W., Francis, M. E., and Booth, R. J., Linguistic inquiry and word count: LIWC 2001. *Mahway: Lawrence Erlbaum Associates* (2001): 71, 2001.

[10] Esuli, A., and Sebastiani, F., Sentiwordnet: A publicly available lexical resource for opinion mining. In *Proceedings of LREC*, vol. 6, pp. 417-422. 2006.

[11] 윤애선, 임경업, 윤애선, 권혁철, 감정 온톨로지를 활용한 가사 기반의 음악 감정 추출, *한국지능정보시스템학회* 2010년 추계학술대회, pp.333-337, 2010.

[12] Kamvar, S. D. and Harris, J., We feel fine and searching the emotional web. *In Proceedings of the fourth ACM international conference on Web search and data mining*, pp. 117-126. ACM, 2011.

바이오 이벤트 추출을 위한 피쳐 개발

이석준○, 김영태, 황민국, 임수종†, 나동열
연세대학교, 컴퓨터정보통신공학부
†한국전자통신연구원
seokjun88@naver.com, wolfnamu@gmail.com, peacsid@nate.com, isj@etri.re.kr, dyra@yonsei.ac.kr

Developing Features for Bio Event Extraction

Seok-Jun Lee○, Young-Tae Kim, Min-Kook Hwang, Soo-Jong Lim†, Dong-yul Ra
Yonsei University, Computer & Telecommunications Engineering Division
†Electronics & Telecommunications Research Institute

요 약

본 논문은 바이오 문서에서의 정보추출 시스템 개발에 대한 것이다. 이 시스템의 목표는 바이오 관련 문서에서 바이오 이벤트의 발생을 탐지하고 이벤트의 타입 및 이벤트에 관여된 필수 논항을 채우는 구문요소를 인식하는 것이다. 우리는 두 개의 별도의 단계를 이용하는 시스템 구성을 사용한다. 첫 단계에서는 SVM을 사용하여 이벤트의 발생 및 이벤트의 타입을 결정한다. 두 번째 단계에서는 이벤트의 논항을 채우는 참여자를 인식하는 작업을 한다. 본 논문은 단계 1에서 사용되는 SVM의 피쳐 리스트의 개발에 대한 문제를 다룬다. 본 논문에서 제안하는 피쳐 리스트를 사용하여 좋은 성능을 가지는 첫 단계에 대한 모듈을 얻을 수 있음을 관찰하였다.

주제어: 정보추출, 이벤트추출, 바이오 이벤트, 바이오 문서, 이벤트 탐지.

1. 서론

구글이나 네이버 등에서 볼 수 있듯이 정보검색 (information retrieval; IR) 기술의 사회에 대한 기여는 막대하다. 이와 마찬가지로 정보추출 (information extraction; IE) 기술의 발전이 사회에 제공할 기여 또한 매우 클 것으로 관측된다. 그러나 IE 의 가장 큰 문제인 텍스트 마이닝 (text mining) 기술의 경우 상용화에 이르기 위해서는 아직 많은 발전을 필요로 하는 실정이다. 이러한 기술의 발전을 위해 자연어처리 연구분야에서는 여러 텍스트 마이닝 관련 학술 대회가 개최되었다.

이들 중에서도 대표적인 것으로 BioNLP 학술대회를 들 수 있다. 이미 여러 해 째 진행되고 있는 이 학술대회는 생명공학과 관련된 문서를 대상으로 하는 정보추출 시스템 개발 기술의 발전을 목표로 한다. 시스템 개발 참가팀들은 미리 공통으로 주어진 작업(shared task)을 수행하는 시스템의 개발을 위해서 미리 제공되는 학습데이타(training data)를 이용할 수 있다.

본 논문에서는 BioNLP-2009 의 ST(shared task) 1에 대한 시스템 개발을 목표로 한다. BioNLP-2009 ST-1은 바이오 문서에서 단백질과 관련된 이벤트의 추출을 목표로 한다[1]. 즉 문서에서 미리 정해진 종류의 이벤트 및 이벤트에 관여된 참여자(participant) 즉 논항(argument)을 인식하는 것을 목표로 한다. 이 작업은 바이오 정보추출의 가장 기본적인 형태로서 정보추출과 관련된 가장 핵심(core)이 되는 기술이라고 생각된다.

이 작업을 위한 시스템의 개발은 여러 가지 접근 방법이 가능하다. 본 논문에서는 다음과 같이 구분된 2 단계로 구성된 시스템을 사용하는 것을 가정한다. 즉 단계 1에서 문서에 존재하는 이벤트 발생 (trigger) 탐지 및 그것의 타입(type)을 결정한다. 그 다음 별도의 다음 단계 2에서 앞에서 탐지된 각 이벤트에 대하여 그것의 논항을 채우는 참여자를 찾아서 이벤트를 완성하도록 한다.

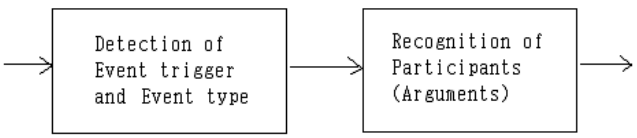


그림 1: 바이오 이벤트 추출 시스템 구성.

그러나 BioNLP-2009 ST-1에 참여한 팀들은 여러 형태의 다른 방식의 접근을 선택한 경우도 많다. 예를 들면 위의 두 단계를 별도로 구분하지 않고 하나의 단계로 하여 이벤트의 존재 및 종류의 결정을 논항의 탐지와 병행하는 기법을 사용하는 것이다.

우리는 단계 1을 위한 모듈의 개발을 위하여 다중-클래스(multiple class) SVM 을 사용하기로 하였다. 본 논문에서는 이를 위한 효과적인 피쳐(feature) 집합의 개발을 목표로 한다. 실험 결과 본 연구에서 제안하는 피쳐 집합을 사용하는 경우 높은 성능의 단계 1 모듈의 개발이 가능한 것으로 생각된다.

단계 2에 대한 모듈의 개발을 위해서 먼저 우리는 기계학습 방법을 사용하는 대신 의존트리 탐색을 통하여 이벤트의 논항을 찾는 휴리스틱 기반 탐색 기법을 시도해 보았으나 이 기법으로는 좋은 성능을 얻지 못하였다. 이 모듈에 대해서는 결국 기계학습을 포함한 다른 기법을 탐구할 필요가 있다.

2. 관련 연구

자연어처리 기술의 발전을 위하여 공유 작업(shared task) 문제를 설정하고 이에 대한 시스템의 개발 후 그 경험을 교류하는 형태의 학술대회가 많이 이용되었다. 특히 텍스트 마이닝의 경우 MUC, TREC, ACE 등은 관련 기술의 발전에 많은 기여를 한 것으로 생각된다. 바이오-텍스트 마이닝의 경우도 예외는 아니어서 TREC Genomics track, JNLPBA, LLL, BioCreative 등과 같은 학술대회가 이에 속한다. 이 중에서도 가장 대표적인 것으로는 BioNLP 를 들 수 있다[1]. 이 학술대회는 2006년 이래로 매년 열리고 있는데 바이오 정보추출 기술의 발전에 매우 중요한 역할을 한 것으로 평가된다. 이러한 학술대회에서는 shared task 에 대한 시스템의 개발에 도움이 되는 자원(resource)을 제공한다. 이의 대표적인 것은 학습에 이용될 수 있는 학습데이타이다. 대부분의 개별 연구자들은 비용과 시간상의 문제로 인해 독자적으로 이러한 학습 데이타를 구축하여 사용할 형편이 되지 못한다. 따라서 시스템 개발에 필수적인 학습데이타의 제공은 많은 연구집단으로 하여금 연구 및 개발을 가능케 하므로 기술의 발전에 매우 중요한 역할을 한다.

3. 문제정의 및 데이타

바이오 정보 추출에 대한 기본적인 기술의 개발을 목적으로 우리는 BioNLP-2009 의 문제 및 데이타를 이용한다. 이렇게 문제를 선정한 이유는 여기에서 주어진 문제를 연구함으로서 가장 기초적이고 핵심적인 이벤트 추출 기술에 대한 연구가 가능한 것으로 판단하였기 때문이다.

3.1. 문제정의

BioNLP-2009 ST(shared task) 1의 문제는 테스트 데이타로 제공된 바이오 문서 안에서 나타난 모든 이벤트에 대하여 다음과 같은 정보를 추출하여야 한다:

- 이벤트의 발생을 알리는 트리거 (스트링) 탐지 (ETD),
- 이벤트의 타입 분류(ETC),
- 이벤트에 관계된 필수 논항 참여자 인식(EPR).

3.2. 데이타

시스템의 개발 및 실험을 위해 다음과 같은 데이타가 제공된다. 이와 같은 데이타는 바이오 이벤트 태깅 데이타 자원인 GENIA에서 추출한 데이타를 가공한 것이다[2].

- 훈련 데이타(training data): 시스템의 학습에 이용되는 데이타.
- 개발 데이타(development data): 시스템의 튜닝(tuning)에 이용되는 데이타.
- 테스트 데이타(test data): 시스템의 성능 측정에 사용되는 데이타.

위의 각 데이타는 GENIA 바이오 문서 집단으로 구성되며 문장에 대하여 다음과 같은 정보가 태깅(tagging) 되어 있다:

- 단백질: 단백질을 나타내는 스트링.
- 이벤트 트리거: 이벤트 발생을 알리는 스트링.
- 이벤트: 고유 번호가 부여된 이벤트.
- 이벤트 논항: 각 이벤트에 참여하는 논항(들).

훈련 데이타와 개발 데이타는 위의 4 가지 정보가 모두 태깅되어 있다. 테스트 데이타의 경우에는 단백질 정보만이 태깅되어 있다.

바이오 이벤트의 종류는 다음 표 1과 같이 9가지이다. 이 표에는 각 타입의 이벤트가 가질 수 있는 주요논항(primary arguments)이 표시되어 있다.

표 1. 추출 대상 이벤트의 종류.

타입(type)	주요논항(primary arguments)
Gene expression	Theme(Pro)
Transcription	Theme(Pro)
Protein catabolism	Theme(Pro)
Phosphorylation	Theme(Pro)
Localization	Theme(Pro)
Binding	Theme (Pro), ...
Regulation	Theme(Pro/Ev), Cause(Pro/Ev)
Positive regulation	Theme(Pro/Ev), Cause(Pro/Ev)
Negative regulation	Theme(Pro/Ev), Cause(Pro/Ev)

(Pro: protein, Ev: event)

위 표에서 괄호 속은 이러한 논항을 채울 수 있는 개체의 타입을 나타낸다. 추출 대상이 되는 9가지의 이벤트 종류는 크게 3 종류로 다음과 같이 다시 구분될 수 있다.

(1) 단순 이벤트	논항 1 개만을 가지는 이벤트들로서 그림 2 에서 처음 5가지가 이에 해당한다.
(2) 다중 이벤트	그림2 의 binding 타입 이벤트로서 1개 이상의 Theme 을 가질 수 있다.
(3) 복합 이벤트	Theme 과 Cause 논항을 한 개씩 가질 수 있으며 이 논항들은 단백질 뿐 아니라 이벤트에 의해서도 채워질 수 있다. 결국 이벤트가 이벤트를 논항으로 포함하는 재귀적인 구조가 가능하므로 매우 복잡한 구조를 가지는 이벤트의 생성이 가능하다.

제공되는 훈련, 개발, 테스트 데이타 이외에 별도로 문장분리, 품사태깅, 의존관계트리 정보도 제공된다. 개발자는 시스템의 개발에 이러한 정보를 최대한 이용할 수 있다.

학습용 데이타의 한 문장을 예로 들면 다음과 같다:

"Phosphorylation of TRAF2 inhibits binding to the CD40 cytoplasmic domain."

이 문장에 대하여 제공되는 이벤트 태깅 정보는 다음 네모 안의 내용과 같다. 여기에서 T1, T2 는 단백질을 나타내며 T15, T16, T17 은 이벤트 트리거를 나타낸다. E1, E2, E3 는 이벤트를 나타내며 각 이벤트의 논항정보도 포함한다.

```
T1      Protein 19 24     TRAF2
T2      Protein 49 53     CD40
T15     Phosphorylation 0 15      Phosphorylation
T16     Negative_regulation 25 33  inhibits
T17     Binding 34 41     binding
E1      Phosphorylation:T15 Theme:T1
E2      Negative_regulation:T16 Theme:E3 Cause:E1
E3      Binding:T17 Theme:T1 Theme2:T2
```

예를 들면 E3 는 E2 의 Theme 논항으로 참여하고 있다. 이 태깅 정보는 텍스트 형태로 되어 있어 전체적인 이벤트들의 파악이 어렵다. 이를 다음과 같이 그림으로 표시하면 보다 이해하기 쉽다.

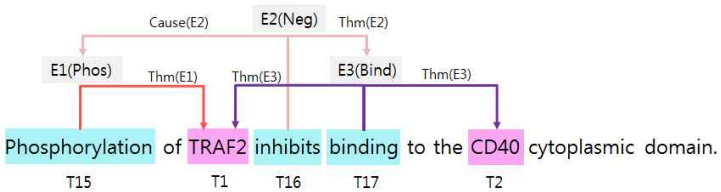


그림 2. 문장의 이벤트 태깅 정보.

4. 바이오 이벤트 시스템 개발

우리는 바이오 이벤트 추출 문제에 대한 시스템의 개발을 위하여 그림 1에 표시된 바와 같이 별도로 분리된 2 단계의 처리를 거치는 기법을 사용한다:

- 단계 1: 이벤트의 발생을 나타내는 단어(열)의 탐지 및 이벤트의 타입의 결정 (ETD-ETC),
- 단계 2: 이벤트의 주요논항을 채우는 참여자의 인식(EPR).

4.1. 이벤트 트리거-타입 탐지를 위한 피쳐 개발

단계 1을 이벤트 트리거 탐지(event trigger detection; ETD)와 이벤트의 타입 결정 (event type classfication; ETC)라는 별도의 두 세부 작업(sub-task)로 나누어 [3] 에서와 같이 접근할 수도 있으나, 우리는 하나의 모듈이 이 두 세부작업을 동시에 처리하는 방법을 택한다. 우리는 이 모듈의 개발을 위해 그림 3 과 같이 다중-클래스(multi-class) SVM을 사용한다.

그림 3에서 보듯이 테스트 데이타 문장의 각 단어에 대하여 이 단어가 이벤트 트리거인지 그리고 그런 경우 그 이벤트의 타입이 무엇인지를 분류해 주는 SVM 기반의 분류기를 사용한다. 이 분류기의 입력은 분류하고자 하는 현재 토큰에 대하여 생성되는 피쳐 리스트 (feature vector)이다. 출력은 1에서 10 사이의 정수로서 10은 트리거가 아님을 나타내며 다른 값은 이벤트의 종류를 나타낸다. 본 논문은 이 ETD/ETC SVM을 위한 효과적인 피쳐집합의 생성에 대하여 자세히 살펴보고자 한다.

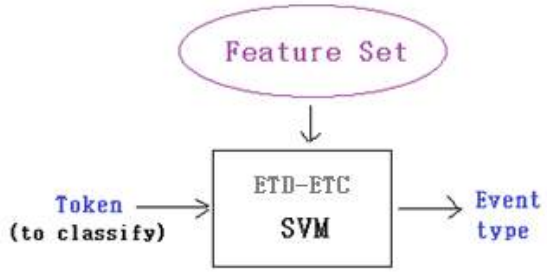


그림 3: SVM에 기반한 이벤트탐지 및 타입인식.

실제로 BioNLP-2009 에 참여한 팀들의 경우 단계 1의 성능이 별로 높지 않았다. [4, 5] 의 경우 단계 1을 위하여 CRF 모델을 사용하였다. 그러나 이 단계에 대한 성능은 F-score 43.3 에 머물고 있다. [3] 의 경우 단계 1은 이벤트 트리거의 존재 여부를 탐지하고(event trigger detection), 그 다음 이벤트의 타입을 결정하는(event type classification) 2 개의 별도의 세부단계로 구성되어 있다. 이들 세부단계는 각각 Maximum entropy 모델을 이용하여 구현되었다. 이 시스템의 성능을 보면 ETD 의 경우 F-score 68.5, ETC 의 경우 85.20 정확도(accuracy)를 가져 단계 1의 결합 성능이 58.32 F-score 를 가지는 것으로 보고되었다.

4.2. ETD-ETC SVM 피쳐 리스트

우리는 다음과 같이 매우 기본적인 피쳐 종류에서 출발하여 점차 다양한 피쳐를 추가함으로써 결과적으로 좋은 성능을 가능케 하는 피쳐 집합을 이용한다.

• BF (Basic features)

대부분의 분류 문제에서 항상 사용하는 가장 기본적인 피쳐집합이다. 현재 토큰 인덱스가 i 라 하자. 즉 Wi 는 현재 토큰(current token)을 나타낸다. 이를 기준으로 좌우-윈도우(Wi, Wi-1, Wi-2, Wi+1, Wi+2) 안의 각 토큰에 대하여 다음과 같이 3 가지의 피쳐를 생성한다.

BF	For each of Wi, Wi-1, Wi-2, Wi+1, Wi+2: prepare features of **String, Stem, POS**

• PF (Protein features)

좌우-윈도우 내의 각 토큰에 대하여 단백질 여부를 나타내는 피쳐이다.

PF	For each of Wi, Wi-1, Wi-2, Wi+1, Wi+2: prepare features of **Is_Protein**

• DF (Dependency features)

실험에 사용되는 모든 데이타 문장에 대한 의존 파스 트리가 제공된다. 따라서 구문구조적인 정보를 이용하기 위하여 우리는 다음과 같은 피쳐를 준비한다. 이 피쳐 집합과 관련하여 주의할 점은 단백질 여부에 대한 피쳐도 생성한다.

DF	For each of LC(closest left child of Wi) RC(closest right child of Wi) LLC(2nd closest left child of Wi) RRC(2nd closest right child of Wi) Par(parent of Wi) Gpar(grandparent of Wi): prepare features of **String, Stem, POS, Is_Protein**

• CCLRC (closest child of left or right child)

현재 토큰의 좌/우 자식의 자식 중 가장 가까운 것이 단백질인지의 여부에 대한 피쳐이다.

CCLRC	For each of CCLC(closest protein child of left child of Wi), CCRC(closest protein chjild of right child of Wi): prepare features of **String, Stem, POS, Is_Protein**

• WCP (wide context protein features)

문장내의 단백질의 개수, 현재 단어 주변의 정해진 크기(현재는 좌우 각 4 단어)의 윈도우 내의 단백질의 개수를 피쳐로 한다.

WCP -w -s	 **Number of proteins in the window** **Number of proteins in the sentence**

• ET (event type of current token)

우리는 훈련 데이타로부터 이벤트 트리거로 사용된 모든 단어의 어근을 수집한다. 그리하여 각 어근에 대하여 이벤트 트리거로 가장 많이 사용된 이벤트 종류를 파악한다. 이에 기반하여 다음 피쳐를 준비한다.

ET	event class number for which current token's stem was used most

4.3. 이벤트 참여자 인식 모듈 개발

단계 2의 작업은 단계 1에서 탐지한 각 이벤트에 대하여 이의 논항이 되는 참여자를 찾는 작업(participant searching; PS)이다. 결국 이 단계에 대한 입력은 "이벤트 트리거" 및 이를 포함하는 "문장"이며, 출력은 이 이벤트의 필수 논항을 채우는 모든 "참여자"이다. 참여자는 이벤트의 타입에 따라서 단백질 또는 다른 이벤트가 될 수 있다.

단계 2의 작업에 대한 접근 방법은 여러 가지가 가능하다. 예를 들면, 단계 1과 마찬가지로 분류의 문제로 접근할 수도 있다. <이벤트트리거, 단백질>, <이벤트트리거, 이벤트> 등과 같이 문장으로부터 가능한 쌍을 구하여 이것을 "참" 또는 "거짓"의 하나로 분류하는 것이다. 이러한 접근 방법의 장점은 기계학습 기법이 제공하는 특징 중의 하나인 높은 recall을 얻을 수 있다는 점이다.

그러나 일차적으로 먼저 우리는 단계 2를 분류 문제로 접근하지 않고 의존트리에 기반한 휴리스틱 규칙을 사용하여 참여자를 탐색하는 기법을 사용하는 접근 방법을 먼저 시도해 보았다. 그 이유는 첫째로 단계 2를 시작하는 시점에서 이벤트 트리거 단어, 이 이벤트를 포함하는 문장, 이 문장의 의존트리, 이 문장 내에 존재하는 모든 단백질에 대한 정보가 사용이 가능하며, 둘째로 술어와 논항 사이에 의존관계가 존재한다는 것이 일반적으로 알려져 있기 때문이다. 즉 이벤트 추출 문제의 경우 다음과 같은 가정이 그럴듯하다고 보기 때문이다:

> 하나의 이벤트와 이 이벤트의 논항을 채우는 참여자는 의존관계 경로로 연결된다.

우리는 학습데이타에 대한 많은 관찰을 통하여 이 가정이 상당히 타당성이 있음을 관찰하였다. 이 가정에 기반한 단계 2의 처리는 결국 이벤트의 종류에 따라 다음과 같은 방법을 사용하기로 한다.

①단순 이벤트	하나의 Theme 논항을 가지며 이는 단백질에 의해 채워진다. 이벤트와 의존관계 경로로 연결된 단백질 하나를 탐색하는 휴리스틱을 개발한다.
②다중 이벤트	1개 또는2개의 Theme 논항을 가지는 경우만을 고려한다. 트리거 단어에 따른 논항의 개수 추정, 의존 경로로 연결된 단백질 중에서 참여자를 탐색하는 휴리스틱을 개발한다.
③복합 이벤트	Theme, Cause 두 가지의 논항이 가능하며 단백질 또는 다른 이벤트가 이를 채울 수 있다. 문장의 의존구조 트리에서 이벤트 노드 주위의 구조를 고려하여 참여자를 찾는 휴리스틱을 개발한다.

[5]의 연구에 따르면 의존관계 구문구조를 효과적으로 이용한다면 단계 2 만의 성능은 상당히 높을 것으로 추정된다. 따라서 우리의 전략은 좋은 SVM 피쳐집합의 선정을 통하여 가능하면 단계 1의 성능을 높이고자 하였다.

5. 실험 결과 및 평가

5.1. 이벤트 트리거 및 타입 탐지 (ETD-ETC) 성능측정

우리는 먼저 전체 시스템의 성능을 측정하는 대신에 단계 1 즉 이벤트 발생 탐지 및 이벤트 타입 결정 작업에 대하여 성능을 측정하여 보았다 (표 2 참조). 우리의 가정은 단계 1의 성능이 전체 시스템의 성능에 큰 영향을 미친다고 보기 때문이다. 앞에서 설명한대로 단계 1에 대한 모듈은 SVM 을 사용하였다. 그래서 이 모듈의 성능은 SVM에 사용되는 피쳐에 매우 큰 영향을 받게 된다. 앞에서 소개한 여러 피쳐 집합의 효과를 측정하기 위해 우리는 점증적인 실험을 수행하였다. 즉 맨 처음에는 가장 기본적인 피쳐집합인 BF를 사용하여 시스템을 구성하고 이에 의한 단계 1 모듈의 성능을 측정하였다 (표의 첫 번째 행). 다음 행은 앞의 시스템에 대하여 PF 피쳐를 추가하여 보다 증가된 피쳐집단으로 시스템을 구성하고 성능을 측정하였다. 이와 같은 방법으로 점진적으로 피쳐를 추가하여 가면서 성능을 측정하였다.

표 2. 단계 1에 대한 성능 측정 결과.

피쳐	Recall	Precision	F-score
BF	43.63	52.48	47.65
BF+ **PF**	47.76	55.37	51.28
BF+PF+ **DF**	46.53	62.13	53.21
BF+PF+DF+ **CCLRC**	52.17	61.31	56.37
BF+PF+DF+CCLRC+ **WCP**	50.07	68.45	57.84
BF+PF+DF+CCLRC+ WCP+ **ET**	59.19	64.77	61.85

위의 단계 1의 모듈에 대한 평가는 개발 데이타 (development data)를 테스트 데이타로 이용하였다. 그 이유는 배포한 테스트 데이타는 이벤트에 대한 태깅이 되어 있지 않기 때문이다. 배포된 테스트 데이타를 이용하지 못한 이유는 테스트 데이타를 이용한 시스템 평가는 전체 시스템에 대한 평가만 가능하기 때문에 단계 1만에 대한 평가는 수행할 수 없기 때문이다. 이러한 문제로 인하여 우리는 훈련 데이타를 두 부분으로 나누어 일부를 훈련에 나머지 일부를 튜닝에 사용하고, 배포된 개발 데이타를 테스트 데이타로 이용하였다.

실험결과 우리 시스템의 단계 1 모듈의 성능은 61.85의 F-score로 측정되었다. 우리 시스템과 유사하게 구분된 단계 1을 사용하는 두 개의 시스템으로 [4] 와 [5]가 있다. 이들 시스템의 경우 CRF를 사용하여 이 모듈을 개발하였다. 그들 시스템의 단계 1에 대한 성능은 43.3의 F-score로 관찰되었다. [3] 의 경우 단계 1을 두 개의 세부 단계로 나누어 구현하였는데 단계 1에 해당하는 성능이 58.32 F-score를 가지는 것으로 보고되었다. 이러한 결과에 따르면 우리 시스템의 단계 1 모듈이 다른 시스템에 비해 좋은 성능을 보임을 알 수 있다. 즉 우리가 제시하는 피쳐들을 사용하는 경우 좋은 시스템을 달성할 수 있다고 할 수 있다.

5.2. 이벤트 참여자 인식(EPR) 성능측정

우리는 단계 2 즉 EPR 에 대한 성능을 별도로 측정하지 않고 단계 1과 단계 2를 합친 전체 시스템의 성능을 측정하여 그 결과를 표 3에 나타내었다. 특히 이벤트의 종류에 따른 이벤트 추출 문제의 난이도를 알아보기 위해 각 종류에 따른 성능도 표시하였다.

표 3. 전체 시스템 성능.

	GS	Sys	GS∩Sys	R	P	F-s
단순 이벤트	559	435	269	48.12	61.84	54.12
다중 이벤트	249	265	76	30.52	28.68	29.57
복합 이벤트	987	663	19	1.93	2.87	2.30
전체 평균	1795	1363	364	20.28	26.71	23.05

GS: 정답집합, Sys: 시스템 출력, GS∩Sys: true positive, R: recall, P: precision, F-s: F-score.

위 표를 보면 단순 이벤트의 경우는 단계 1의 성능에 별로 떨어지지 않는 전체 성능을 보이는 것으로 볼 때 EPR 모듈만의 성능은 상당히 좋은 것을 알 수 있다. 그러나 복합 이벤트의 경우 2.30 F-score의 성능을 가지는데 이는 이 종류의 거의 모든 이벤트의 추출에 실패함을 알 수 있다. 특히 복합 이벤트의 수가 전체 이벤트들에서 차지하는 비중이 55%를 차지하므로 전체 시스템의 성능을 23.05로 저하시키게 되었다.

이벤트 추출 시스템의 전체 성능 향상을 위해서는 각 이벤트 종류별로 성능 향상을 추구하여야 하지만 특히 아주 낮은 성능을 가지는 복합 이벤트와 관련하여 보다 개선된 의존트리 기반 휴리스틱의 기법을 개발하여야 할 것이다.

6. 결론

본 논문에서는 바이오분야 문서에서 바이오 이벤트의 발생 및 그 타입 그리고 이벤트의 논항을 채우는 참여자의 인식을 수행하는 시스템 개발에 대한 문제를 다룬다. 우리가 제안한 시스템은 이벤트 발생의 탐지 및 그 타입의 인식 (ETD-ETC)을 위한 별도의 단계를 가지는데 여기에서는 다중-클래스 분류에 적합한 SVM을 이용한다. 우리는 여러 가지 다양한 피쳐를 사용함으로써 높은 성능을 가지는 ETD-ETC 단계를 위한 모듈의 개발이 가능함을 보였다. 이벤트 논항의 참여자 인식 (EPR) 모듈의 경우

현재로서는 의존트리 기반 휴리스틱에 의한 탐색 기법을 사용하여 보았다. 그러나 보다 더 높은 성능을 위해서는 기계학습 기법 등 다른 기법을 시도할 필요가 있음을 알게 되었다.

참고문헌

[1] J. Kim, T. Ohta, S. Pyysalo, Y. Kano and J. Tsujii, "Overview of BioNLP'09 Shared Task on Event Extraction," Proceedings of the Workshop on BioNLP: Shared Task, pages 1-9, 2009.

[2] J. Kim, T. Ohta and J. Tsujii, "Corpus annotation for mining biomedical events from literature," BMC Bioinformatics 9:10, 2008.

[3] H. Lee, H. Cho, M. Kim, J. Lee, G. Hong and H. Rim, "A Multi-Phase Approach to Biomedical Event Extraction," Proceedings of the Workshop on BioNLP: Shared Task, pages 107-110, 2009.

[4] G. Georgiev, K. Ganchev, V. Momtchev, D. Peychev, P. Nakov and A. Roberts, "Tunable Domain-Independent Event Extraction in the MIRA Framework," Proceedings of the Workshop on BioNLP: Shared Task, pages 95-98, 2009.

[5] F. Sarafraz, J. Eales, R. Mohammadi, J. Dickerson, D. Robertson and G. Nenadic, "Biomedical Event Detection using Rules, Conditional Random Fields and Parse Tree Distances," Proceedings of the Workshop on BioNLP: Shared Task, pages 115-118, 2009.

질의 응답 시스템을 위한 반교사 기반의 정답 유형 분류

박선영O, 이동현, 김용희, 류성한, 이근배
포항공과대학교, 컴퓨터공학과
{sypark322, semko, ttti07, ryush, gblee}@postech.ac.kr

Semi-Supervised Answer Type Classification For Question-Answering System

Seonyeong ParkO, Donghyeon Lee, Yonghee Kim, Seonghan Ryu, Gary Geunbae Lee
Department of Computer Science and Engineering, POSTECH

요 약

기존 연구에서는 질의 응답 시스템에서 정답 유형을 분류하기 위해 패턴 매칭 방식이나 교사 학습(Supervised Learning)을 이용했다. 패턴 매칭 방식은 질의 분석을 통해 수동으로 패턴을 구축해야 한다. 교사 학습에서는 훈련 데이터 전체에 정답 유형이 태깅(Tagging)되어야 하며, 이를 위해서는 사용자의 질의에 정답 유형을 수동으로 태깅하는 작업이 많이 필요하다. 웹을 통해 정답 유형이 태깅되지 않은 대용량의 사용자 질의 말뭉치를 구할 수 있지만, 이 데이터에는 정답 유형이 태깅되어 있지 않다. 따라서, 대용량의 사용자 질의에 비례하여, 정답 유형을 수동으로 태깅하는 작업량이 증가한다. 앞서 언급한 두 가지 방법론에서, 정답 유형 분류를 위해 수작업이 많이 필요하다는 문제점을 해결하고자 본 논문에서는 일부 태깅된 훈련 데이터를 필요로 하는 반교사 학습(Semi-supervised Learning)에 기반한 정답 유형 분류를 제안한다. 이는 정답 유형 분류 작업에 필요한 노동력을 최소화함으로 대용량의 데이터를 통한 효율적 질의 응답 시스템 구축을 가능하게 한다.

주제어: 정답 유형, 질의 응답 시스템, 잠재 디리쉴레 할당(Latent Dirichlet Allocation, LDA)

1. 서론

질의 응답 시스템은 많은 양의 정보를 바탕으로 사용자의 질문에 정확한 답을 찾아주는 시스템이다. 질의 응답 시스템은 기존의 검색엔진과 다르게, 불필요한 정보를 제외하고, 사용자가 찾고자 하는 정보만을 제공한다는 장점이 있다. 따라서, 질의 응답 시스템이 제공하는 서비스는 정보 검색의 궁극적인 목표와 부합하며, 빅 데이터 시대에 정보의 효율적 사용이 필요하다는 측면에서 각광받고 있다. 이러한, 질의 응답 시스템 개발은 해외에서 뿐만 아니라 국내에서도 중요한 이슈로 떠오르고 있다.

질의 응답 시스템은 크게 질의 분석 단계, 정답과 관련된 문서 추출 단계, 문서로부터 정답을 추출하는 단계로 나눠져 있다. 질의 분석 단계에서는 정답과 관련된 문서검색을 위하여, 사용자의 질의에서 키워드, 정답 유형 등을 추출한다. 질의 응답 시스템에서 정답 유형이란 사용자가 질의를 통해 찾고자 하는 정답의 유형을 말한다. 따라서, 질의 분석 단계에서 분석된 정답 유형은 문서에서 정답을 찾을 때, 검색 제약 조건으로 활용된다.

사용자 질의에서 정답 유형을 추출하는 작업은 질의 응답 시스템에 필수적인 요소이다. 대부분의 질의 응답 시스템 개발에서 정답 유형 정보를 활용하고 있다[1-8]. 정답 유형 결정은 n가지 정답 유형들 중 하나의 유형으로 분류하는 문제로 정의 되어 왔다. 기존의 시스템 개발에서는 사용자의 질의에서 정답 유형을 결정하기 위해 교사학습을 이용하거나 패턴과 규칙을 활용했다[1-8]. 표 1은 정답 유형과 질의에 대한 예를 보여준다.

표 1 정답 유형의 예

정답 유형	질의
Animal	What is the Ohio state bird?
Human	Who wrote "the divine Comedy"?
City	What is the capital of Mongolia?
...	...

패턴과 규칙에 기반한 방법을 사용하기 위해서는 사용자 질의를 분석하여 수작업으로 패턴을 구축해야 한다. 최근의 질의 응답 시스템에서는 패턴과 규칙만을 이용하여 정답 유형 분류를 하지 않고, 통계 모델을 함께 사용하는 경우가 대부분이다[2-6,8]. 통계 모델을 사용하는 최근의 정답 유형 분류 연구는 교사 학습을 이용한다[2~6,8]. 교사 학습에서 정답 유형 분류 모델 훈련을 위해서는 각 사용자 질의에 정답 유형이 모두 태깅된 훈련 데이터가 필요하다. 사용자 질의 데이터는 웹을 통해서 쉽게 수집할 수 있지만, 사용자 질의에 정답 유형이 태깅된 데이터는 구하기 어렵다. 따라서 기존 연구에서는 사용자 질의에 정답 유형을 수동으로 태깅하여 훈련 데이터를 제작했다[2-6,8].

정답 유형 수동 태깅에 대한 작업량을 줄이고자, 본 논문에서는 교사 학습 방법 대신에 반교사 학습 방법을 이용하여 정답 유형을 분류하였다. 반교사 학습은 일부만 태깅된 훈련 데이터를 통해 정답 유형 분류 모델 제작이 가능하다. 본 논문에서는 반교사 학습 기반의 잠재 디리쉴레 할당(Semi-Supervised Latent Dirichlet

Allocation, Semi-Supervised LDA)을 이용하였다. 일부 태깅된 데이터를 이용하여 정답 유형을 분류하였고, 정확도를 측정하였다. 또한 같은 양의 태깅된 훈련 데이터를 이용하였을 때 교사 학습 방법에 의한 정답 유형 분류보다 정확도가 높다는 결론을 얻었다.

본 논문의 구성은 다음과 같다. 2장에서는 관련 연구를 소개하고, 3장에서는 방법론 소개와 본 논문에서 제안하는 시스템 구조 및 정답 유형 분류 방법에 대해 설명한다. 4장에서는 실험과정에 대해 서술하고, 결과를 분석하였다. 마지막으로, 5장에서는 결론 및 향후 연구에 대해 기술하였다.

2. 관련 연구

정답 유형이라는 개념은 1999년 TREC(Text REtrieval Conference)-8에 출전했던, 질의 응답 시스템 LASSO에 의해 처음으로 도입되었다[1]. 이후 많은 질의 응답 시스템에서 정답 유형 분류 단계를 질의 응답 시스템 개발에서 활용하였다[1-8].

2001년 TREC-10에 출전했던 질의 응답 시스템 SiteQ[7]는 2단계 계층 구조를 가지는 정답 유형 분류 체계를 구성하였다. 또한 어휘 의미 패턴(Lexical-Semantic Patterns, LSP)을 이용하여 정답 유형을 결정하였다. 사용자의 질의를 미리 정의한 LSP에 대응시켜서 정답 유형을 결정하였다. 예를 들면,(%who)(%be)(@person)->PERSON이라는 LSP가 있다. “Who was president Cleveland's wife?"라는 예문은 (%who)(%be)(@person)라는 LSP에 대응하기 때문에 예문의 정답 유형은 PERSON이 된다. 이 방법론을 적용하기 위해, 기존 QA Track에 사용한 질문들과 웹 데이터를 수집하여 수작업으로 361개의 LSP를 구성하는 노력이 선행되었다. 2007년도에 TREC에 출전한 Ephyra라는 질의 응답 시스템은 154개의 정답 유형을 사용했으며,[6]과 마찬가지로 계층 구조를 가지는 정답 유형 분류 체계를 구성했다. TREC에서 사용한 질문들을 반영하여 정답 유형 클래스를 구성했다. 정답 유형 분류에 대해 비중 있게 다루고 있으며, 규칙 기반의 방법과 통계적 학습 모델 방법 두 가지를 하이브리드 하여 사용하였다[3].

ETRI에서는 [3]의 Ephyra 시스템과 마찬가지로, 구조적 자질 벡터 기계(structured Support Vector Machine)와 규칙에 기반한 방법을 하이브리드(Hybrid)하였다[5]. 통계 방법을 적용하기 위한 학습데이터를 제작하기 위해, 약 82000개 질문에 대해 정답 유형을 수동 태깅하는 노동력이 필요했다. 이 때 사용한 질문데이터는 국내 지식검색 사이트에서 추출한 것이다. 이처럼, 정답 유형이 태깅된 데이터를 구하는 것은 쉽지 않기에 자체적으로 수동 태깅을 한 경우가 대다수이다. 또한 정답 유형이 태깅된 데이터는 영어 외에 기타 언어에서는 양적인 측면에서 더욱 부족한 실정이므로, 영어 데이터를 번역하여 훈련 데이터로 사용한 경우도 있었다[2].

앞서 언급했던 패턴 매칭 방법이나 교사학습을 이용한 정답 유형 결정 방법은 다량의 수작업을 필요로 한다. 본 논문에서는 반교사 학습을 통해 정답 유형 분류에 필요한 노동력을 최소화하는 방법을 제안한다.

3. 반교사 정답 유형 추출 방법

3.1 LDA를 이용한 정답 유형 분류

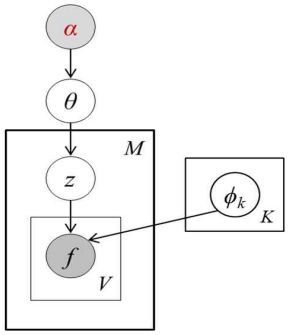


그림 1 정답 유형 분류를 위한 LDA의 도식

LDA는 2003년 M.Blei에 의해 제안된 비교사 학습(Unsupervised Learning)으로 텍스트 말뭉치와 같은 이산형 데이터(Discrete data) 집합에 대한 생성 확률 모델(Generative Probabilistic Model)이다. LDA는 베이지안 모델(Bayesian model)로써, 생성 확률 모델은 확률과 파라미터로부터 데이터가 생성된다는 관점을 나타낸 것이다. 정답 유형을 결정하기 위해서 LDA를 적용할 때는, 기존의 정답 유형 분류와 다르게, 정답 유형을 기준으로 클러스터가 생성된다고 본다. 즉, 정답 유형으로부터 여러 자질들로 표현되는 질의가 생성되는 것으로 볼 수 있다. 관련 연구로는 사용자의 발화 의도로부터 발화를 구성하는 단어들이 생성된다는 관점에서의 연구가 있었다[9].

정답 유형의 분포를 나타내는 θ는 하이퍼 파라미터(Hyper Parameter) α의 디리슐레 프로세스(Dirichlet Process)로부터 생성 된다. 상태(State),즉 클러스터에 해당하는 z는 정답 유형을 나타낸다. z는 θ의 다항 분포(Multinomial distribution)를 따른다. 각 문장에서 추출한 자질들을 나타낸 f는 정답 유형 z와 파라미터 ϕ_k의 다항분포에 의해 생성 된다. 본 논문에서는 일부 태깅된 데이터를 이용하여 Semi-Supervised LDA를 정답 유형 분류에 적용한다.

3.2 제안하는 질의 응답 시스템

본 논문에서는 DBpedia[1] 나 YAGO[2](Yet Another Great Ontology) 와 같이 구조화된 DB를 활용한 질의 응답 시스템을 제안한다. 위키피디아에서 구조적인 정보를 추출하여 공개적으로 제공하는 DBpedia를 활용하였고, 위키피디아와 워드넷(WordNet)에서 추출한 온톨로지인 YAGO도 DB로 활용할 계획이다. 뿐만 아니라 FreeBase[3]

1) http://dbpedia.org
2) http://www.mpi-inf.mpg.de/yago-naga/yago/
3) http://www.freebase.com/

나 기타 구조화된 DB를 추가하여 DB를 확장하는 연구를 진행하고 있다. 궁극적으로는 웹 기반 오픈 도메인 질의 응답 시스템으로 개발을 확장할 것이다. 본 논문에서 제안하는 질의 응답 시스템은 그림 2와 같다.

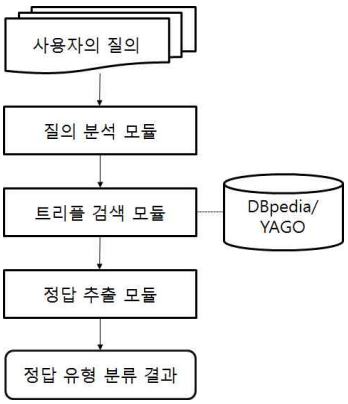


그림 2 제안하는 질의 응답 시스템 구조도

3.2.1 질의 분석 모듈

사용자의 질의에서 <Subject-Property-Object> 형태의 트리플을 추출한다. 사용자의 질의에서 사용자가 찾고자 하는 정보인 질문 초점(Question Focus)을 찾는다. 질문 초점은 예를 들어 "What is the **largest city** in Germany?" 라는 문장이 있을 때, Largest city가 된다. 질문 초점을 Subject또한 Object로 추출하고, 질문 초점의 유형으로, 정답 유형을 추출한다. 정답 유형을 결정하여 최종적으로 정답을 추출하는데 활용한다. Parser와 relation Extractor를 혼용하여 Property 와 Subject 또는 Object를 추출한다.

3.2.2 트리플 검색 모듈

사용자의 질의에서 <Subject-Property-Object> 형태의 트리플과 정답 유형을 추출하여 DBpedia 및 YAGO 와 같은 구조화된 DB에서 검색한다. Property와 Object 또는 Subject가 일치하고 정답 유형이 일치하거나, 정답 유형의 하위 유형(정답 유형이 사람일 때, 정답 유형 분류 체계가 사람-가수일 때, 검색 대상이 가수인 경우)이 일치하거나, 상위 유형(정답 유형이 가수일 때, 검색 대상이 사람인 경우)이 일치하는 트리플들을 검색한다.

3.2.3 정답 추출 모듈

정답 유형과 Subject, Property, Object 등을 이용하여 Scoring measure를 통해 순위를 정한다. 최종적으로 가장 정답일 확률이 높은 것을 정답으로 추출한다.

3.3 제안하는 정답 유형 분류

본 논문에서는 일부 태깅된 데이터를 이용한 Semi-Supervised LDA를 적용하여 정답 유형을 분류한다. 이를 통해 정답 유형 태깅에 대한 노동력 절감 효과와 수동 태깅에서 비롯된 모호성 문제를 해결할 수 있다. 본 논문에서 제안하는 정답 유형 분류는 그림 3과 같다.

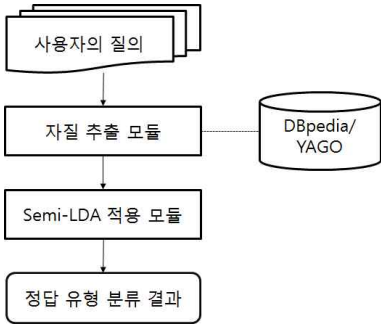


그림 3 제안하는 정답 유형 분류

3.3.1 자질 추출 모듈

LASSO 시스템에서 정답 유형은 의문사를 이용한 질문 유형(Question Type)과 질문 초점(Question Focus)을 통해 결정한다[1]. 따라서, 의문사와 질문 초점을 자질로 사용했다. 그 외에 자질로는 사용자 질의의 본동사(main verb)가 있다. 본동사는 사용자의 질의 의도를 반영하는 경우가 많기 때문에 자질로서 추출했으며, TREC에서 우수한 성능을 보인 Ephyra 시스템[3]에서도 추출하였다. 본동사 추출을 위해서 Stanford Parser[1]를 이용하였다. 또한 질의에서 의문사의 앞 뒤 어휘 정보를 이용하였다. 이를 통해 how many, how much 등을 구별할 수 있다. 그 외에 어휘 정보로는 다른 품사보다 문장에서 중요한 역할을 하는 명사, 동사에 해당하는 어휘 정보를 사용하였다. 또한 DBpedia를 활용하여 질의에 고유 명사가 있는 경우 DBpedia에서 제공하는 <개체명(Named-Entity),개체 유형(Named-Entity Type)> 데이터에서 검색하여 고유명사의 개체 유형 정보를 자질로 이용하였다.

3.3.2 Semi-LDA 적용 모듈

앞서, 추출된 자질을 바탕으로 Semi-Supervised LDA를 적용하여 정답 유형을 분류하였다. 정답 유형 분류는 구조화된 DB에서의 검색을 위해 위키피디아와 Wordnet을 연동한 YAGO의 분류 체계를 활용하였다. YAGO의 분류 체계는 약 1,700,000가지 개체들로 구성되어 있다. UIUC의 정답 유형은 이에 모두 1:1 대응 가능하다. 뿐만 아니라 UIUC의 other과 같은 유형은 더 세분화 가능하여 대응 가능하다. 일부 태깅된 데이터를 포함하여 클러스터링 하기 때문에 각각의 클러스터 ID에 해당하는 정답 유형이 정해져 있다. 정답 유형 분류에 대한 성능은 실험에서 기술하였다.

4. 실험

4.1 실험 설계

정답 유형 분류에서 많이 사용하고 있는 교사학습 방법과 본 논문에서 제안하는 반교사 학습 방법을 비교한다. 반교사 학습은 교사 학습과 다르게 일부 태깅 데이터를 통해 학습이 가능하다는 장점이 있음으로 이 부분을 검증한다.

교사 학습 방법으로 CRF(Conditional Random Fields)

1) http://nlp.stanford.edu/software/lex-parser.shtml

를 이용한다. 정답 유형 분류에 대한 CRF와 Semi-Supervised LDA의 정확도를 측정한다. 전체 학습 데이터에서 정답 유형이 태깅된 데이터의 비율에 따른 각 알고리즘의 성능을 측정한다. 랜덤으로 태깅 데이터를 추출하며, 교사 학습인 CRF와의 성능 비교를 위해 각 정답 유형이 1회 이상 훈련 데이터에 반영되도록 한다. 각각의 정답 유형에 대한 성능을 측정하여 성능이 높은 정답 유형과 성능이 낮은 정답 유형을 비교 분석한다.

4.2 데이터

UIUC(University of Illinois at Urban-Champaign)가 정답 유형 분류 실험에서 사용한 약 5500개 질의를 훈련 데이터로 사용하였다. 테스트 데이터로는 TREC 10에서 사용한 500개를 사용하였다. UIUC에서 정답 유형 분류 실험에 사용한 데이터는 기존의 연구들에서도 많이 사용한 공신력 있는 데이터이다[2]. TREC에서 제공하는 테스트 데이터도 질의 응답 시스템 관련 연구에서 성능 측정의 목적으로 활발히 활용하고 있다[2].

본 논문에서는 실험 결과를 실제 질의 응답 시스템 개발에 활용하는 것을 목적에 두고 있다. 따라서, 위키피디아 기반 질의 응답 시스템 제작을 위해 YAGO의 분류 체계와 UIUC의 정답 유형을 대응하는 작업을 선행하였다. 숫자나 날짜 등 어떤 글자도 YAGO 온톨로지에서 개체로 표현되어 있다. 따라서, YAGO의 분류 체계에 UIUC의 각 정답 유형이 모두 대응 가능함으로 UIUC의 50가지 정답 유형 분류 체계와 동일한 분류 체계를 이용하였다.

표 2 UIUC에서 제공하는 정답 유형 분류 체계

Coarse Class	Fine Classes
ABBREV.	abbreviation, expression
ENTITY	animal,body,color,creative,currency,disease,event,food,instrument,lang,letter,other,plant,product,religion,sport,substance,symbol,technique,term,vehicle,word
DESCRIPTION	definition,description,manner,reason
HUMAN	group,individual,title,description
LOCATION	city,country,mountain,other,state
NUMERIC	code,count,date,distance,money,order,other,period,percentage,speed,temp,volumesize,weight

4.3 실험 결과 및 분석

표 3은 훈련 데이터 중 정답 유형이 태깅된 데이터의 비율에 따라 Semi-Supervised LDA와 CRF의 정확도를 비교한 것이다. Semi-Supervised LDA는 정답 유형이 태깅된 데이터와 태깅되지 않은 데이터를 모두 훈련에 사용하였다. CRF는 훈련 데이터에 모두 정답 유형이 태깅되어야 하기 때문에, 태깅된 데이터만 훈련 데이터로 사용하였다. 표 3에 따르면, 태깅된 데이터가 10% 일 때, 즉 5000개 이상의 훈련 데이터 중에서 500개 이상의 태깅된 데이터를 포함하여 클러스터링 하였을 때, CRF보다 높은 성능을 보인다. 이 결과는 반교사 학습을 통해 정답 유형을 분류하는 것이 정답 유형 태깅 노동력을 줄일 수 있다는 가능성을 나타낸다. 25% 태깅된 경우는 거의 동일한 성능을 보이며, 50%, 60%, 75%, 100% 태깅 되었을 경우에는 CRF와 같거나 높은 성능을 보인다. 실험 결과에 대한 분석을 위해, 높은 정확도를 보인 정답 유형들과 낮은 정확도를 보인 정답 유형들을 분석하였다.

표 3 태깅된 데이터 비율에 따른 정확도 비교

Percentage	Semi-Supervised LDA	CRF
1%	0.036	0.098
10%	0.376	0.252
25%	0.422	0.444
50%	0.410	0.410
60%	0.528	0.450
75%	0.578	0.450
100%	0.630	0.454

Semi-Supervised LDA을 통해 정답 유형을 분류하였을 때, 수량(NUM_COUNT), 날짜(NUM_date), 사람(HUM_ind)에 해당하는 정확도가 높았다. 반면, 스포츠(ENTY_sport)나 화폐(ENTY_currency), 교통수단(ENTY_veh), 사건(ENTY_event) 등은 정확도가 낮았다.

정확도가 높았던 정답 유형들을 특징들을 분석한다. 첫째, 정확도가 가장 높았던 정답 유형인 수량은 "How many ~" 로 시작하는 경우가 대부분이라는 특징을 보였다. “How many” 또는 “How much" 는 정답 유형이 수량인 경우를 제외하고 거의 쓰이지 않은 어휘 정보이다. 둘째, 높은 정확도를 보였던 날짜 역시 주로 "When was" 로 시작되는 문장 유형이 많았고, 질문 초점인 “year"가 빈번하게 나타났다. 또한, 정답 유형이 날짜로 클러스터된 질의의 자질 중에 본동사가 was인 경우가 대부분이다. 셋째, 정답 유형이 사람인 경우도 의문사가 대부분 “who" 라는 특징이 두드러지며, 본동사로 "was" 나 "is"가 포함된 문장이 많다. 또한, 문장 내에서 고유 명사를 포함하는 경우가 많다. 뿐만 아니라, 전체 훈련 데이터에서 정답 유형이 사람인 질의가 차지하는 비중이 약 1/5 이상으로 높다.

정확도가 낮았던 정답 유형들 중 교통수단에 대한 정확도를 분석한다. 실제 정답 유형은 교통수단이지만, 정답 유형이 사람, 그룹, 날짜, 동물 등 다양한 정답 유형으로 클러스터링 된 경우가 많다. 예를 들어 정답 유형이 사람인 것에 클러스터링 된 경우는 “What was the name of the plane Lindbergh flew solo across the Atlantic ?”가 있는데 본동사와 고유 명사와 같은 자질에 영향을 받았을 것으로 분석된다. 클러스터링은 “빈익빈 부익부” 의 성격을 갖기 때문에, 새로운 데이터는 기존에 형성된 클러스터의 크기에 비례하여 클러스터링 된다. 즉, 새로운 데이터는 크기가 큰 클러스터에 속할 확률이 크다[10]. 실제로 크기가 큰 클러스터에 다른 정답 유형을 가지는 질의가 속한 경우는 많았다. 하지만, 다른 정답 유형을 가지는 질의가 교통수단 클러스터처럼 작은 클러스터에 속한 경우는 거의 없었다. 즉, 훈련 데

이터에서 큰 비중을 차지하는 사람과 같은 정답 유형은 사람이라는 클러스터에 다른 정답 유형들이 함께 포함되어 성능이 떨어지는 경향이 있다. 반면에, 훈련 데이터에서 작은 비중을 차지하는 교통수단과 같은 정답 유형은 교통수단에 속하는 질의가 다른 클러스터들에 속하게 되어 정확도가 떨어지는 경향을 보였다. 아래 그림 4는 훈련 데이터에서 큰 비중을 차지하는 정답 유형이 정확도가 높은 예들을 보여준다. 가로축은 각 정답 유형이 훈련 데이터에서 나타나는 빈도수이며 세로축은 각 정답 유형의 정확도이다. 훈련 데이터의 태깅 비율이 달라도 데이터의 분포는 그림 4와 비슷한 양상을 띤다.

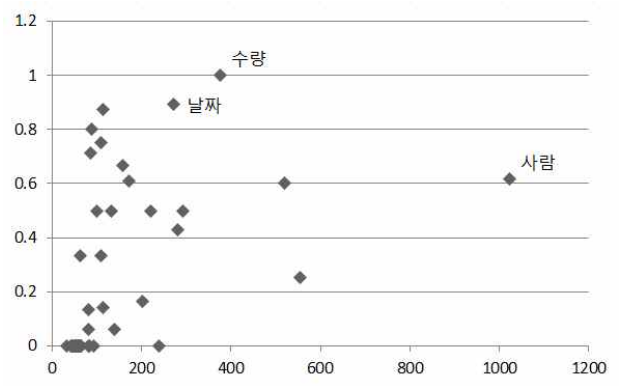


그림 4 50% 태깅 데이터로 훈련했을 때, 훈련 데이터에서 차지하는 비중에 따른 정확도 분포

5. 결론 및 향후 연구

본 논문의 실험을 통해 동일한 양의 태깅 데이터가 주어졌을 때, Semi-supervised LDA가 CRF보다 대체적으로 높은 성능을 보이는 것을 확인할 수 있었다. Semi-supervised LDA는 태깅된 데이터와 태깅이 안된 데이터를 함께 이용할 수 있지만, CRF는 태깅된 데이터만 학습에 이용할 수 있다. 따라서, 반교사 기반의 정답 유형 분류 방법을 통해 정답 유형 태깅 노동력을 줄일 수 있다는 결론을 얻었다.

하지만 임의의 질의 말뭉치들을 정답 유형 별로 클러스터링 할 때, 전체 데이터에서 차지하는 비중이 작은 클러스터들의 정확도를 향상시키는 연구가 필요하다.

또한 웹에서 수집한 질문의 경우 정답 유형이 정해져 있지 않다. 이러한 대용량의 데이터에 정답 유형을 분류하기 위해서 계층적 드리쉴레 프로세스(Hierarchical Dirichlet Process ,HDP)를 이용한 연구를 진행 중이다. HDP를 통해 분류된 각각의 클러스터들을 YAGO의 분류 체계에 대응시키는 연구를 진행할 것이다.

뿐만 아니라, 본 논문의 연구결과를 활용하여, 정답 유형 레이블이 부족한 한국어 질의 데이터를 자동 레이블링할 것이다. 이를 바탕으로, 한국어 질의 응답 시스템 개발에 대한 연구를 진행할 것이다.

*본 연구는 미래창조과학부 및 한국산업기술평가관리원의 산업융합원천기술개발사업(정보통신)의 일환으로 수행하였음. [10044508 , 비기호적 기법 기반 인간 모사형 자가학습 지능 원천기술 개발]

참고문헌

[1] Dan Moldovan et al., "LASSO: A Tool for Surfing the Answer Net," TREC, Vol.8, p.65-73, 1999

[2] Anne-Laure Ligozat, "Question Classification Transfer," Proceedings of the Association for Computational Linguistics, p.429-433, 2013

[3] Nico Schlaefer et al., "Semantic Extensions of the Ephyra QA System for TREC 2007," TREC, 2007

[4] Xin Li et al., "Learning question classifiers," Proceedings of the international conference on Computational linguistics", Vol.1, p.1-7, 2002

[5] 허정 외, "오픈 도메인 질의응답을 위한 검색문서 제약 및 정답유형 분류기술," 정보과학회논문지:소프트웨어 및 응용, 제39권, 제2호, 2012

[6] Xin Li et al., "The role of semantic information,"Natural Language Engineering, Vol.12, no.3, p.229-249, 2006

[7] Gary Geunbae Lee et al., SiteQ: Engineering High Performance QA System Using Lexico-Semantic Pattern Matching and Shallow NLP," TREC, 2001

[8] 송일현 외 , 실시간 검색어를 이용한 주제어 기반의 질의 응답 시스템, 제 23회 한글 및 한국어 정보처리 학술대회, 2011

[9] Donghyeon Lee et al., "Unsupervised modeling of user actions in a dialog corpus," Proceeding of the IEEE international conference on acoustics, speech, and signal processing, 2012

[10] David Blei et al., "The nested chinese restaurant process and bayesian nonparametric inference of topic hierarchies," Journal of the ACM, Vol.57, no.2 , 2010

제25회 한글 및 한국어 정보처리 학술대회
25th Annual Conference on Human and Cognitive Language Technology

구두발표 2A

상품 평가 텍스트에 암시된 사용자 관점 추출

장경록○, 이강욱, 맹성현
한국과학기술원
kyoungrok.jang@kaist.ac.kr, chaximeer@kaist.ac.kr, myaeng@kaist.ac.kr

Extracting Implicit Customer Viewpoints from Product Review Text

Kyoungrok Jang○, Kangwook Lee, Sung-Hyon Myaeng
Korea Advance Institute of Science and Technology

요 약

온라인 소비자들은 amazon.com과 같은 온라인 상점 플랫폼에 상품 평가(리뷰: review) 글을 남김으로써 대상 상품에 대한 의견을 표현한다. 이러한 상품 리뷰는 다른 소비자들의 구매 결정에도 큰 영향을 끼친다는 관점에서 볼 때, 매우 중요한 정보원이라고 할 수 있다. 사람들이 남긴 의견 정보(opinion)를 자동으로 추출하거나 분석하고자 하는 연구인 감성 분석(sentiment analysis)분야에서 과거에 진행된 대다수의 연구들은 크게는 문서 단위에서 작게는 상품의 요소(aspect) 단위로 사용자들이 남긴 의견이 긍정적 혹은 부정적 감정을 포함하고 있는지 분석하고자 하였다. 이렇게 소비자들이 남긴 의견이 대상 상품 혹은 상품의 요소를 긍정적 혹은 부정적으로 판단했는지 여부를 판단하는 것이 유용한 경우도 있겠으나, 본 연구에서는 소비자들이 '어떤 관점'에서 대상 상품 혹은 상품의 요소를 평가했는지를 자동으로 추출하는 방법에 초점을 두었다. 본 연구에서는 형용사의 대표적인 성질 중 하나가 자신이 수식하는 명사의 속성에 값을 부여하는 것임에 주목하여, 수식된 명사의 속성을 추출하고자 하였고 이를 위해 WordNet을 사용하였다. 제안하는 방법의 효과를 검증하기 위해 3명의 평가자를 활용하여 실험을 하였으며 그 결과는 본 연구 방향이 감성분석에 있어 새로운 가능성을 열기에 충분하다는 것을 보여주었다.

주제어: 전자상거래, 감성 분석, 상품 속성 추출, 상품 리뷰
E-commerce, Sentiment Analysis, Aspect Extraction, Product Reviews

1. 서론

웹은 사람들이 모여 다양한 주제에 대한 의견을 교환하는 장소가 되었다. 사용자들이 웹상에서 상품이나 서비스에 대해 표현한 의견을 담고 있는 평가 텍스트는 전자상거래 분야에서 특히 중요한데, 그 이유는 그러한 의견들이 다른 온라인 소비자들의 구매 결정에 직접적인 영향을 미치기 때문이다. 웹의 개방적인 특성 덕분에 소비자들이 온라인상에 남기는 '상품 리뷰'를 담고 있는 텍스트의 양이 급격하게 증가하였고, 자연스럽게 상품 리뷰에 담긴 방대한 양의 의견 정보를 자동으로 분석하는 방법이 필요해졌다. 감성 분석 (sentiment analysis)[3, 4, 7, 8, 13, 14, 16]은 이러한 필요에 맞춰 대두된 연구 분야로서, 텍스트 형태로 표현된 의견 정보를 자동으로 추출하고 분석하는 것을 목표로 한다. 주로 텍스트에 포함된 의견이 상품을 긍정적이거나 부정적으로 판단했는지 여부를 판단하는 것을 판단하게 된다.

감성 분석 연구의 초기에는 개별 '문서' 단위별 분석에서 시작하였으나, 최근에는 상품의 특성 (aspect) 단위로 분석하고자 하는 특성 기반 감성 분석 (aspect-based sentiment analysis) 연구가 많은 주목을 받기 시작했다. 특성 기반 감성 분석은 소비자가 대상 상품의 어떤 특성 (예: 화면, 외관)에 대해 긍정적이거나 부정적 의견을 표했는지를 분석하는 것을 목표로 한다. 예를 들어 "이 스마트폰의 화면은 밝지만, 배터리 수명이 너무 짧다"라는 문장에서, 두 가지의 다른 의견이 대상 스마트폰의 두 가지 특성에 대해 표현되었다. 화면(특성)은 밝다는 긍정적 의견과 배터리 수명(특성)은 짧다는 부정적 의견이 같이 존재하므로 특성 별 판단이 중요하다. 같은 상품이라도 각 특성 별로 표현된 평가는 다를 수 있기 때문에, 대상 상품의 각 특성 별로 어떤 의견이 표현되었는지를 인식하는 것은 매우 중요하다.

비록 현재로선 특성 기반 감성 분석 [4, 6, 7, 10, 15]이 가장 상세한 수준의 분석 방식이지만, 이 분석 방식에도 여전히 한계는 남아 있다. 대부분의 관련 연구들이 대상 측면에 대해 표현된 의견의 극성 (polarity)을 판별하는데 그쳤기 때문이다. 물론 이러한 형태의 분석이 유용한 경우도 있겠지만 좀 더 상세한 형태의 분석도 가능하다. 예를 들어 "이 엔진은 시끄럽다"라는 문장에서, 우리는 의견을 표현한 소비자가 엔진을 부정적으로 평가하고 있다는 것뿐만 아니라, 엔진을 소음이라는 '관점 (viewpoint)'에서 평가하고 있다는 것 역시 알 수 있다. 소비자들은 같은 상품이라도 다양한 관점에서 평가하기 마련이므로, 이러한 관점들을 파악하는 것은 소비자들이 왜 상품에 대해 긍정적이거나 부정적인 의견을 표현하는지를 이해하는데 도움을 줄 수 있다.

본 논문에서는 사람들이 상품을 평가하기 위해 표현한 의견에 암시된 관점을 추출하는 문제를 정의하고 그 방법을 제안한다. 본 논문에 등장하는 '대상 (target)'이라는 용어는, 상품 그 자체(예: '자동차')를 나타내거나 상품의 특정 aspect(예: '엔진')를 가리키기 위해

사용된다. 본 연구에서 제안하는 방식은 의견이 나타난 문장에 포함된 형용사로부터 관점 (예: 소음)을 추출하는 것인데, 이는 형용사가 의견을 표현하기 위해 가장 흔히 사용되는 품사이기 때문이다. 의견에 암묵적으로 표현되는 관점을 추출하게 되면 사람들이 상품의 어떤 측면에 관심을 기울이고 평가하였는지를 이해할 수 있게 되어 다양한 의견 텍스트에 존재하는 평가 정보를 요약하는데 도움이 된다 (그림 1).

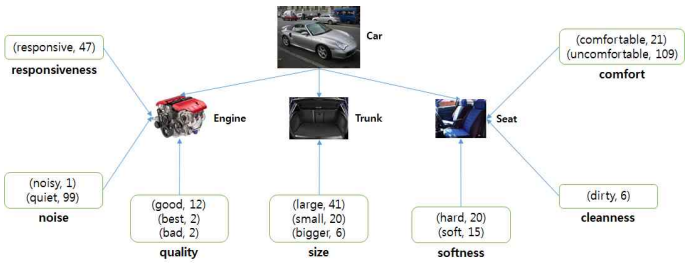


그림 1 관점(viewpoint) 기반 감성 분석

언어학분야에서 수행된 연구에 따르면, 형용사가 가진 유일한 역할은 그것이 수식하는 명사의 특정 속성(attribute)에 어떠한 값 (value)을 부여하는 것이다 [11]. 예를 들어 “이 카메라는 무겁다” 라는 문장에서, 형용사인 ‘무겁다’ 는 그것이 수식하는 명사인 카메라의 속성 중 하나인 ‘무게’ 에 값을 부여하고 있다. 본 연구에서는 이러한 속성이 사람들이 상품을 평가할 때 가졌던 관점과 일치한다는 가설을 세웠다. 즉 사람들이 ‘무겁다’ 라는 형용사를 사용하여 의견을 표현한 이유는, 그들이 카메라를 ‘무게’ 의 관점에서 평가하고자 하였기 때문인 것이다. 따라서 우리가 제안하는 태스크는 형용사가 가지고 있는 값을 사용해서 수식하고 있는 그 대상이 가지고 있는 속성을 추출하는 것이라고도 말할 수 있다.

형용사가 값을 부여하는 속성을 추출하기 위해, 본 연구에서는 거대한 영문 어휘 리소스인 워드넷 (WordNet) [12]을 활용하였다. 워드넷에서 모든 형용사 synset은 2,500개 이상의 클러스터로 구성되어 있으며, 각각의 클러스터들의 중심에는 head synset이라는 특수한 유형의 synset이 한 쌍 존재한다. 그 한 쌍의 head synset은 서로 반의어 관계를 가지고 있어 반대말 쌍(antonymous pair)이라고 불리며, 해당 클러스터에 속한 형용사들이 값을 부여하는 속성의 양 극값 (bipolar values)을 나타낸다. 예를 들어, ‘small-large’ 로 이루어진 반대말 쌍에서는 ‘size’ 라는 속성의 양 극값을 나타내며, 이 반대말 쌍을 중심으로 그와 유사한 의미를 가진 형용사 synset (satellite synset)들이 뭉쳐 하나의 클러스터를 이루고 있다. 따라서 같은 클러스터에 포함된 모든 형용사 synset들은 모두 양 극값 중 하나를 나타내고 있다. 따라서 형용사가 값을 부여하는 속성을 알아내기 위해서는, 먼저 해당 형용사가 어떤 클러스터에 속하는지를 알아내고, 그 클러스터를 대표하는 head synset을 인식한 다음, 마지막으로 그 head synset이 나타내는 속성을 추출하면 된다. head synset이 나타내는 속성은 워드넷 상에 코딩되어 있어 쉽게 알아낼 수 있다.

하지만 워드넷 상에서 모든 head synset이 자신과 연관된 속성을 가지고 있지는 않다. 워드넷 상에 있는 2,500개 이상의 형용사 클러스터 중 오직 620개의 head synset들만이 연관된 속성을 가지고 있다 (워드넷 3.0 기준). 이 커버리지 문제를 해결하기 위한 방법으로, 사용하려고 하는 head synset과 연관된 속성이 워드넷에 없는 경우 그와 유사한 다른 head synset에서 대신 연관된 속성을 추출하도록 하였다. 워드넷에서 유사한 head synset끼리는 ‘also see’ 라는 관계를 통해 서로 연결되어 있으므로 이 관계를 활용하면 된다. 또 다른 문제는 같은 형용사라 할지라도 그것이 나타내는 의미(sense 혹은 synset)가 여러 개 있을 수 있기 때문에 여러 개의 head synset과 연관되어 있을 수 있다는 것이다. 해당 형용사의 어떤 head synset을 선택하느냐에 따라 최종적으로 추출되는 속성(attribute)이 달라지므로 사용된 의미에 맞는 head synset을 선택하는 것을 매우 중요하다. 본 연구에서는 사람들이 상품에 대한 의견을 표현하기 위해 형용사들을 사용한 만큼 극성 점수(polarity score)가 가장 높은 의미로 형용사를 사용했을 것이라고 간주 하였고, 그러한 의미를 나타내는 head synset을 선택하는 것이 속성 추출의 정확도에 기여할 것이라고 가정하였다. 형용사의 각 의미가 가진 극성 점수를 계산하는 데는 SentiWordNet [1]을 활용하였다. SentiWordNet은 워드넷에 포함된 모든 synset들에 극성 점수를 부여한 영문 리소스이다.

본 연구에서는 위에서 설명한 방식으로 형용사로부터 추출한 ‘속성’ 이, 실제로 사람들이 상품에 대해 표현한 의견에 암시된 ‘관점’ 과 일치하는지를 검증하고자 하였다. 평가를 위해 110개의 테스트 케이스들이 생성되었고 각 테스트 케이스는 상품을 평가한 문장과, 그 문장에 포함된 형용사로부터 추출한 속성으로 구성되어 있다. 우리는 평가자들에게 생성한 테스트 케이스들을 보고 “추출된 속성이 의견 문장에 암시된 관점과 일치하는가?” 를 질문하였다. 실험 및 평가 결과는 본 논문에서 제안하는 관점 추출 태스크가 쉽지는 않지만 충분히 발전 가능성이 있다는 점을 보여주고 있다.

2. 관련 연구

본 연구는 특성 기반 감성 분석 [7, 10]과 관련되어 있다. 해당 태스크는 ‘몇 퍼센트의 사람들이 타겟을 긍정적이거나 부정적으로 평가했는지’ 와 같은 식으로 의견을 ‘정량적’ 으로 요약하는 것을 목표로 한다. 이런 수준의 분석도 물론 유용하지만, 만약 우리가 “사람들이 어떤 관점에서 타겟을 평가하였는지” 라는 질문에 대답할 수 있다면, 사람들의 의견에 대해 더욱 자세하게 이해할 수 있을 것이다. 우리가 아는 한 우리의 연구는 이러한 질문에 답하고자 한 최초의 연구이다.

암시된 특성 검출 (implicit aspect extraction) 연구 [4, 6, 15] 역시 우리의 연구와 관련되어 있다. 해당 태스크는 텍스트 속에서 명시적으로 나타나지 않았지만 의견 속에 암시된 특성을 추출하는 것을 목표로 한다. 예를 들어 우리는 “이 가방은 무겁다” 라는 의견 문장 속

에 '무게'라는 타겟이 암시되어 있다는 것을 알 수 있다. 본 논문에서 추출하고자 하는 관점 역시 문장에 명시적으로 나타나지 않은 암시적인 것이지만, 우리가 말하는 '관점'은 상품의 특성과는 다르다. 관점은 "사람들이 상품을 어떻게 평가하는가?"와 관련이 있는 개념이며, 그러므로 이는 일반적으로 말하는 '상품의 스펙'과는 다른 의미를 지닌다. 이런 측면에서 볼 때, 우리의 태스크는 사람들이 타겟을 어떤 관점에서 바라보고 있는지를 이해하는데 더 적합하다고 할 수 있다.

형용사들로부터 속성을 추출하기 위해, 우리는 [4]에서 사용된 것과 유사한 접근방식을 사용하였다. 이 연구는 우리의 연구와 마찬가지로 형용사로부터 그것이 값을 부여하는 속성을 추출하는 것을 목표로 하였는데, 워드넷이라는 단일 리소스만 이용해서는 커버리지가 충분치 않을 수 있다는 점에 주목하였다. 속성 추출 커버리지를 증가시키기 위해, 해당 연구의 저자들은 복수의 온라인 사전 리소스를 활용하는 방식을 제시한다. 그들이 제시한 방식은 크게 3개의 단계로 나눠진다:

1. 복수의 온라인 사전으로부터 형용사의 '주해(glosses)'를 가져온다.
2. 주해에 포함된 모든 명사를 추출하고, 추출된 명사들을 형용사의 속성 후보들로 간주한다.
3. 추출된 속성 후보들의 사전적 관계 (예: 동의어/반의어 관계)를 이용하여, 후보들이 실제로 속성을 나타내는지 아닌지를 자동으로 분류한다.

우리의 연구는 다음과 같은 점에서 그들의 것과 다르다: (1) 그들은 형용사로부터 암시된 상품의 특성을 추출하는 것을 목표로 하였다. 반면, 우리 연구의 목표는 사람들이 표현한 의견 속에 암시된 관점을 추출하고자 하였다. 의견 문장에 의견의 타겟이 명시적으로 나타났는지 아닌지 여부와는 상관이 없다. (2) 그들의 연구가 최대한 많은 형용사로부터 속성을 추출하는데 초점을 둔 반면, 우리 연구는 형용사가 값을 부여하는 속성이 의견에 암시된 관점과 일치하는지를 검증하고자 하였다.

3. 제안된 방법

이 단원에서는 먼저 워드넷에서 형용사들이 어떻게 구성되어 있는지에 대해 설명한다. 그 다음, 관점 추출에 필요한 단계를 설명한다. 마지막으로, 우리가 관점 추출의 커버리지와 정확도를 향상시키기 위해 취했던 조치들에 대해 설명한다.

3.1 워드넷의 형용사 구성

워드넷에는 descriptive, relational 이렇게 두 가지 유형의 형용사가 있다 [11]. 본 연구에서 우리는 descriptive 형용사에 초점을 맞추었는데, 이 유형의 형용사들이 우리가 원하는 형용사의 특성, 즉 그것이 수식하는 명사의 속성에 값을 부여하는 특성을 가지고 있기 때문이다. 예를 들어 "이 가방은 작다"라는 문장에서, '작다'라는 형용사는 가방의 '크기' 속성에 값을 부여하고 있다. 다음은 가방의 '크기' 속성에 '작다'라는 값을 부여하는 것을 함수 형태로 표현한 것이다.

크기(가방) = 작다

워드넷에서 형용사들은 '반의어 관계 (antonymy)'를 중심으로 조직되어 있는데, 이는 대부분의 경우 형용사가 값을 부여하는 속성은 양 극의 값 (bipolar values)을 갖기 때문이다 [11]. 예를 들어, '작다'의 반의어는 '크다'이고, '작다-크다'는 함께 '무게'(속성)의 양 극값을 나타낸다. 이러한 반의어 쌍을 워드넷에서는 antonymous pair라고 부르며, antonymous pair를 구성하고 있는 synset들을 head synset이라고 부른다. (그림 2)는 형용사 클러스터가 어떻게 구성되어 있는지를 보여준다. 워드넷에는 이와 같은 형용사 클러스터가 2,500여개 있고, 각 클러스터의 중심에는 앞서 설명한 대로 antonymous pair가 있다. 그 주위로 head synset과 유사한 의미를 가진 satellite synset들이 모여 하나의 클러스터를 구성하게 된다. 워드넷 상에서는 오직 head synset들만이 연관된 '속성'을 가지고 있으며, 따라서 특정 형용사로부터 속성을 추출하기 위해선 먼저 해당 형용사가 속한 클러스터의 head synset을 찾아야만 한다.

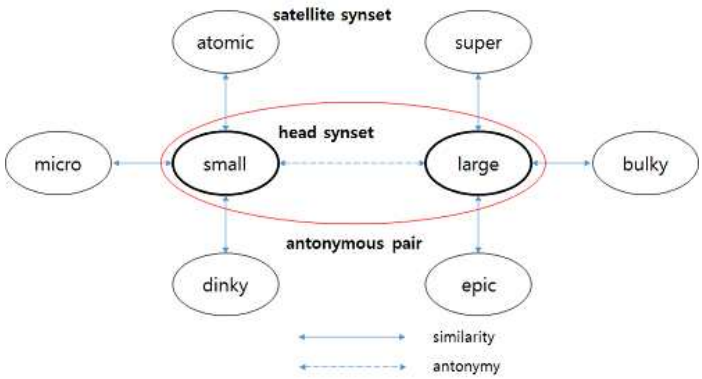


그림 2 형용사 클러스터

3.2 관점 추출 과정

형용사로부터 속성을 추출하기 위해서는 먼저 대상 형용사의 head synset을 알아내야 하는데, 그 때 발생하는 두 가지 이슈는 다음과 같다: (1) 찾아낸 head synset과 연관된 속성이 워드넷에 없을 때의 대처법, (2) 형용사가 여러 개의 head synset과 연관되어 있을 경우, 그들 중 하나를 선택하는 법. 본 단원에서는 형용사로부터 추출하는 과정과 각 이슈에 대처하는 법에 대해 자세히 설명한다.

3.2.1 전처리

먼저 형용사와 그것이 수식하는 명사를 판별하기 위한 품사 (POS) 태깅을 수행한다. 그리고 우리는 '의견 문장'을 '형용사와 그것이 수식하고 있는 명사를 포함하고 있는 문장'이라고 정의하고, 품사 태그로 구성된 {<NN><VB><JJ>} 패턴을 사용하여 그러한 문장들만을 골라 추출한다. 예를 들어, 앞에서 설명한 패턴을 이용하면 "engine is quiet"와 같은 문장이 추출된다. 문장에 포함된 형용사는 의견을 나타내며, 명사는 타겟을 나타낸다. 또한 우리는 텍스트에 포함된 노이즈를 줄이기 위해, 추출된 의견(형용사) 및 타겟(명사) 단어에 대해 스펠링 체크 및 lemmatization을 수행하였다.

3.2.2 형용사의 head synset 판별

형용사로부터 속성을 추출하기 위해선, 먼저 형용사의 head synset이 무엇인지를 알아내야 한다. 그러기 위해 우리는 대상 형용사를 lemma로 가지고 있는 head synset을 먼저 찾는다. Synset의 lemma는 그 synset이 나타내는 의미를 가지고 있는 구체적인 단어를 지칭하며, 따라서 어떤 형용사가 한 head synset의 lemma라는 것은, head synset이 나타내는 의미를 그 형용사가 가지고 있다는 것을 의미한다. 이 과정에서 워드넷을 사용하기 위해, 우리는 Natural Language Toolkit (NLTK) [2]에서 제공하는 워드넷 인터페이스를 활용하였다.

3.2.3 형용사의 head synset 선택

하나의 형용사는 여러 개의 sense를 가질 수 있으며, 그런 경우 여러 개의 head synset과 연관되게 된다. 본 논문에서 우리는 속성 추출에 사용할 head synset을 선택하는 두 가지 정책을 제시한다.

Top synset: 가장 자주 사용되는 head synset을 선택하는 정책이다. 워드넷에서 head synset은 사용 빈도순으로 정렬되어 있기 때문에, 가장 자주 사용되는 head synset을 쉽게 판별할 수 있다.

Polar synset: 가장 높은 극성 (polarity)를 가지고 있는 head synset을 선택하는 정책이다. 이 정책은 사람들이 상품을 평가하기 위해 형용사를 사용한 만큼, 높은 극성 (polarity)을 가진 의미로 형용사를 사용하였을 것이라는 가정에 기반하고 있다. head synset의 극성 점수를 계산하기 위해, 우리는 SentiWordNet [1]을 활용하였다. SentiWordNet는 워드넷의 모든 synset에 대한 객관성 점수 (objective score)를 계산해 놓은 리소스이다. 객관성 점수는 synset이 얼마나 객관적인 의미인지를 점수화해서 표현한 것으로, synset의 객관성 점수가 낮을수록 synset의 극성 점수가 높다.

3.2.4 속성 추출

바로 전 단계에서 선택된 하나의 head synset에서 연관된 속성을 추출한다. 워드넷에서 head synset은 자신이 연관된 속성과 'attribute'라는 관계로 연결되어 있다. 이 attribute 관계를 이용해서 head synset과 연관된 속성을 가져오면 된다.

3.2.5 유사한 head synset의 속성 추출

워드넷에 있는 모든 head synset이 그와 연관된 속성을 가지고 있지는 않다. 저자들이 파악한 바에 따르면 오직 620개의 head synset들만이 연관된 속성을 가지고 있다. 워드넷에 2,500개 이상의 형용사 클러스터가 있고, 각 클러스터가 반의어 관계에 있는 한 쌍의 head synset을 중심으로 가지고 있다는 점을 감안하면, 워드넷의 형용사 속성 커버리지는 낮은 편이라고 할 수 있다. 형용사의 속성 추출률을 높이기 위해, 우리는 워드넷에서 head synset 간에 존재하는 'also see' 관계를 활용하였다. also see 관계는 서로 다른 클러스터에 있지만, 관련된 의미를 가진 head synset들 사이를 연결하고 있다. 우리는 형용사의 head synset과 연관된 속성이 없는 경우, also see 관계를 이용해서 다른 유사한 head synset을 찾은 후, 그 head synset에서 다시 연관된 속성을 추출하도록 하였다. (그림 3)이 그 예시이다.

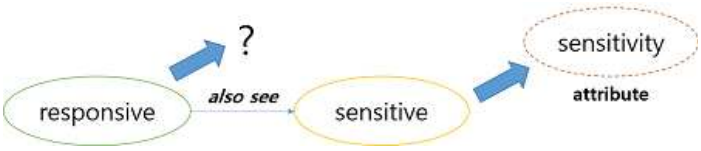


그림 3 유사한 head synset의 속성 추출

(그림 4)는 전체적인 속성 추출 과정을 그림으로 보여준다.

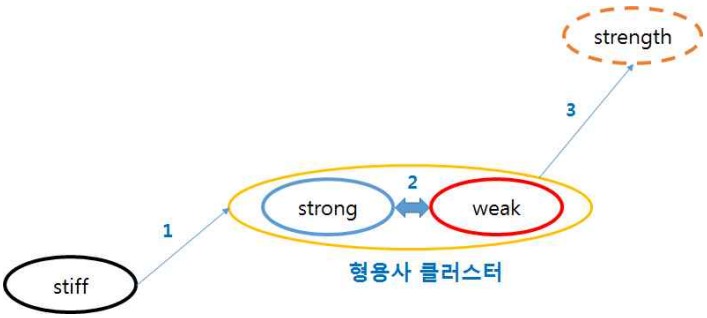


그림 4 속성 추출 과정

여기서 번호 1은 형용사('stiff')가 속한 클러스터를 알아내는 과정, 번호 2는 클러스터의 중심을 이루고 있는 한 쌍의 head synset인 antonymous pair ('strong-weak')를 찾는 과정, 마지막으로 번호 3은 head synset과 연관된 속성 ('strength')를 찾는 과정을 나타내고 있다.

4. 실험 결과

실험에는 Ganesan과 ChengXiang의 연구 [5]에서 사용된 데이터셋이 활용되었다. 정확하게는 전체 데이터셋 중 자동차 리뷰 데이터셋만이 활용되었는데, 데이터셋에는 edmund.com에서 수집한 2007년, 2008년, 2009년 식 자동차들에 대해 남겨진 리뷰들이 포함되어 있다. 각 년도 별로 140개에서 250개까지의 자동차가 있으며, 전체 자동차 리뷰의 개수는 42,230개이다 [5].

우리는 분석하고자 하는 타겟이 이미 주어져 있다고 가정하고 실험을 진행하였다. 우리는 11개의 타겟을 데이터셋에서의 출현 빈도수에 기반하여 선정하였다. 타겟에는 자동차 자체를 나타내는 car가 포함되어 있고, 자동차의 10가지 특성이 포함되어 있다. 선정된 11개 타겟의 리스트는 다음과 같다 (그림 5): *car, door, cabin, engine, navigation, seat, suspension, trunk, transmission, stereo, system.*

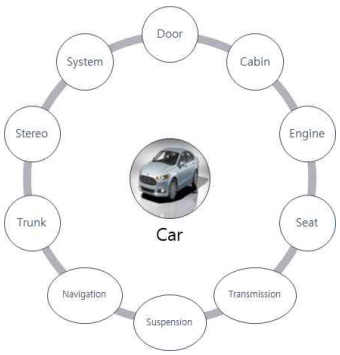


그림 5 타겟 리스트

분석할 타겟을 선정한 다음, 우리는 **3.2.1 전처리 단원**에서 설명한 패턴을 이용해서 데이터셋으로부터 타겟이 포함된 '의견 문장' 및 의견 문장에 포함된 형용사를 추출하였다. 예를 들어 대상 타겟이 engine이라면, 패턴을 이용하여 데이터셋으로부터 "engine is quiet" 라는 문장을 추출하고, 이 문장에 포함된 형용사인 'quiet' 역시 함께 추출한다. 그 다음, 우리는 각 타겟을 수식하는 형용사들을 출현 빈도순으로 정렬하였다. 이 단계를 거치는 이유는 각 타겟을 평가할 때 주로 사용되는 형용사가 무엇인지를 파악하기 위해서이다. 최종적으로 각 타겟 별 테스트 케이스를 생성할 때에는 빈도수 기준으로 top 10 형용사만을 선택하여 사용하였다.

형용사로부터 속성을 추출하기 위해, 우리는 3.2 단원에서 소개한 추출 방식들을 조합하여 총 네 가지의 알고리즘을 고안하였다:

1. *top_head: 가장 자주 사용되는 head synset으로부터 속성 추출*
2. *top_head_similar: 가장 자주 사용되는 head synset으로부터 속성 추출 & 실패할 경우 그와 유사한 head synset으로부터 속성 추출 시도*
3. *polar_head: 가장 높은 극성을 가진 head synset으로부터 속성 추출*
4. *polar_head_similar: 가장 높은 극성을 가진 head synset으로부터 속성 추출 & 실패할 경우 그와 유사한 head synset으로부터 속성 추출 시도*

위에서 제시한 네 가지 알고리즘의 성능을 비교하기 위해, 우리는 각 알고리즘별로 총 110개의 테스트 케이스로 이루어진 테스트 셋을 생성하였다. 각 테스트 케이스는 다음과 같은 세 개의 요소를 포함하고 있다:

- 의견 문장: **"engine is responsive"**
- 알고리즘이 추출한 속성: **sensitivity**
- 속성의 정의: **the ability to respond to physical stimuli**

(표 1)은 타겟 중 하나인 engine에 대해 생성된 테스트 케이스 일부를 보여주고 있다.

의견 문장	관점	관점 정의
engine is smooth	evenness.n.02	a quality of uniformity and lack of variation
engine is powerful	strength.n.01	the property of being physically or mentally strong
engine is great	quality.n.01	an essential and distinguishing attribute of something or someone

표 1 테스트 케이스 예시 ('engine')

평가를 위해, 우리는 세 명의 평가자에게 형용사로부터 추출된 속성이 의견 문장에 암시된 관점과 일치하는지를 평가해달라고 요청하였다. 측정 기준으로는 precision, recall, F1 점수와 더불어, 평가 agreement를 알아보기 위한 fleiss' kappa [9] 점수를 사용하였다. 실험 결과는 (표 2)에 나타나 있다.

알고리즘	precision	recall	F1	kappa (n=3)
top_head	0.6000	0.8182	0.6923	0.7033
top_head_similar	0.6182	0.8727	0.7237	0.6347
polarity_head	0.6455	0.8909	0.7486	0.5989
polarity_head_similar	0.6636	0.9182	0.7704	0.5927

표 2 실험 결과

실험 결과는 비록 우리가 제안한 태스크가 쉽지는 않지만, 발전 가능성이 있음을 보여주고 있다. 평가자들의 agreement 정도가 그리 높지 않다는 점에서 볼 때 이 태스크가 쉽지 않다는 것을 알 수 있다. 계산된 kappa 점수는 0.5에서 0.7사이로, 평가자들 사이의 agreement 수치가 보통에서 꽤 높음 사이라는 것을 의미한다. 사용한 방식 중 polarity_head 방식이 top_head 방식보다 높은 precision과 recall을 보였다는 것은 우리의 가설 - 사람들은 형용사를 극성이 높은 의미로 사용했을 것이다 - 이 합리적이었다는 것을 증명해준다. 더불어, 속성 추출에 실패할 경우 유사한 head synset으로부터 또다시 추출을 시도하는 정책은 precision 및 recall 향상에 기여했다는 것을 알 수 있다. recall 뿐만 아니라 precision이 함께 향상된 이유는, 만약 한 알고리즘이 속성 추출에 실패했을 경우, 다른 알고리즘과의 비교를 위해 그 알고리즘은 무조건 틀렸다고 간주되었기 때문이다.

5. 결론

본 논문에서는 의견에 암시되어 있는 상품 평가의 관점을 추출하는 태스크를 정의하고 하나의 방법론을 제안하였다. 의견 문장에 포함된 형용사로부터 추출된 속성이 사용자가 평가할 때 가졌던 관점과 일치한다는 가정하에, 워드넷을 활용하여 형용사로부터 속성을 추출하는 방법을 제안하였고, 평가자들을 통해 그 둘이 실제로 일치하는지를 평가하였다. 세 명의 평가자들로부터 받은 평가 결과는 이러한 접근 방식이 발전 가능성이 있지만, 여전히 개선할 여지가 많다는 것을 보여주었다. 보다 정확한 속성 추출을 위해선 단어의미 중의서 해소방법이 도입되어야 한다고 예상하고 있으며 이를 향후 연구로 수행 해나갈 예정이다.

참고문헌

[1] Baccianella, S. et al. 2010. Sentiwordnet 3.0: An enhanced lexical resource for sentiment analysis and opinion mining. Proceedings of the 7th conference on International Language Resources and Evaluation (LREC' 10) (2010), 2200-2204.

[2] Bird, Steven, E.L. and E.K. 2009. Natural Language Processing with Python. O' Reilly Media Inc.

[3] Dave, K. et al. 2003. Mining the peanut gallery: Opinion Extraction and Semantic Classification of Product Reviews. Proceedings of the twelfth international conference on World Wide Web (WWW' 03) (New York, New York, USA, May. 2003),

519.
[4] Fei, G. et al. 2012. A Dictionary-Based Approach to Identifying Aspects Implied by Adjectives for Opinion Mining. COLING (Posters)'12. 2, December 2012 (2012), 309-318.
[5] Ganesan, K. and Zhai, C. 2011. Opinion-based entity ranking. Information Retrieval. 15, 2 (Aug. 2011), 116-150.
[6] Hai, Z. et al. 2011. Implicit Feature Identification via Cooccurrence Association Rule Mining. Computational Linguistics and Intelligent Text Processing SE - 31. 6608, (2011), 393-404.
[7] Hu, M. and Liu, B. 2004. Mining and summarizing customer reviews. Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining (KDD'04) (New York, New York, USA, Aug. 2004), 168.
[8] Hu, M. and Liu, B. 2004. Mining opinion features in customer reviews. Proceedings of the 19th national conference on Artifical intelligence (Jul. 2004), 755-760.
[9] Joseph, F.L. 1971. Measuring nominal scale agreement among many raters. Psychological bulletin. 76, 5 (1971), 378-382.
[10] Liu, B. et al. 2005. Opinion Observer : Analyzing and Comparing Opinions on the Web. Proceedings of the 14th international conference on World Wide Web (WWW'05) (New York, New York, USA, May. 2005), 342.
[11] Miller, G.A. et al. 1993. Introduction to WordNet : An Online Lexical Database. International Journal of Lexicography . 3 , August (Dec. 1993), 235-244.
[12] Miller, G.A. 1995. WordNet: A Lexical Database for English. Communications of the ACM. 38, 11 (1995), 39-41
[13] Moghaddam, S. and Ester, M. 2010. Opinion digger. Proceedings of the 19th ACM international conference on Information and knowledge management (CIKM'10) (New York, New York, USA, Oct. 2010), 1825.
[14] Pang, B. et al. 1988. Thumbs up? Sentiment Classification sing Machine Learning Techniques. Proceedings of the ACL-02 conference on Empirical methods in natural language processing-Volume 10. (1988), 79-86.
[15] Su, Q. et al. 2008. Hidden Sentiment Association in Chinese Web Opinion Mining. Proceedings of the 17th international conference on World Wide Web (WWW'08) (New York, NY, USA, 2008), 959-968.
[16] Zhuang, L. et al. 2006. Movie Review Mining and Summarization. Proceedings of the 15th ACM international conference on Information and knowledge management (CIKM'06)(New York, New York, USA, 2006), 43.

읽기 매체의 다양성과 흥미도를 고려한 가독성 측정

김아영○, 박성배, 이상조
경북대학교, 컴퓨터학부
{aykim, sbpark}@sejong.knu.ac.kr sjlee@knu.ac.kr

Revisiting Readability: An aspect of the reading environments and perspectives

A-Yeong Kim○, Seong-Bae Park, Sang-Jo Lee
Kyungpook National University, Department of Computer Science and Engineering

요 약

가독성이란 글을 읽음에 있어 얼마나 쉽게 쓰여졌는가, 얼마나 흥미로운가의 정도를 나타낸다. 기존 연구에서는 가독성을 측정함에 있어 인쇄물에 한정하고 글의 구성 관점에서만을 반영하였다. 하지만, 글은 인쇄물뿐만 아니라 모바일, 태블릿 등 다양한 읽기 매체를 통해 읽어질 수 있으며 글의 흥미에 따라 가독성이 달라지므로 기존의 연구 방법으로는 적절히 반영할 수 없다. 본 논문에서는 읽기 매체에 따라 가독성이 어떻게 달라지는지, 가독성을 측정함에 있어 관점들이 어떻게 반영되는지를 살펴보고자 한다. 서로 다른 읽기 매체들을 사용하여 동일한 뉴스 기사에 대해 가독성을 측정하였으며, 측정된 결과를 사용하여 각 읽기 매체가 가독성를 측정하는 요소들 중 어떤 요소에 큰 영향을 받는지를 살펴본다. 또한, 가독성을 측정함에 있어 글의 구성뿐만 아니라 흥미 관점을 추가하였으며, 이로부터 가독성의 요소들이 글의 구성 관점과 흥미 관점에서 어떠한 차이점을 보이는지 살펴본다.

주제어: 가독성, 읽기 매체, 가독성 관점

1. 서론

가독성(Readability)이란 글을 읽음에 있어 얼마나 쉽게 쓰여 있는가(well-written)[1], 얼마나 흥미로운가(interesting)를 의미한다[2]. 가독성은 글을 읽음에 있어 난이도를 도입하여 학생들의 수준에 맞는 글을 가르치고자 측정하게 되었다[3]. 일반적으로 가독성은 가독성을 결정하는 각 요소(factor)들에 의해 측정된다[4,5,6,7]. 자연언어처리 관점에서 가독성을 결정하는 요소로는 글의 길이, 명사구 또는 동사구의 비율, 대명사의 비율 등이 존재한다[4]. 최근 들어 교육적인 목적뿐만 아니라 다양한 매체에서 가독성을 구하거나 가독성에 영향을 주는 새로운 요소를 제안하는 연구들이 활발히 진행되고 있다[4,5,8].

기존의 자연언어처리 측면에서의 가독성 측정 연구들은 주로 인쇄물에서 집중되어 왔다[4,5,6,7]. 하지만, 최근 들어 읽기 매체가 다양해짐에 따라 사용자는 인쇄물뿐만 아니라 스마트폰, 태블릿과 같은 휴대용 기기를 사용하여 글을 읽을 수 있게 되었다. 각 읽기 매체들은 매체들 마다 특성이 존재하므로 동일한 글이라도 가독성이 달라질 수 있다. 또한 기존의 방법들은 글의 구성(well-written) 관점만을 고려하여 가독성을 측정하였다[2,4,5]. 하지만 가독성은 글의 구성뿐만 아니라 흥미도에도 영향을 받을 수 있기 때문에 가독성을 측정함에 있어 각 관점을 고려할 필요가 있다.

본 논문에서는 읽기 매체의 다양성 및 가독성의 다양한 관점을 고려하여 가독성을 측정하고 다음의 이슈들을 살펴보고자 한다. 1) 가독성을 결정하는 요소들은 인쇄물과 휴대용 기기에서 각각 다른 중요도로 적용 될 것인가? 2) 가독성을 측정하는 관점에 따라 가독성 측정에 영향을 미치는 요소들이 달라지는가?

본 논문에서는 뉴스 기사를 이용하여 인쇄물, 스마트폰, 태블릿에서 글의 구성, 글의 흥미를 반영하여 가독성을 측정한다. 같은 글에 대해 인쇄물과 휴대용 기기에서, 가독성을 결정하는 요소들과 가독성이 어떠한 상관관계를 가지는지 살펴본다. 또한 글의 구성을 나타내는 관점과 흥미도의 관점에서, 가독성의 요소와 가독성이 어떠한 상관관계를 가지는지를 관점마다 비교하며, 관점에 따라 서로 다른 특징을 살펴본다.

본 실험은 60개의 뉴스 기사에 대해 4명의 실험 대상자로부터 가독성을 측정하였다. 실험 결과, 첫 번째로 각 읽기 매체에 대해 상관관계가 유의미하게 보인 요소들이 서로 다름을 보이며, 이를 통해 읽는 매체마다 가독성의 차이가 존재함을 보였다. 다음으로 관점에 따라 영향을 주는 요소들의 차이가 존재하였다. 글의 구성 측면에서는 읽은 매체들이 서로 다른 요소들에 의해 영향을 받는 반면 글의 흥미 측면에서는 읽은 매체에 관계없이 요소들이 서로 비슷한 경향을 보였다. 이를 통해 글의 구성적인 측면에서는 매체마다 차이점을, 글의 흥미 측면에서는 매체가 무관한 경향을 보여주었다.

본 논문의 구성은 다음과 같다. 2장에서는 가독성 측정과 관련된 기존 연구들을 살펴본다. 3장에서는 가독성 측정 방법에 대하여 간략히 다루고, 본 논문에서 사용한 가독성을 결정하는 요소에 대해서 설명한다. 4장에서는 실험 및 결과를 분석하고, 마지막 5장에서는 결론을 다룬다.

2. 관련 연구

가독성을 측정하고자 하는 연구는 오래전부터 많이 진행되어왔다. 현재까지 가장 보편하게 널리 쓰이고 있는 측정 방법들은 주로 단어의 길이나 문장의 길이 또는 쉬운 단어의 빈도수를 바탕으로 측정하였다[6,7]. Flesch는 가독성을 "읽기의 쉬움" 이라 표현하고 말뭉치로부터 무작위로 100개의 단어를 선정, 단어의 평균 길이와 문장의 길이를 이용해 가독성 측정 공식을 도출하였다[6]. 이후 첫 번째 식을 보정하여 말뭉치의 특성을 더 잘 반영하도록 가독성 측정 공식을 확장하였다[7]. 이를 Flesch-Kincaid 식이라고 하며 현재까지도 교과서나 교육을 목적으로 한 책의 난이도를 측정하는데 쓰이고 있다. Dale은 3천개의 쉬운 단어 리스트를 미리 정의해두고, 단어가 포함된 문장의 평균 길이와 그 목록에 해당하는 단어의 백분율을 이용하였다[2]. 위 연구들은 단어의 길이나 문자의 길이 등 글의 표면적인 요소들만을 이용하여 가독성을 측정하였다. 하지만, 구문적인 요소를 반영하지 않기 때문에 정확한 가독성 측정을 할 수 없다는 한계를 보인다[5].

앞선 연구들의 단점을 보완하기 위하여 가독성을 측정하는 새로운 요소들을 추가하고 가독성의 측정 성능을 높이기 위한 요소들의 조합을 찾아내는 연구들이 진행되어 왔다[4,5]. Thomas et al.은 프랑스어를 외국어로 쓰는 사람들에게 맞는 새로운 가독성 측정 공식과 함께 기존 요소를 포함하여 46개의 요소들을 제안하였다[5]. 크게 어휘, 구문, 의미 그리고 프랑스어의 특성을 반영하는 요소와 같이 크게 4개로 분류하여 정의하였으며, 이를 바탕으로 가독성을 측정하였다. Pitler et al.은 글의 어휘, 구문, 담화를 나타내는 요소와 가독성의 관계를 살펴보았다[4]. 제안한 방법은 실험자가 글을 읽어 글에 대한 가독성 점수를 매긴 다음 그 점수와 가독성을 결정할 수 있는 요소간의 상관계수를 구했다. 상관계수를 이용하여 그 요소가 가독성을 판단하는데 얼마나 주요하게 영향을 미치는지 분석하였다. 그 결과, 문장 당 동사구의 평균 개수 및 글의 길이, 담화와 관련된 요소들이 가장 중요하게 영향을 끼쳤음을 보였다. 하지만 위의 방법들은 인쇄물 이외의 다른 읽기 매체들을 고려하지 않았으며, 글을 읽는 관점 또한 글의 짜임 및 구성(well-written)만을 고려하여 가독성을 측정하였다는 단점을 지닌다.

3. 가독성 측정

일반적으로 글 d가 주어졌을 때 가독성 R은 가독성을 결정할 수 있는 요소 x_i와 그 요소들의 중요도 α_i를 사용하여 계산될 수 있다[2,4,5,6,7]. 각 요소들이 가독성에 독립적으로 영향을 끼친다고 가정을 하면 가독성은 수식 (1)과 같이 선형 모델로 표현할 수 있다.

$$R(d) = \sum_{i}^{i} \alpha_i x_i \qquad (1)$$

각 요소의 가중치는 측정된 가독성 점수와의 선형 회귀(linear regression)를 통해 추정할 수 있다.

본 논문에서는 가독성을 결정하는 요소로 [4]에서 제안한 요소들 중 한국어에 적용할 수 있는 8가지를 선택하여 사용한다. 각 요소는 표면(Superficial), 어휘(Lexical), 구문(Syntactic), 문장의 연결성(Sentence Continuous)과 같이 4가지로 분류한다. 각 요소에 대한 설명은 다음과 같다.

표면적인 요소

표면적인 요소(Superficial factor)는 글의 구성적인 측면을 반영하고 있는 요소들이다[4,5]. 본 논문에서는 글의 길이(TL), 평균 문장 길이(SL), 문장 당 평균 어절 수(WS), 어절 당 평균 어절 길이(CW)를 사용하였다. 글의 길이는 글의 양을 측정하며 글이 지나치게 길게 되면 글의 흐름이 쉽게 끊길 수 있다. 글의 길이와 문장 당 평균 단어의 수는 수식어구의 영향을 받는다. 수식어구가 많으면 글의 길이와 문장 당 평균 단어 수도 같이 길어지거나 증가한다. 반대로 수식어구의 수가 적으면 글의 길이와 문장 당 평균 단어수가 같이 짧아지거나 줄어든다. 어절 당 평균 어절 길이는 전문용어와 같은 복합명사들을 반영할 수 있기 때문에 가독성에 영향을 주게 된다.

어휘적인 요소

어휘적인 요소(Lexical factor)는 글이 말뭉치로부터 자주 나오는 단어들로 구성되어있는지를 반영한다. 본 논문에서는 유니그램(unigram) 모델로부터 글에 쓰인 단어에 대한 확률을 구하였으며 식은 아래와 같다[4].

$$\prod_{w} P(w|M)^{C(w)}$$

여기서, $P(w|M)$은 말뭉치 M에 대한 단어 w가 나올 확률이고, $C(w)$는 글에서 단어 w가 나타나는 횟수이다. 위 수식에 대한 로그우도(log likelihood, LL)식은 다음과 같이 정의할 수 있다.

$$\sum_{w} C(w)\log(P(w|M))$$

글의 로그 우도의 값이 높으면 그 글은 주어진 말뭉치에서 주로 사용하는 어휘를 사용하고 있다는 것을 의미한다.

구문적인 요소

구문적인 요소(Syntactic factor)는 글의 구문적인 측면을 반영한다[4]. 문장에 구문적인 요소가 많을수록 문장을 복잡하게 만든다는 특성을 가지고 있다. 본 논문에서는 구문적인 요소로 문장 당 평균 명사구 수(NP), 문장 당 평균 동사구 수(VP)를 사용하였다. 명사구와 동사구는 많을수록 문장이 복잡해지지만 이해와 흥미를 높여주기 때문에 글의 흥미도를 반영할 수 있는 요소이다[9].

표 1 뉴스 기사에 대한 간단한 통계 자료

구분	최소값	최대값	평균값
글의 길이	107	546	326.23
문장 수	1	13	5.63
어절 수	3	47	17.48

문장의 연결성

문장의 연결성(Sentence Continuous)이란 주어진 문장들이 의미적으로 얼마나 잘 연결 되어있는지를 나타낸다. 본 논문에서는 문장의 연결성을 나타내는 요소로 문장 당 평균 대명사의 수(PRP)만을 사용하였다[4]. 대명사는 그 수가 많을수록 문장의 복잡도가 증가하지만 글의 흐름을 매끄럽게 하기 때문에 글의 구성적인 측면을 반영한다.

4. 실험

4.1 실험 데이터 및 가독성 채점 기준

실험을 위해 본 논문에서는 Naver에서 뉴스를 수집하였다. 6월 10일부터 6월 25일까지의 연예, 스포츠, 정치 3개의 카테고리에서 기사 60개를 수집하였다. 표 1은 실험에 사용된 기사 60개에 대한 통계적 정보를 보여준다. 가장 길이가 짧은 기사는 107자이며, 가장 긴 기사는 546자로 평균적으로 326.23자의 기사로 이루어져있다. 기사 한 개에 최소 1문장부터 최대 13문장으로 이루어져 있으며 평균적으로 5.63개의 문장으로 이루어져 있다. 한 문장은 최소 3개의 어절에서 최대 47개의 어절로 이루어져 있고 평균적으로 17.48개의 어절을 가지고 있다.

가독성 측정을 위해 학부생 2명에게 스마트폰(5인치), 태블릿(10.1인치), 인쇄물(A4용지) 총 세 가지 읽기 매체를 사용하여 60개의 기사를 읽게 하였다. 각 읽기 매체의 폰트 크기는 동일하게 하였다. 채점 기준은 두 가지로 다음과 같다.

▶ 주어진 기사는 구성이 좋다.

글이 얼마나 읽기 좋게 쓰여 있는지를 평가한다. 기존 연구에서 사용했던 주요 채점 방법이다. 본 논문에서는 주로 글의 짜임새나 흐름이 잘 흘러가는지를 중심으로 채점하였다.

표 2 채점 점수에 대한 통계 자료

구분	항목	최저 점수	최고 점수	평균 점수	분산
읽기 매체	스마트폰	2.5	5	3.767	0.432
	태블릿	2.5	5	3.867	0.295
	인쇄물	2	4.75	3.867	0.490
관점	글의구성	2	5	3.908	0.420
	글의흥미	1	5	3.550	0.735

▶ 주어진 기사는 흥미롭다.

주어진 기사의 흥미로움을 채점한다. 기존 연구에서는 가독성 측정에서의 글의 흥미도에 대한 언급은 하였지만 반영을 하지 않았다[4]. 하지만 가독성은 글의 구성뿐만 아니라 글의 흥미도와 함께 복합적으로 영향을 받기 때문에 본 논문에서는 글의 흥미도 또한 채점 기준으로 고려하였다.

점수는 5점(매우 좋다, 매우 흥미롭다)에서 1점(전혀 좋지 않다, 전혀 흥미롭지 못하다)까지 다섯 개의 등급으로 각 기준에 대하여 채점하였다. 최종적으로 각 기사마다 두 개의 기준에 대한 점수에 평균을 하여 가독성을 매겼다. 표 2는 채점 점수에 대한 통계 수치를 나타낸다. 글의 흥미를 제외한 나머지 항목은 최저 점수가 2점대였고, 인쇄물을 제외한 나머지 항목은 최고 점수가 5점 이었다. 평균 점수는 모든 항목들이 3점대 후반이었다. 카파 계수(Kappa value)는 0.626이다.

어휘적인 요소를 반영하는 로그우도를 계산하기 위해 말뭉치로 2013년도 1월 1일부터 2013년 9월 6일까지 Naver 기사를 수집하여 사용하였다. 수집된 전체 기사의 개수는 298,729개이다.

4.2 실험 결과

4.2.1 가독성 결정 요소와 읽기 매체와의 관계

표 3은 가독성 결정 요소와 가독성과의 상관관계를 읽기 매체별로 나타낸 것이다. 가독성 측정을 위해 사용한 8개의 요소에서 유의미한 값 중 상관관계가 높은 순서대로 나타내었다. 먼저 스마트폰의 경우 화면 크기의 제약 때문에 글의 길이(TL)가 크게 영향을 받음을 볼 수 있다. 이와 반대로 태블릿과 인쇄물은 화면의 제약이 없어

표 3 읽기 매체별 가독성 결정 요소와 가독성의 관계

스마트폰		태블릿		인쇄물	
요소	값	요소	값	요소	값
글의 길이(TL)	**-0.257**	**문장 당 평균 명사구 수(NP)**	**0.335**	**문장 당 평균 단어 수(WS)**	**0.315**
문장 당 평균 대명사 수(PRP)	**0.104**	**문장 당 평균 단어 수(WS)**	**0.276**	**문장 길이(SL)**	**0.294**
로그우도(LL)	-0.100	문장 길이(SL)	0.234	**문장 당 평균 명사구 수(NP)**	**0.264**
어절 당 어절 길이(CW)	-0.096	문장 당 평균 동사구 수(VP)	0.212	문장 당 평균 동사구 수(VP)	0.158

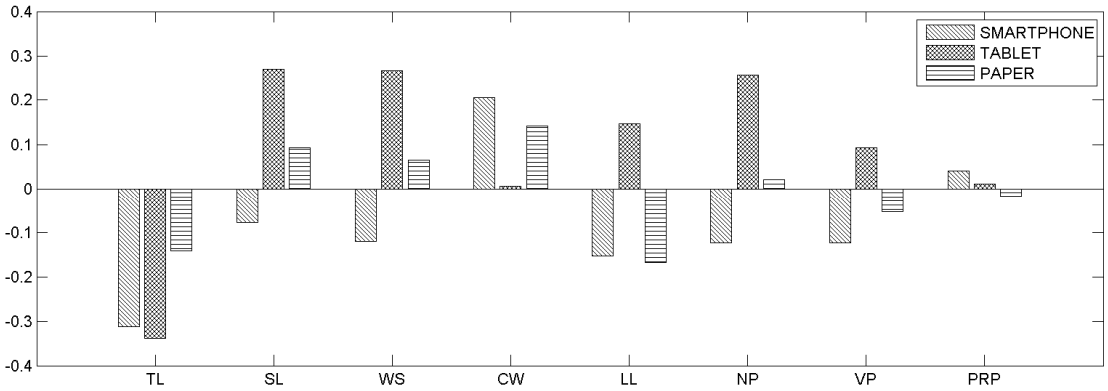


그림 1 글의 구성 관점에 대한 가독성과 요소간의 상관관계

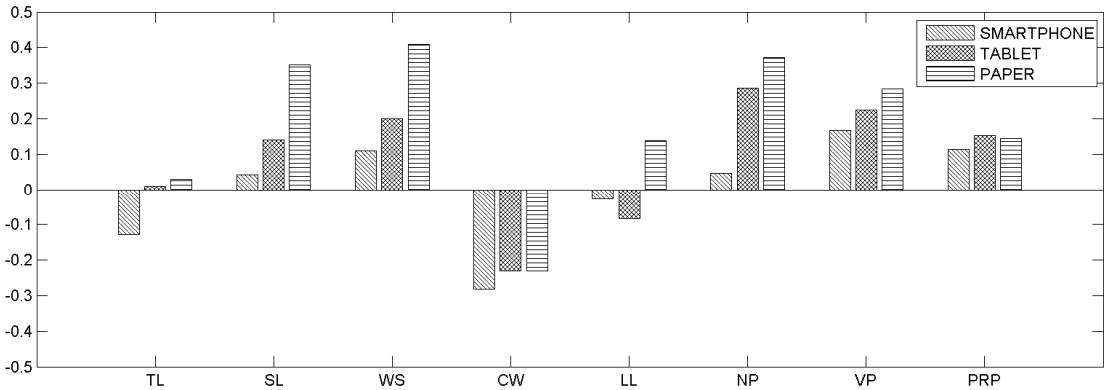


그림 2 글의 흥미 관점에 대한 가독성과 요소간의 상관관계

글의 내용과 관련된 요소들(NP, VP)에 영향을 받음을 볼 수 있다. 스마트폰에서의 대명사의 경우 수가 많을수록 문장의 복잡도가 높아지지만, 글의 흐름을 이어주는 역할을 하므로 가독성에 좋은 영향을 주는 것으로 분석이 된다. 태블릿과 인쇄물의 경우 명사구 요소(NP)가 큰 영향을 주었는데 이는 가독성이 구성과 흥미 점수를 평균화한 것이기 때문에 나타난 현상으로 간주할 수 있다. 일반적으로 명사구는 많을수록 독자에게 흥미를 줄 수 있기 때문에 양의 상관성을 크게 보이게 된다. 문장의 길이와 문장 당 단어 수가 양의 상관성을 크게 보였는데 이는 문장 내에서의 설명 등을 자세히 기술함을 의미하여 가독성을 좋게 하는 것으로 보인다. 위 결과를 통해 가독성을 결정하는 요소들이 읽기 매체마다 다른 상관관계를 가지는 것을 볼 수 있다.

4.2.2 가독성 결정 요소와 글의 구성 관점과의 관계

그림 1은 가독성 결정 요소와 가독성과의 상관관계를 글의 구성 관점 기준으로 나타낸 것이다. 읽기 매체와 상관없이 글의 길이가 길수록 글의 구성에 대한 가독성은 떨어짐을 알 수 있다. 스마트폰에서는 글의 구성 측면에서 글의 길이가 가장 큰 영향을 미쳤다. 길이 요소가 크게 영향을 주어 나머지 요소들에 대해서는 0.2 미만의 값으로 약한 상관관계를 보이게 된다. 태블릿의 경우 스마트폰과 동일하게 글의 길이가 가장 크게 영향을 주었다. 하지만 스마트폰과의 경우와는 조금 다르다. 문장의 길이, 문장 당 단어 수, 문장 당 평균 명사구 수 또한 0.2 이상의 값으로 영향을 주고 있다. 인쇄물의 경우 전체적으로 영향을 주고 있다. 위 결과는 이전 논문과 비슷한 결과를 보이고 있다[4]. 글의 구성 관점에서 글의 길이를 제외한 나머지 요소들은 읽기 매체마다 다른 특성을 보인다.

4.2.3 가독성 결정 요소와 글의 흥미 관점과의 관계

그림 2는 가독성 결정 요소와 가독성과의 상관관계를 글의 흥미 관점 기준으로 나타낸 것이다. 흥미도의 경우 읽기 매체와 상관없이 비슷한 양상의 상관관계를 보임을 알 수 있다. 어절 당 평균 어절 수의 요소가 모든 읽기 매체에 대해서 음의 상관관계를 보였다. 이는 어절 당 평균 어절 길이가 전문용어와 같은 복합명사를 나타내므로 흥미 측면에서 가독성을 떨어트릴 수 있기 때문이다. 기존 연구와 같이 문장 당 평균 명사구 및 동사구의 수에 대한 요소의 경우 명사구와 동사구의 개수가 많을수록 글의 흥미를 상승시켜주는 효과를 가져왔다. 이러한 효과는 인쇄물에서 문장의 길이와 문장 당 평균 단어 수도 같이 반영이 되어 보임을 알 수 있다.

전체적으로 스마트폰은 세 가지의 결과에 대해 글의 길이에 대해서는 공통적으로 모두 음의 상관관계를 보여 기기의 특성을 반영함을 알 수 있다. 글의 구성 측면에

서는 인쇄물은 요소들이 고루 영향이 끼침을 보이는 반면 태블릿은 문장 길이, 문장 당 단어 수, 문장 당 평균 명사구 수에 대해서 큰 영향을 받음을 보였다. 글의 흥미도 측면에서는 태블릿의 경우 인쇄물의 크기와 흡사하여 전체적으로 비슷한 양상을 보이지만 스마트폰의 경우 글의 길이나 로그 우도에서 상관관계의 부호가 달라지는 등의 차이를 보였다.

5. 결론

본 논문에서는 뉴스 기사를 이용하여 다양한 읽기 매체(인쇄물, 태블릿, 스마트폰)에서 글의 구성, 흥미를 반영한 가독성을 측정하였다. 측정된 가독성을 통해 본 논문에서는 읽기 매체에 따라 같은 글에 대해서 가독성을 결정하는 요소와 가독성이 차이가 있는지를 살펴보았다. 게다가, 글의 구성과 흥미도의 관점에서 중요하게 다뤄지는 가독성의 요소가 읽기 매체와 글의 관점마다 차이가 있는지를 살펴보았다.

실험 결과, 같은 글에 대해 읽기 매체에 따라 가독성이 달라짐을 보였다. 특히, 화면이 작은 스마트폰에서는 글의 길이와 같이 표면적인 요소에 영향을 많이 받는 것을 볼 수 있었으며, 상대적으로 화면이 큰 태블릿, 인쇄물은 구문적인 요소에 영향을 많이 받는 것을 볼 수 있었다. 또한 가독성과 글의 구성 관점에서는 기기에 따라 차이가 있음을 보였다. 글의 길이는 공통적으로 영향을 많이 받았으나, 다른 요소들은 기기마다 다르게 영향을 받음을 볼 수 있었다. 또한 가독성과 글의 흥미 관점에 대해서는 글의 구성 관점과는 달리 기기에 상관없이 일정한 관계를 가짐을 볼 수 있었다.

위의 실험 결과를 바탕으로 가독성이란 읽기 매체와 읽는 관점에 따라 다르게 반영이 되어야 함을 알 수 있다. 이에 향후 연구로는 다르게 반영된 가독성이 사용자들이 실제로 글을 읽음에 얼마나 도움이 되는지를 분석할 예정이다.

감사의 글

본 논문은 지식경제부 산업원천기술개발사업(10035348, 모바일 플랫폼 기반 계획 및 학습 인지 모델 프레임워크 기술 개발)의 지원으로 수행되었음

참고문헌

[1] J. C. Richards, J. Platt, and H. Platt, Longman dictionary of language teaching and applied linguistics, 1992.

[2] E. Dale and J. S. Chall, "The concept of readability", *Elementary English,* Vol. 26, No. 23. 1949.

[3] T. Virtanen-Ulfhielm "Linguistic Complexity in Two Major American Newspapers and The Associated Press Newswire, 1900-2000," *Master's thesis,* Abo Akademi, 2004.

[4] E. Pitler and A. Nenkova, "Revisiting Readability: A Unified Framework for Predicting Text Quality," In *Proceedings of EMNLP*, pp 186-195, 2008

[5] T. Francois and C. Fairon, "An "AI readability" formula for French as a foreign language, ", In *Proceedings of EMNLP and CoNLL*, pp. 466-477, 2012.

[6] R.Flesch, "A new readability yardstick" *Journal of Applied Psychology*, vol.32, pp.221-233, 1948.

[7] J. P. Kincaid, R. P. Fishburne, R. L. Robers, B. S. Chissom, "Derivation of New Readability Formulas(Automated Reliability Index, Fog Count and Flesch Reading Ease Formula) for Navy Enlisted Personnel," *Research Branch Report* pp.8-75, 1975.

[8] X. Yan, D. Song, and X. Li, "Conecpt-based Document Readability in Domain Specific Information Retrieval," In *Proceedings of CIKM,* pp. 540-549, 2006.

[9] R. Barzilay and M. Lapata, "Modeling Local Coherence: An Entity-based Approach," In *Proceedings of ACL*, page 141-148, 2005.

토픽 모델을 이용한 모바일 앱 설명 노이즈 제거

윤희근O, 김솔, 박성배
경북대학교
{hkyoon,skim,sbpark}@sejong.knu.ac.kr

Noise Elimination in Mobile App Descriptions Based on Topic Model

Hee-Geun YoonO, Sol Kim, Seong-Bae Park
Kyungpook National University

요 약

스마트폰의 대중화로 인하여 앱 마켓 시장이 급속도로 성장하였다. 이로 인하여 하루에도 수십개의 새로운 앱들이 출시되고 있다. 이러한 앱 마켓 시장의 급격한 성장으로 인해 사용자들은 자신이 흥미를 가질만한 앱들을 선택하는데 큰 어려움을 겪고 있어 앱 추천 방법에 대한 연구에 많은 관심이 집중되고 있다. 기존 연구에서 협력 필터링 기반의 추천 방법들을 제안하였으나 이는 콜드 스타트 문제를 지니고 있다. 이와는 달리 컨텐츠 기반 필터링 방식은 콜드 스타트 문제를 효율적으로 해소할 수 있는 방법이지만 앱 설명에는 광고, 공지사항등 실질적으로 앱의 특징과는 무관한 노이즈들이 다수 존재하고 이들은 앱 사이의 유사관계를 파악하는데 방해가 된다. 본 논문에서는 이런 문제를 해결하기 위하여 앱 설명에서 노이즈에 해당하는 설명들을 자동으로 제거할 수 있는 모델을 제안한다. 제안하는 모델은 모바일 앱 설명을 구성하고 있는 각 문단을 LDA로 학습된 토픽들의 비율로 나타내고 이들을 분류문제에서 우수한 성능을 보이는 SVM을 이용하여 분류한다. 실험 결과에 따르면 본 논문에서 제안한 방법은 기존에 문서 분류에 많이 사용되는 Bag-of-Word 표현법에 기반한 문서 표현 방식보다 더 나은 분류 성능을 보였다.

주제어: 모바일 앱 추천, 노이즈 필터링, LDA

1. 서론

스마트폰의 대중적인 보급으로 인하여 수많은 모바일 앱이 출시되고 있다. 대표적인 스마트폰 앱 마켓인 애플의 앱스토어와 구글 플레이 스토어에는 현재 100만개 이상의 앱이 등록되어 있으며, 매일 수십여 개의 새로운 앱들이 등록되고 있다. 이러한 앱 마켓의 급진적인 성장은 사용자로 하여금 자신이 흥미를 가질만한 새로운 앱을 선택하는데 큰 어려움을 겪게 만들었다. 이로 인해 최근에는 사용자가 관심을 가질만한 앱을 추천해주는 서비스에 대한 연구가 증가하고 있다.

기존의 추천은 크게 협력 필터링(Collaborative filtering)과 컨텐츠 기반 필터링(Content-based filtering), 2가지 유형으로 구분된다. 협력 필터링 방법은 사용자들 사이의 상관관계에 기반하여 비슷한 취향의 사용자들의 정보에 기반하여 추천을 수행한다. 이 방법은 사용자들이 아이템에 대하여 평가한 이력이 충분하게 존재한다면 우수한 추천 성능을 보여주며, Amazon, CDnow 등 다양한 상업 사이트에 적용되었다. 하지만 평가 이력이 존재하지 않는 콜드 스타트 문제에 대해서는 협력 필터링 방법은 매우 취약하다. 특히 새롭게 출시되는 모바일 앱의 경우는 사용자들의 사용 및 평가 이력이 존재하지 않아 심각한 콜드 스타트 문제를 안고 있다. 그렇기 때문에 협력 필터링 방법에 기반한 모바일 앱 추천은 큰 한계를 지니고 있다.

컨텐츠 기반 필터링 방법은 사용자가 관심을 가졌던 과거 컨텐츠와 내용적으로 유사한 컨텐츠를 추천하는 방법이다. 컨텐츠 기반 필터링은 사용 이력이 존재하지 않더라도 컨텐츠의 유사성에 기반하여 추천을 수행하기 때문에 협력 필터링의 콜드 스타트 문제를 효과적으로 해소할 수 있다. 이와 같은 컨텐츠 기반 필터링을 적용하기 위해서는 앱의 내용을 잘 표현할 수 있는 컨텐츠에 대한 정의가 필요하다.

대표적인 모바일 앱 스토어인 구글 플레이 스토어와 애플의 앱스토어는 앱의 특징을 잘 표현할 수 있는 앱의 설명이 기술되어 있다. 이들 내용은 개발자가 직접 앱의 특징을 기술한 내용이므로 컨텐츠 기반 필터링 방법에 사용하기에 적합하다. 하지만 모바일 앱 설명의 모든 부분이 앱의 특징을 설명하고 있는 것은 아니다. 예를 들어 그림 1은 앱 설명에서 실질적으로 앱과 관계없는 노이즈에 대한 예를 보여주고 있다. 앱의 설명 부분은 앱의 직접적인 내용이나 특징을 설명하기도 하지만 때로는 앱의 내용과는 무관한 이벤트, 공지사항 등의 노이즈 내용이 포함되어 있다. 노이즈 설명들과 앱의 설명을 표현하고 있는 부분들은 서로 다른 문단에 포함되어 구성되어 있음을 볼 수 있다. 노이즈 문단들은 전체 앱 설명에 섞여 앱 설명들 사이의 유사도를 바탕으로 앱의 유사성을 측정하는데 방해가 된다,

본 논문에서는 모바일 앱 추천의 첫 단계로 앱 설명의 노이즈를 제거하기 위하여 Latent Dirichlet Allocation (LDA)와 Support Vector Machine (SVM) 모델에 기반한 앱 설명 문단 노이즈 제거 모델을 제안한다. 개발자가 작성한 앱 설명에는 앱의 특징을 설명하는 문단들과 앱의 특징과는 무관한 노이즈 문단들이 혼재되어 나타난

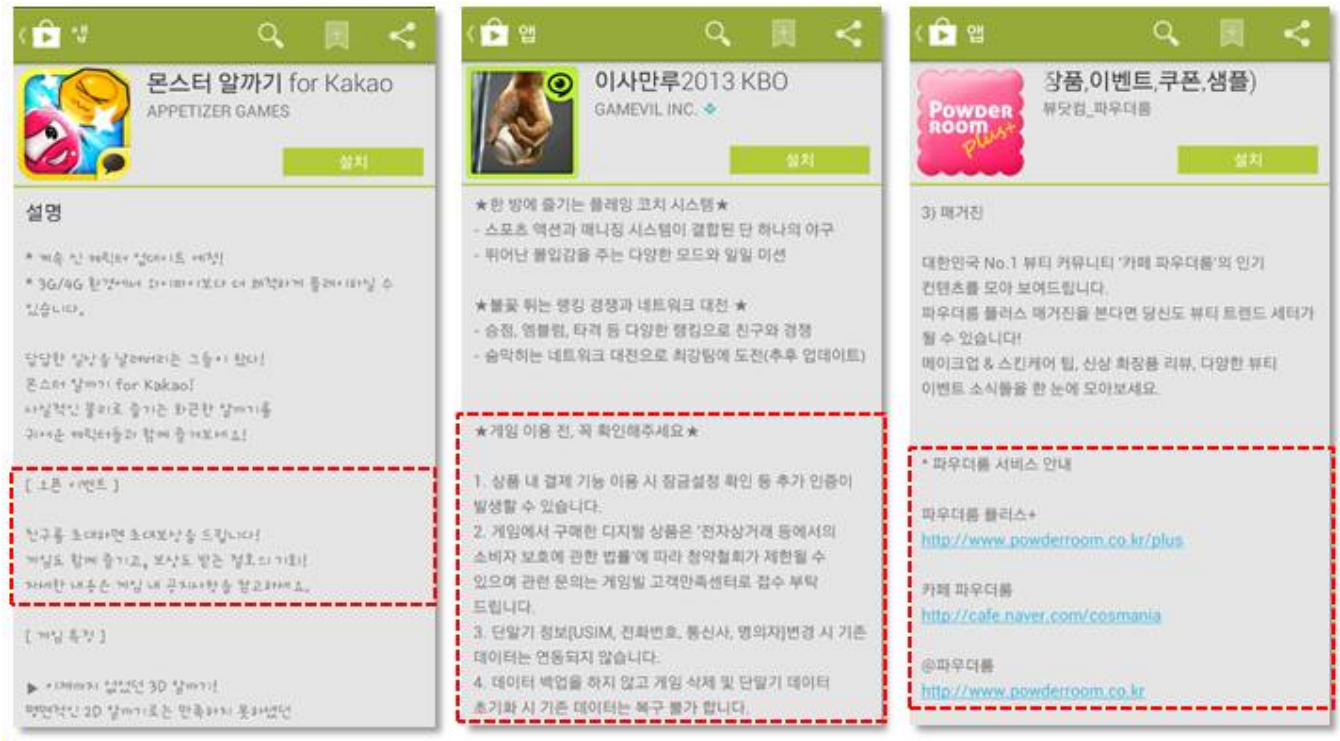


그림 1. 앱 설명에 포함되어 있는 노이즈 문단의 예

다. 일반적으로 앱 설명들은 일반적인 문서 분류 문제에서 사용되는 문서들에 비해 길이가 짧다는 특징이 있다. 특히 본 논문에서 분류의 대상으로 삼고 있는 단위는 문단이기 때문에 훨씬 적은 수의 단어로 구성되어 있다. 이런 이유로 문서 분류 문제에서 자주 적용되는 Bag-of-Word(BOW) 기반의 문서 표현법을 사용할 경우 데이터 부족 문제 (data sparseness problem)으로 인해 성능이 저하되는 문제가 발생할 수 있다. 이에 본 논문에서 제안하는 방법은 토픽 모델에 기반하여 앱 설명을 토픽들의 구성 비율로 표현하고 이를 바탕으로 SVM을 이용하여 분류하는 방법론을 제안한다. 실험 결과에 따르면 제안한 방법은 약 61%의 정확도를 보이는데, 이는 약 54%의 성능을 보이는 BOW 기반의 표현법을 사용하는 모델보다 훨씬 나은 성능을 보이며 제안한 모델이 앱 설명에서 노이즈를 제거하기에 적합함을 보여준다.

본 논문의 구성은 다음과 같다. 2장에서는 모바일 앱 추천 시스템과 토픽 모델링에 대한 기존 연구를 살펴보고, 3장에서는 LDA 모델에 기반한 문서 표현법과 이를 바탕으로 노이즈 문단을 분류하는 방법에 대하여 설명한다. 4장에서는 제안한 모델의 성능을 평가하기 위한 실험을 설명하고 성능을 평가한다. 5장에서는 결론을 다룬다.

2. 관련 연구

최근 모바일 앱 시장의 폭발적인 성장으로 인하여 모바일 앱과 관련한 다양한 연구들이 이루어지고 있다. 특히 모바일 앱 추천 또한 큰 관심을 받고 있다. 노우현 [1] 등은 사용자의 앱 사용 이력과 상황을 고려하여 앱의 카테고리를 추천하는 모델을 제안하였다. 해당 모델에서는 각 상황별 각 카테고리의 적합도를 계산하기 위하여 베이지안 모델을 이용하였다.

많은 기존의 연구들은 협력적 필터링 기반 방법을 통한 추천을 제안하였다. Yan et al.[2] 은 협력적 필터링 기반 시스템 AppJoy를 소개하였다. AppJoy는 자체적으로 개발한 스마트폰 앱을 통해 사용자 사용 이력들을 수집하고, 협력적 필터링을 통하여 모바일 앱의 추천을 수행한다. 사용자의 앱 사용 이력의 누락으로 인한 콜드 스타트 문제를 풀기 위하여 새 사용자의 경우 초기에 사용자의 기기에 설치되어 있는 앱들을 바탕으로 추천을 수행하였다. Ozaki et al.[3]은 기존의 협력적 필터링과 사용자들 사이의 사회적인 관계까지 함께 고려하는 시스템을 제안하였다. 이 시스템에서는 일반적인 협력적 필터링 방법에 기반하여 추천 앱을 선정한 뒤, 다시 사용자들 사이의 사회적인 관계를 반영하여 앱 추천 스코어를 재평가하는 방법으로 적용하였다. 이 두 방법 모두 높은 성능을 보여주었으나, 협력적 필터링 방법의 한계로 인하여 새롭게 출시되어 평가 이력이 존재하지 않는 앱들에 대해서는 추천을 수행할 수 없는 문제가 존재하였다.

Lin et al.[4]은 협력적 필터링 방법의 콜드 스타트 문제를 해결하기 위하여 외부 자원을 활용하는 방법을 소개하였다. 이 시스템에서는 대표적인 SNS서비스인

twitter를 이용하여 사용자의 사용 이력과 앱의 평가 이력의 부족을 해소하고자 하였다. 이 논문은 앱의 추천을 위해서 외부자원을 효율적으로 활용하는 방안을 제시하였으나, 이 역시도 앱 추천을 위해서 앱 이외의 자원에 관한 이력이 존재해야 한다는 면에서 콜드 스타트 문제를 원천적으로 해결하지는 못하였다.

Kim et al.[5]은 모바일 앱 추천을 위하여 콘텐츠 기반 유사도 방법론을 소개하였다. 이 모델에서는 모바일 앱의 콘텐츠를 정의하기 위하여 모바일 앱을 위한 온톨로지를 설계하고 이를 바탕으로 각 앱의 콘텐츠를 정의하였다. 앱에 관하여 개발자, 이름, 다운로드 수 등 다양한 정보를 반영하였다. 하지만 앱 개발자가 직접적으로 앱 특징에 대하여 기술한 앱 설명 정보를 활용하지 않아 많은 정보 손실을 야기하였다.

문서 분류 문제에 토픽 모델을 도입하기 위하여 다양한 연구가 이루어졌다. Rubin et al.[6]은 멀티레이블 문서를 분류하기 위한 토픽 모델을 제안하였다. 기존 LDA를 멀티레이블 문서 분류에 사용하기 위하여 확장한 Flat-LDA, Prior-LDA, Dependency-LDA 모델을 제안하였다. 비록 판별 모델인 SVM에 비해 낮은 성능을 보여주긴 하였지만 그 성능 차이가 크지 않아 토픽 모델을 이용하여 멀티레이블 문서 분류를 구생할 수 있음을 보여주었다. Zhou et al.[7] 역시 LDA에 기반한 LDACLM을 제안하였다.

3. 앱 설명의 노이즈 제거

분류 모델을 이용하여 분류하기 위해서는 문서들을 컴퓨터로 처리할 수 있는 형태로 표현할 수 있는 방법이 필요하다. 본 장에서는 LDA에 기반하여 앱 설명 문단들을 벡터로 표현하는 방법과 이를 이용하여 노이즈 문단을 분류할 수 분류 방법을 설명한다.

3.1. Latent Dirichlet Allocation

LDA는 Blei[8]에 의해서 제안된 대표적인 토픽 모델 중 하나로, 하나의 문서는 다양한 토픽의 혼합으로 구성되어 있다고 가정하는 모델이다. LDA에서는 문서들이 자신이 가진 토픽들로부터 생성된 단어들로 표현된다고 본다. 여기서 토픽이란 문서를 구성하는 단어들 중 서로 연관성이 높은 단어들의 집합으로 볼 수 있다. 이러한 토픽들은 다항분포로서 정의되며 각 문서는 자신이 가진 토픽의 분포와 각 토픽들이 가진 단어들의 분포에 기반하여 추출된 단어들로 구성된다. 그리고 이렇게 추출된 단어들이 나열되어 해당 문서가 작성되는 것으로 본다. 그림 2는 LDA의 그래피컬 표현을 나타낸다. LDA는 다음과 같은 문서 생성 과정을 모델링한다.

- 문서 d가 가지고 있는 토픽들의 다항분포 θ_d를 Dirichlet 분포인 $\theta_d \sim Dir(\alpha)$로부터 추출한다.
- 각 토픽 k에 대한 단어들의 다항분포 Φ_k를 Dirichlet 분포인 $\Phi_k \sim Dir(\beta)$로부터 추출한다.
- 이를 바탕으로 i번째 단어 w_i는 다음과 같이 추출된다.

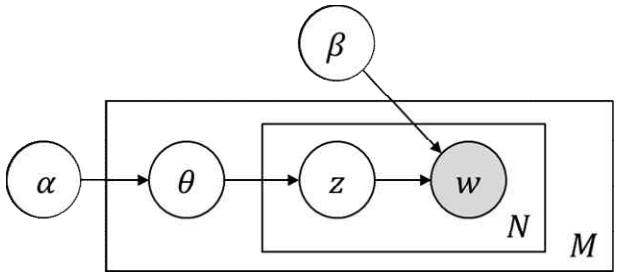


그림 2. LDA의 그래피컬 표현

- i번째 토픽 z_i를 다항분포 $z_i \sim Multinomial(\theta_d)$로부터 추출한다.
- i번째 단어 w_i를 다항분포 $w_i \sim Multinomial(\Phi_{z_i})$로부터 추출한다.

이 과정에서 모델이 추정해야 할 값은 하이퍼 파라매터인 α와 β이다. LDA는 주어진 학습 문서들을 이용하여 EM기반의 방법으로 로그 우도가 최대가 되는 α와 β를 찾는다. 이렇게 α와 β가 학습이 되면 새로운 문서의 단어 집합 $\mathbf{w}$가 주어질 때, 이 문서의 토픽들의 분포인 θ는 다음과 같이 구해질 수 있다.

$$p(\theta,\mathbf{z}|\mathbf{w},\alpha,\beta)=\frac{p(\theta,\mathbf{z},\mathbf{w}|\alpha,\beta)}{p(\mathbf{z}|\alpha,\beta)}$$

여기서 구해진 새로운 문서의 토픽 분포 θ는 해당 문서를 구성하고 있는 단어들을 대신하여 해당 문서를 벡터로 표현하는데 사용될 수 있다.

본 논문에서 제안하는 방법은 LDA 모델에 기반하여 앱 설명 문단들을 단어들이 아닌 각 토픽의 구성 비율로 표현한다. 기존에 문서 분류에서 자주 사용되는 BOW의 경우 문단의 길이가 짧고 데이터의 수가 많지 않을 경우에 데이터 부족 현상 때문에 성능이 저하되는 문제가 있다. 특히 본 논문에서 분류 대상으로 삼고 있는 단위는 앱 설명의 문단이기 때문에 그 길이가 매우 짧아 데이터 부족 현상에 의해 큰 영향을 받을 수 있다. 이에 본 논문에서는 BOW 모델 대신에 LDA 모델을 이용하여 문서를 어휘 수 보다 훨씬 적은 차원인 토픽으로 표현함으로써 데이터 부족 문제에 좀 더 강건한 데이터를 생성할 수 있도록 한다.

3.2. Support Vector Machine

SVM은 Vapnik[9]의 의해서 제안된 모델로 매우 우수한 성능을 보이는 이진 분류 모델 중 하나이다. SVM은 두 클래스의 데이터가 주어졌을 때, 두 클래스의 데이터를 잘 분류할 수 있는 초평면(hyperplane)을 찾는 모델이다. 경우에 따라서 두 클래스를 완전히 구분할 수 있는 초평면이 매우 많거나 또는 무한하게 존재할 수 있는데,

표 1. 실험 데이터 통계

속성	값
카테고리의 수	25
앱의 수	703
문단의 수	4,500
문단 별 평균 단어 수	24.33
노이즈 문단 수	2,104
비 노이즈 문단 수	2,396

이때 초평면에 가장 가까운 각 클래스의 데이터와 초평면 사이의 거리를 의미하는 마진(margin)이 가장 최대가 되는 초평면을 찾는다. 이렇게 찾은 초평면을 이용하여 새로운 데이터가 주어졌을 때, 해당 데이터의 클래스를 분류한다.

데이터 집합 $D=\left\{(\mathbf{x_i},y_i)|\mathbf{x_i}\in R^m, y_i\in\{-1,+1\}\right\}_{i=1}^{n}$가 주어지면, SVM의 초평면은 다음과 같이 정의될 수 있다.

$$H=\mathbf{w}\cdot\mathbf{x}+\mathrm{b}$$

SVM에서는 $y_i(\mathbf{w}\cdot\mathbf{x_i}-b)\geq 1$을 만족하면서 $\|\mathbf{w}\|$이 최소화되는 $\mathbf{w}$를 찾음으로써 최적의 초평면을 찾는다. 그리고 이렇게 학습된 초평면을 이용하여 새로운 데이터 x가 주어지면 다음과 같이 새로운 데이터의 클래스를 추정한다.

$$y=\begin{cases}1 & \text{if } w\cdot x_i-b\geq 1\\ -1 & \text{if } w\cdot x_i-b\leq -1\end{cases}$$

4. 실험

본 논문에서 제안한 방법의 성능을 보이기 위하여 실제 모바일 앱의 설명들을 수집하여 실험을 수행하였다. 실험을 위하여 플레이 스토어[1]에 등록되어 있는 모바일 앱들의 설명을 수집하였다. 표 2는 실험에 사용된 데이터의 통계를 보여준다. 플레이 스토어에 존재하는 전체 25개 카테고리의 703개의 앱에 대한 설명을 수집하였다. 이들 703개의 앱 설명을 html 태그에 기반하여 문단 단위로 분리하였다. 각 앱 설명에서 특수기호 및 URL은 제거하였다. 길이가 짧은 문단의 경우 앱 설명의 각 문단의 제목을 나타내는 경우가 대부분이기 때문에 문단 내에 포함된 음절의 길이가 15이하인 문단은 제외하였다. 최종적으로 남은 4,500개의 문단에 대하여 노이즈 여부를 수작업으로 판단하였다. 사용자가 모바일 앱의 카테고리와 문단을 함께 고려하여 각 문단의 노이즈 여부를 판단하였다. 실험에 사용된 문단들은 평균적으로 24.33개의 단어로 구성되어 있으며 전체 테스트 데이터 중 약 47%가 노이즈 문단으로 구성되어 있다. 전체 4,500개의 데이터 중 80%는 모델의 학습에 사용되었고 나머지 20%는 성능 측정하기 위한 테스트 데이터로 사용되었다.

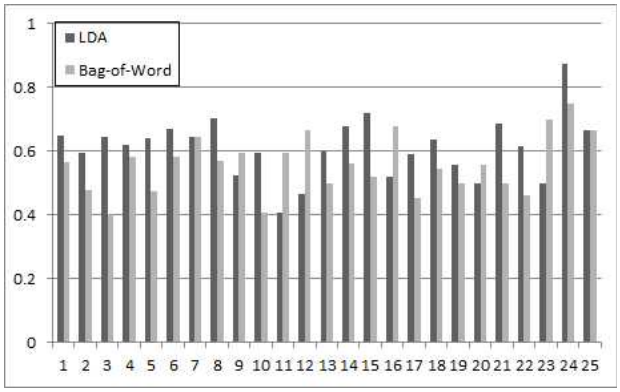


그림 3. 카테고리별 분류 정확도

표 2. 전체 카테고리에 대한 정확도

문단 표현 방법	정확도 (%)
Bag-of-Word	54.09
LDA	61.52

LDA 모델의 파라매터 중 토픽의 수는 100개로 지정하였다. 그리고 각 분산의 하이퍼 파라매터(Hyper Parameter) α와 β의 초기값은 0.1로 지정하였으며 학습 반복 횟수는 총 1,000회로 지정되어 학습되었다. SVM 분류모델에서 LDA 표현 기반 데이터는 다항 커널을 이용하였고 BOW 표현에 기반한 데이터는 선형 커널을 이용하여 수행하였다. 실험에 사용된 모든 파라매터는 실험적으로 각 모델에서 가장 우수한 성능을 보여준 값으로 선택되었다.

실험은 수집된 데이터 셋에 포함된 총 25개 카테고리에 대해서 독립적으로 수행되었다. 그림 3은 각 카테고리별 노이즈 문단의 분류 성능을 보여준다. 그래프의 가로축 카테고리 목록과 각 카테고리 별 데이터 비율은 표 4에 나타나 있다. 제안한 방법으로 표현된 문서 기반 분류 성능은 BOW로 표현된 데이터에 비하여 훨씬 많은 카테고리에서 더 높은 성능을 보여주었다. 특히 실험데이터에서 큰 비중을 차지하고 있는 카테고리들에서 비슷하거나 더 높은 분류 정확도를 보여주었다. 표 3은 전체 데이터 집합에 대한 분류 정확도를 보여준다. BOW 모델을 이용하여 표현된 문단들은 54.09%의 정확도를 보여주었다. 하지만 본 논문에서 제안하는 LDA에 기반한 문단들은 61.52%의 분류 정확도를 보여주었다. 비록 LDA에 기반한 모델이 BOW 표현에 기반한 모델보다 우수한 성능을 보여주긴 하나, 전반적으로 낮은 정확성을 보여주었다.

낮은 정확성의 원인을 파악하기 위하여 오류 데이터를 분석해보았다. 해당 데이터들을 분석해본 결과 많은 양의 앱들이 잘못된 카테고리로 분류되어 있는 것을 확인할 수 있었다. 앱의 카테고리는 개발자의 의해서 결정되는 것으로 명확한 규정이 존재하지 않아 오분류된 앱들

1) http://play.google.co.kr

표 3. 실험에 사용된 앱 카테고리 및 데이터 수

번호	카테고리	데이터 수	번호	카테고리	데이터 수	번호	카테고리	데이터 수
1	게임	707	2	교육	478	3	라이프스타일	309
4	데코레이션	274	5	도서 및 참조자료	264	6	도구	234
7	음악 및 오디오	220	8	엔터테인먼트	216	9	여행 및 지역정보	208
10	커뮤니케이션	157	11	소셜 콘텐츠	154	12	미디어 및 동영상	148
13	건강 및 운동	146	14	비지니스	122	15	만화	122
16	사진	122	17	생산성	110	18	금융	108
19	교통	86	20	스포츠	86	21	의료	76
22	쇼핑	61	23	날씨	44	24	뉴스 및 잡지	35
25	라이브러리 및 데모	13						

이 매우 많이 존재한다. 앱의 설명에 기술되어 있는 문단들은 내용이 비슷해 보이더라도 카테고리에 따라 다르게 해석될 수 있다. 이에 본 논문에서는 노이즈 분류를 위한 모델을 카테고리별로 구축하여 실험을 수행하였는데, 잘못된 카테고리에 포함된 수많은 앱들이 영향으로 인해 분류 성능이 낮음을 확인할 수 있었다. 예를 들어 'ZLOTUS는 GO 실행기 테마를 사랑' 이라는 앱은 핸드폰을 꾸미기 위한 테마 앱으로 데코레이션 또는 도구 카테고리가 존재하지만, 실제로 이 앱은 만화 카테고리에 포함되어 있었다. 이런 앱들의 영향으로 인하여 BOW와 LDA에 기반한 모델 모두 전반적으로 낮은 분류 정확성을 보여주었다. 하지만 본 논문에서 제안한 모델은 동일한 환경 하에서 BOW에 비하여 더 우수한 분류 성능을 보여주어 앱 설명에서 노이즈를 제거하기에 적합함을 보여주었다.

5. 결론

본 논문에서는 앱 설명의 노이즈를 제거하기 위하여 앱 설명을 토픽으로 표현하여 분류하는 모델을 제안하였다. 제안한 방법은 앱 설명의 각 문단을 대표적인 토픽 모델인 LDA를 통해 표현하고 이들 데이터를 분류 문제에서 우수한 성능을 보여주는 SVM을 이용하여 분류한다. 제안한 방법은 앱 설명에 사용된 단어를 그대로 사용하지 않고 이를 토픽으로 표현함으로써 데이터 부족 문제에 대하여 강건한 노이즈 제거 방법이다. 이를 통해 BOW 모델에 비하여 우수한 성능을 보여주었다. 실험 결과에 의하면 BOW에 기반한 모델은 모든 카테고리에 대하여 54.09%의 정확도를 보여주는 반면 제안한 모델은 61.52%의 성능을 보여주었다. 비록 앱들의 카테고리 오분류 문제에 의해 전반적으로 낮은 분류 성능을 보여주었지만, 모바일 앱의 설명에서 노이즈 문단을 제거하기에 BOW에 기반한 모델보다 본 논문에서 제안한 LDA에 기반한 모델이 더욱더 적합함을 보여주었다.

현재 앱 마켓에는 이미 수백만 개의 앱들이 등록되어 있으며 이들을 설명하고 있는 문서의 수도 매우 방대하다. 이들 앱 설명을 수집하는 것은 어렵지 않으나 모델의 학습을 위하여 각 문단의 정답을 수작업으로 부착하는 것은 매우 큰 비용이 발생하는 문제이다. 향후 연구로 이미 존재하는 대량의 앱 설명을 큰 비용 없이 효율적으로 활용하기 위하여 반지도 또는 비지도 학습 방법에 기반한 분류 모델을 연구할 예정이다. 또한 본 논문의 실험에서 성능저하를 일으킨 앱의 카테고리 오분류에 의한 성능저하를 효율적으로 해결할 수 있는 모델에 대한 연구 또한 함께 진행할 예정이다.

감사의 글

본 논문은 지식경제부 산업원천기술개발사업(10035348, 모바일 플랫폼 기반 계획 및 학습 인지 모델 프레임워크 기술 개발)의 지원으로 수행되었음.

참고문헌

[1] 노우현, 조성배, "베이지안 네트워크를 이용한 상황별 모바일 앱 카테고리 추천 시스템", 한국정보과학회 2013 한국컴퓨터종합학술대회 논문집, pp.1408-1410, 2013.

[2] B. Yan and G. Chen, "AppJoy: Personalized Mobile Application Discovery", MobiSys '11 Proceedings of the 9th international conference on Mobile systems, applications, and services, pp.113-126, 2011.

[3] T. Ozaki and M. Ehoh, "Experimental Analysis of the Effects of Social Relations on Mobile Application Recommendation", Proceedings of the International MultiConference of Engineers and COmputer Scientists, 2012.

[4] J. Lin, K. Sugiyama, M. Kan and T. Chua, Addressing Cold-Start in App Recommendation: Latent User Models Constructed from Twitter Followers", Proceedings of SIGIR 2013, 2013.

[5] J. Kim, S. Kang, Y. Lim and H. Kim, "Recommendation algorithm of the app store by using semantic relations between apps", The Journal of Supercomputing, vol.65, pp.16-26, 2011.

[6] Timothy N. Rubin, America Chambers, Padhraic Smyth and Mark Steyvers, "Statisitcal topic

models for multi-label document classification", Journal of Machine Learning, vol.88, pp.157-208, 2012.
[7] Shibin Zhou, Kan Li and Yushu Liu, "Text Categoization Based on Topic Model", International Journal of Computational Intelligence Systems, vol.2, no.4, pp.398-409, 2009.
[8] David Blei, Andrew Ng, Michael Jordan, "Latent Dirichlet allocation", Journal of Machine Learning Research, vol.3, pp.993-1022, 2003.
[9] Corinna Cortes and Vladimir N. Vapnik, "Support-Vector Networks", Machine Learning, vol.20, 1995.

Latent Structural SVM을 확장한 결합 학습 모델

이창기O
강원대학교
leeck@kangwon.ac.kr

Jointly Learning Model using modified Latent Structural SVM

Changki LeeO
Kangwon National University

요 약

자연어처리에서는 많은 모듈들이 파이프라인 방식으로 연결되어 사용되나, 이 경우 앞 단계의 오류가 뒷 단계에 누적되는 문제와 앞 단계에서 뒷 단계의 정보를 사용하지 못한다는 단점이 있다. 본 논문에서는 파이프라인 방식의 문제를 해결하기 위해 사용되는 일반적인 결합 학습 방법을 확장하여, 두 작업이 동시에 태깅된 학습 데이터뿐만 아니라 한 작업만 태깅된 학습데이터도 동시에 학습에 사용할 수 있는 결합 학습 모델을 Latent Structural SVM을 확장하여 제안한다. 실험 결과, 기존의 한국어 띄어쓰기와 품사 태깅 결합 모델의 품사 태깅 성능이 96.99%였으나, 본 논문에서 제안하는 결합 학습 모델을 이용하여 대용량의 한국어 띄어쓰기 학습데이터를 추가로 학습한 결과 품사 태깅 성능이 97.20%까지 향상 되었다.

주제어: 결합 학습 모델(Jointly Learning Model), Latent Structural SVM, 한국어 띄어쓰기, 품사 태깅

1. 서론

자연어처리에서는 많은 모듈들이 파이프라인 방식으로 연결되어 사용되어 왔으며(예를 들어, 단어 분리와 품사 태깅, 구문 분석 등), 이러한 경우 앞 단계의 오류가 뒷 단계에 누적되는 문제와 앞 단계에서 뒷 단계의 정보를 사용하지 못한다는 단점이 있다 [1, 2]. 이러한 문제를 해결하기 위해 최근에 중국어와 일본어 등에 대해서 단어 분리(word segmentation)와 품사 태깅의 결합 학습(jointly learning) 모델에 대한 많은 연구가 수행되었다[1,2,3,4].

그러나 대부분의 결합 학습(jointly learning) 모델들은 두 모듈의 태깅이 같이 되어있는 학습 데이터를 필요로 하고 있다. 예를 들어, 중국어의 단어 분리와 품사 태깅의 결합 학습을 위해서 단어 분리와 품사 태그의 정답이 동시에 태깅된 학습 데이터를 사용하고 있다[1,2,3,4,]. 그러나 일반적으로 두 작업(task) 중에서 한쪽 작업의 학습 데이터 구축이 쉬우며, 한쪽 작업의 대용량의 학습 데이터가 이미 존재하는 경우가 많다. 예를 들어, 한국어 띄어쓰기의 경우 띄어쓰기가 잘 된 신문 기사 등으로부터 대용량의 학습데이터를 쉽게 구축할 수 있으나, 기존의 한국어 띄어쓰기와 품사 태깅 결합 학습 연구에서는 한국어 띄어쓰기와 품사 태그가 동시에 태깅된 학습 데이터만을 사용하여 결합 학습을 수행하였다[4].

본 논문에서는 파이프라인 방식의 문제를 해결하기 위해 사용되는 일반적인 결합 학습 방법을 확장하여, 두 작업이 동시에 태깅된 학습 데이터뿐만 아니라 한 작업만 태깅된 학습데이터도 동시에 학습에 사용할 수 있는 결합 학습 모델을 제안한다. 본 논문에서 제안하는 결합 학습 모델은 기존의 Structural SVM에 은닉 변수가 추가된 Latent Structural SVM을 확장하여 사용한다.

실험 결과, 기존의 Structural SVM 기반의 한국어 띄어쓰기와 품사 태깅 결합 모델의 품사 태깅 성능이 96.99%였으나 (테스트 문장의 띄어쓰기를 제거한 경우의 성능임), 본 논문에서 제안하는 결합 학습 모델을 이용하여 대용량의 한국어 띄어쓰기 학습데이터를 추가로 학습한 결과 품사 태깅 성능이 97.20%까지 향상 되었다.

2. 관련 연구

자연어처리에서는 단어 분리, 품사 태깅, 구문 분석, 의미 분석 등의 각 단계 모듈이 주로 파이프라인 방식으로 연결 되어 사용되어 왔으며, 최근에 파이프라인 방식의 문제를 해결하기 위해서 결합 모델이 주로 연구되고 있다[1,2,3,4].

중국어와 일본어에 대해서 단어 분리(word segmentation)와 품사 태깅의 결합 학습 모델(jointly learning)에 대한 많은 연구가 수행되었다[1,2]. [1]에서는 N-best reranking 방법을 사용하여, 앞 단계인 중국어 단어 분리(word segmentation)의 N-best 결과에 품사 태깅을 수행한 후 재순위화(reranking)를 통해 파이프라인 방식의 문제를 해결하려 했으나, 단어 분리의 N-best 결과만 사용한다는 한계가 있고 N-best 결과 모두에 품사 태깅을 수행하므로 속도 저하 문제가 발생했다. [2]에서는 중국어 단어 분리와 품사 태깅 문제를 한 음절에 대한 sequence labeling 문제로 접근하였고, 음절에 대한 태그는 단어 분리 태그와 품사 태그가 결합한 형태로 쓰여 태그 수가 증가하여 검색 공간이 증가되는 문제가 발생했다. 이 밖에도 영어에 대해서 개체명 인식과 구문 분석을 동시에 수행하는 연구가 있었다[3].

기존의 대부분의 결합 학습 연구들은 두 작업의 정답

이 동시에 태깅된 학습 데이터를 사용하고 있으나, 본 논문에서 제안하는 결합 학습 모델은 두 작업의 정답이 태깅된 학습 데이터뿐만 아니라 한 작업의 학습 데이터만 태깅된 학습 데이터도 같이 사용하여 성능을 향상시킬 수 있다.

3. Latent Structural SVM을 확장한 결합 학습 모델

본 논문에서는 은닉 변수가 추가된 Latent Structural SVM을 확장한 결합 학습 모델을 제안한다.

3.1 Structural SVM

Structural SVM은 기존의 SVM을 확장한 기계학습 알고리즘으로, 기존의 SVM이 바이너리 분류, 멀티클래스 분류 등을 지원하는 반면에, structural SVM은 더욱 일반적인 구조의 문제(예를 들어, sequence labeling, 구문분석 등)를 지원하며 다음과 같이 정의할 수 있다[5].

$$\min_{\mathbf{w}} \frac{1}{2} \| \mathbf{w} \|^2 + \frac{C}{n} \sum_{i}^{n} [\max_{\mathbf{y} \in \mathbf{Y}} \{\Delta(\mathbf{y}_i, \mathbf{y}) + \mathbf{w} \cdot \Phi(\mathbf{x}_i, \mathbf{y})\} - \mathbf{w} \cdot \Phi(\mathbf{x}_i, \mathbf{y}_i)] \quad (1)$$

위 식에서 $(\mathbf{x}_i, \mathbf{y}_i)$는 학습데이터의 i번째 문장과 그에 대응되는 정답 태그 열을 나타내고, $\Delta(\mathbf{y}_i, \mathbf{y})$는 정답 태그 열 $\mathbf{y}_i$와 예측결과 태그 열 $\mathbf{y}$사이의 다른 태그 개수를 반환하며, $\Phi(\mathbf{x}, \mathbf{y})$는 자질(feature) 벡터 함수를 나타낸다.

3.2 Latent Structural SVM

Latent Structural SVM은 기존의 Structural SVM에 은닉 변수(hidden variable) z를 추가한 것으로, 다음과 같이 정의된다[6].

$$\min_{\mathbf{w}} \frac{1}{2} \| \mathbf{w} \|^2 + \frac{C}{n} \sum_{i}^{n} \left[\max_{(\mathbf{y},\mathbf{z}) \in \mathbf{Y} \times \mathbf{Z}} \{\Delta(\mathbf{y}_i, \mathbf{y}, \mathbf{z}) + \mathbf{w} \cdot \Phi(\mathbf{x}_i, \mathbf{y}, \mathbf{z})\} \right] - \frac{C}{n} \sum_{i}^{n} \max_{\mathbf{z} \in \mathbf{Z}} \mathbf{w} \cdot \Phi(\mathbf{x}_i, \mathbf{y}_i, \mathbf{z}) \quad (2)$$

위 식에서 $\Delta(\mathbf{y}_i, \mathbf{y}, \mathbf{z})$는 정답 태그 열 $\mathbf{y}_i$와 예측결과 태그 열 $\mathbf{y}$와 은닉 변수 $\mathbf{z}$를 입력으로 받는 loss 함수로 일반적으로 은닉변수 $\mathbf{z}$는 무시하여 $\Delta(\mathbf{y}_i, \mathbf{y})$와 같아지고 (즉, $\mathbf{y}_i$와 $\mathbf{y}$사이의 다른 태그 개수), $\Phi(\mathbf{x}, \mathbf{y}, \mathbf{z})$는 은닉 변수 $\mathbf{z}$가 추가된 자질(feature) 벡터 함수를 나타낸다. 은닉 변수 $\mathbf{z}$를 무엇으로 정의하냐에 따라 자질 벡터 함수 $\Phi(\mathbf{x}, \mathbf{y}, \mathbf{z})$가 달라지게 되며 이를 사용자가 정의해 주어야 한다.

Latent Structural SVM은 은닉 변수가 추가 되어 더 이상 convex optimization 문제가 아니기 때문에 유일한 해가 존재하지 않고 지역 극소점(local minima)에 빠질 수 있다. 또한 학습을 위해서 기존의 Structural SVM의 학습 알고리즘을 그대로 사용 못하고 Concave-Convex Procedure (CCCP) 알고리즘을 사용한다[6].

3.3 Latent Structural SVM을 확장한 결합 학습 모델

기존의 결합 학습 모델은 대부분 두 개의 작업(task)이 동시에 태깅된 학습데이터를 사용하였다 (예를 들어, 단어 분리(word segmentation) + 품사 태깅). 그러나 이렇게 두 개의 작업이 동시에 태깅된 학습 데이터를 구축하는 것은 비용과 시간이 많이 걸리며, 일반적으로 한쪽 작업(task)만 태깅된 학습데이터가 기존에 존재할 수도 있고, 그렇지 않은 경우에도 한쪽 작업만 태깅된 학습데이터를 구축하는 것이 두 개의 작업이 동시 태깅된 학습데이터를 구축하는 것 보다 훨씬 비용이 적게 든다.

본 논문에서는 두 작업이 동시 태깅된 학습데이터와 한쪽의 작업만 태깅된 학습데이터를 모두 사용할 수 있도록 하기 위해서 Latent Structural SVM을 확장하여 결합 학습 모델(Jointly Learning Model)을 수식 (3)과 같이 정의한다. 아래 모델에서 두 개의 작업이 동시 태깅된 학습데이터는 1~n 문장이며 (n개의 문장), 한쪽의 작업만 태깅된 학습데이터는 n+1~n+m 문장이다 (m개의 문장). 두 개의 작업이 동시 태깅된 학습데이터(1~n 문장) 부분에서는 일반 Structural SVM과 유사하며(두 작업이 동시 태깅된 것을 하나의 합쳐진 태그로 가정하고 학습), 한쪽의 작업에만 태깅이 된 학습데이터(n+1~n+m) 부분에서는 태깅이 안된 다른 작업의 태그를 은닉 변수(hidden variable)로 보고 Latent Structural SVM과 유사하게 학습한다.

$$\min_{\mathbf{w}} \frac{1}{2} \| \mathbf{w} \|^2 + \frac{C_{\mathbf{y:z}}}{n+m} \sum_{i=1}^{n} L_{\mathbf{y:z}}(\mathbf{x}_i, \mathbf{y}_i, \mathbf{z}_i; \mathbf{w}) + \frac{C_{\mathbf{y}}}{n+m} \sum_{i=n+1}^{n+m} L_{\mathbf{y}}(\mathbf{x}_i, \mathbf{y}_i; \mathbf{w}),$$

where

$$L_{\mathbf{y:z}}(\mathbf{x}_i, \mathbf{y}_i, \mathbf{z}_i; \mathbf{w}) = \max_{(\mathbf{y},\mathbf{z}) \in \mathbf{Y} \times \mathbf{Z}} \{\Delta(\mathbf{y}_i, \mathbf{z}_i, \mathbf{y}, \mathbf{z}) + \mathbf{w} \cdot \Phi(\mathbf{x}_i, \mathbf{y}, \mathbf{z})\} - \mathbf{w} \cdot \Phi(\mathbf{x}_i, \mathbf{y}_i, \mathbf{z}_i),$$

$$L_{\mathbf{y}}(\mathbf{x}_i, \mathbf{y}_i; \mathbf{w}) = \max_{(\mathbf{y},\mathbf{z}) \in \mathbf{Y} \times \mathbf{Z}} \{\Delta(\mathbf{y}_i, \mathbf{y}) + \mathbf{w} \cdot \Phi(\mathbf{x}_i, \mathbf{y}, \mathbf{z})\} - \max_{\mathbf{z} \in \mathbf{Z}} \mathbf{w} \cdot \Phi(\mathbf{x}_i, \mathbf{y}_i, \mathbf{z}). \quad (3)$$

위 식에서 $(\mathbf{x}_i, \mathbf{y}_i, \mathbf{z}_i)$는 학습데이터의 i번째 문장과 그에 대응되는 첫 번째 작업의 태그 열($\mathbf{y}_i$)및 두 번째 작업의 태그 열($\mathbf{z}_i$)을 나타내고, $\Delta(\mathbf{y}_i, \mathbf{z}_i, \mathbf{y}, \mathbf{z})$는 정답 태그 열 $\mathbf{y}_i$ 및 $\mathbf{z}_i$와 예측결과 태그 열 $\mathbf{y}$ 및 $\mathbf{z}$의 loss 함수 (정답 태그 $\mathbf{y}_i$ 및 $\mathbf{z}_i$에 대해서 두 작업의 $\mathbf{y}$ 및 $\mathbf{z}$의 태그가 모두 정답 태그와 같을 경우만 맞았다고 계산하여 틀린 개수를 반환함)를 나타내며, $\Phi(\mathbf{x}, \mathbf{y}, \mathbf{z})$는 문장 $\mathbf{x}$, 첫 번째 작업 태그 열 $\mathbf{y}$, 두 번째 작업 태그 열 $\mathbf{z}$로 구성되는 자질(feature) 벡터 함수를 나타낸다.

Latent Structural SVM과 유사하게 한쪽의 작업만 태깅된 학습데이터 부분(n+1~n+m)에서는 은닉 변수가 추가되었기 때문에 더 이상 convex optimization 문제가 아니고, 따라서 유일한 해가 존재하지 않으며, 지역 극소점(local minima)에 빠질 수 있다. 학습을 위해서 기존의 Latent Structural SVM과 같이 Concave-Convex Procedure (CCCP) 알고리즘을 사용할 수 있다[6]. 본 논문에서는 CCCP와 유사한 방식을 사용하면서 Stochastic Gradient Descent (SGD) 방식을 적용한다. 이는 SVM의 학습 알고리즘으로 제안된 Pegasos [7] 알고리즘을 은닉 변수가 추가된 결합 학습 모델(Jointly Learning Model)에 확장한 것이다. 이를 위해서 수식 (3)의 목적 함수(object function)를 학습데이터의 일부인 k개를 사용하도록 approximation 하기 위해서 아래와 같이 목적 함수 $f(\mathbf{w};A_t)$를 정의한다.

$$f(\mathbf{w};A_t) = \frac{1}{2}\|\mathbf{w}\|^2 + \frac{C_{\mathbf{y}:\mathbf{z}}}{k}\sum_{i:i\le n} L_{\mathbf{y}:\mathbf{z}}(\mathbf{x}_i,\mathbf{y}_i,\mathbf{z}_i;\mathbf{w}) + \frac{C_{\mathbf{y}}}{k}\sum_{i:i>n\wedge i\le n+m} L_{\mathbf{y}}(\mathbf{x}_i,\mathbf{y}_i;\mathbf{w}).$$

where

$$L_{\mathbf{y}:\mathbf{z}}(\mathbf{x}_i,\mathbf{y}_i,\mathbf{z}_i;\mathbf{w}) = \max_{(\mathbf{y},\mathbf{z})\in\mathbf{Y}\times\mathbf{Z}}\{\Delta(\mathbf{y}_i,\mathbf{z}_i,\mathbf{y},\mathbf{z}) + \mathbf{w}\cdot\Phi(\mathbf{x}_i,\mathbf{y},\mathbf{z})\} - \mathbf{w}\cdot\Phi(\mathbf{x}_i,\mathbf{y}_i,\mathbf{z}_i),$$

$$L_{\mathbf{y}}(\mathbf{x}_i,\mathbf{y}_i;\mathbf{w}) = \max_{(\mathbf{y},\mathbf{z})\in\mathbf{Y}\times\mathbf{Z}}\{\Delta(\mathbf{y}_i,\mathbf{y}) + \mathbf{w}\cdot\Phi(\mathbf{x}_i,\mathbf{y},\mathbf{z})\} - \max_{\mathbf{z}\in\mathbf{Z}}\mathbf{w}\cdot\Phi(\mathbf{x}_i,\mathbf{y}_i,\mathbf{z}).$$

(4)

위 식에서 A_t는 전체 학습셋(1~n+m 문장) 중에서 임의의 k개의 문장을 선택한 것이다. 위 (4) 식의 목적 함수 $f(\mathbf{w};A_t)$의 gradient를 구하면 다음과 같다.

$$\nabla f(\mathbf{w};A_t) = \mathbf{w} + \frac{C_{\mathbf{y}:\mathbf{z}}}{k}\sum_{i:i\le n}\{\Phi(\mathbf{x}_i,\hat{\mathbf{y}}_i,\hat{\mathbf{z}}_i) - \Phi(\mathbf{x}_i,\mathbf{y}_i,\mathbf{z}_i)\} + \frac{C_{\mathbf{y}}}{k}\sum_{i:i>n\wedge i\le n+m}\{\Phi(\mathbf{x}_i,\bar{\mathbf{y}}_i,\bar{\mathbf{z}}_i) - \Phi(\mathbf{x}_i,\mathbf{y}_i,\mathbf{z}_i^*)\}$$

where

$$(\hat{\mathbf{y}}_i,\hat{\mathbf{z}}_i) = \arg\max_{(\mathbf{y},\mathbf{z})\in\mathbf{Y}\times\mathbf{Z}}\{\Delta(\mathbf{y}_i,\mathbf{z}_i,\mathbf{y},\mathbf{z}) + \mathbf{w}\cdot\Phi(\mathbf{x}_i,\mathbf{y},\mathbf{z})\},$$

$$(\bar{\mathbf{y}}_i,\bar{\mathbf{z}}_i) = \arg\max_{(\mathbf{y},\mathbf{z})\in\mathbf{Y}\times\mathbf{Z}}\{\Delta(\mathbf{y}_i,\mathbf{y}) + \mathbf{w}\cdot\Phi(\mathbf{x}_i,\mathbf{y},\mathbf{z})\},$$

$$\mathbf{z}_i^* = \arg\max_{\mathbf{z}\in\mathbf{Z}}\mathbf{w}\cdot\Phi(\mathbf{x}_i,\mathbf{y}_i,\mathbf{z}),$$

(5)

식 (5)를 이용하여 gradient descent 방식의 weight vector $\mathbf{w}$의 업데이트 식을 구하면 다음과 같다 (learning rate $\eta=1/t$ 가정).

$$\mathbf{w}_{t+1} = \mathbf{w}_t - \eta\cdot\nabla f(\mathbf{w}_t;A_t) = \left(1-\frac{1}{t}\right)\mathbf{w}_t + \frac{1}{t}\cdot\frac{C_{\mathbf{y}:\mathbf{z}}}{k}\sum_{i:i\le n}\{\Phi(\mathbf{x}_i,\mathbf{y}_i,\mathbf{z}_i) - \Phi(\mathbf{x}_i,\hat{\mathbf{y}}_i,\hat{\mathbf{z}}_i)\} + \frac{1}{t}\cdot\frac{C_{\mathbf{y}}}{k}\sum_{i:i>n\wedge i\le n+m}\{\Phi(\mathbf{x}_i,\mathbf{y}_i,\mathbf{z}_i^*) - \Phi(\mathbf{x}_i,\bar{\mathbf{y}}_i,\bar{\mathbf{z}}_i)\}.$$

(6)

식 (6)을 이용한 결합 학습 모델(Jointly Learning Model)의 학습 알고리즘은 다음과 같다.

1 : Input: $S_{y:z}$, S_y, $C_{y:z}$, C_y, T, T_{init}, k
2: Initialize: $\mathbf{w}_1 = \mathbf{0}$
3: For $t = 1,2,\ldots,T$
4: If $t \le T_{init}$
5: Choose $A_t \subseteq S_{y:z}$, where $|A_t| = k$
6: $\forall(x_i,y_i,z_i)\in A_t$: $(\hat{y}_i,\hat{z}_i) = \arg\max_{y,z}\{\Delta(y_i,z_i,y,z) + \mathbf{w}_t\cdot\Phi(x_i,y,z)\}$
7: $\mathbf{w}_{t+1} = (1-\frac{1}{t})\mathbf{w}_t + \frac{C_{y:z}}{t\cdot k}\sum_{i:i\le n}\{\Phi(x_i,y_i,z_i) - \Phi(x_i,\hat{y}_i,\hat{z}_i)\}$
8: Else
9: Choose $A_t \subseteq (S_{y:z}\cup S_y)$, where $|A_t| = k$
10: $\forall(x_i,y_i,z_i)\in A_t\wedge(x_i,y_i,z_i)\in S_{y:z}$: $(\hat{y}_i,\hat{z}_i) = \arg\max_{y,z}\{\Delta(y_i,z_i,y,z) + \mathbf{w}_t\cdot\Phi(x_i,y,z)\}$; $\forall(x_i,y_i)\in A_t\wedge(x_i,y_i)\in S_y$: $(\bar{y}_i,\bar{z}_i) = \arg\max_{y,z}\{\Delta(y_i,y) + \mathbf{w}_t\cdot\Phi(x_i,y,z)\}$, $z_i^* = \arg\max_{z}\{\mathbf{w}_t\cdot\Phi(x_i,y_i,z)\}$
11: $\mathbf{w}_{t+1} = (1-\frac{1}{t})\mathbf{w}_t + \frac{C_{y:z}}{t\cdot k}\sum_{i:i\le n}\{\Phi(x_i,y_i,z_i) - \Phi(x_i,\hat{y}_i,\hat{z}_i)\} + \frac{C_y}{t\cdot k}\sum_{i:i>n\wedge i\le n+m}\{\Phi(x_i,y_i,z_i^*) - \Phi(x_i,\bar{y}_i,\bar{z}_i)\}$
12: Output: $\mathbf{w}_{T+1}$

4. 실험

기존의 일반적인 결합 학습 모델과 본 논문에서 제안하는 Latent Structural SVM을 확장한 결합 학습 모델의 성능 비교를 위해서, [4]와 같이 한국어 띄어쓰기 문제와 품사 태깅 문제에 결합 학습 모델을 적용하였다. 띄어쓰기와 품사 태깅이 동시에 태깅된 학습데이터는 [4]와 동일하게 세종 품사 태깅 코퍼스를 사용하였으며, 실험에 사용한 자질(feature)도 [4]와 동일하게 사용하였다. 본 논문에서 제안하는 결합 학습 모델은 추가적으로 세종 원시 코퍼스(raw corpus)를 띄어쓰기의 학습데이터로 추가 사용하였다. 이 경우, 식 (3)의 $\mathbf{x}$는 입력 문장을 의미하고, $\mathbf{y}$는 띄어쓰기 태그 열, $\mathbf{z}$는 음절 기반 품

사 태그 열을 의미한다.

표 1은 한국어 띄어쓰기 및 품사 태깅 실험 결과이다. 평가 데이터의 띄어쓰기 정보가 완벽한 경우에 품사 태깅의 성능은 98.03%(F1)이고, 평가 데이터의 띄어쓰기 정보를 제거한 한 후, 파이프라인 방식으로 띄어쓰기 적용 후 품사 태깅을 적용한 경우 품사 태깅 성능(F1)은 96.77%이였다[4]. [4]의 띄어쓰기와 품사 태깅의 결합 모델을 적용한 경우, 띄어쓰기 96.86% (음절 단위 정확도)와 품사 태깅 96.99%(F1)의 성능을 보였다. 본 논문에서 제안하는 Latent Structural SVM을 확장한 결합 모델을 띄어쓰기 추가 학습데이터 50만 문장과 함께 적용한 경우 띄어쓰기 97.17%와 품사 태깅 97.10%를 보였으며, 추가 학습데이터 180만 문장과 함께 적용한 경우 띄어쓰기 97.33%와 품사 태깅 97.20%를 보였다. [4]의 결합 모델과 비교하여, 본 논문에서 제안하는 결합 모델을 사용할 경우 띄어쓰기와 품사 태깅 성능이 각각 0.47과 0.21 만큼 향상 되었다.

표 1. 한국어 띄어쓰기 성능 (음절단위 정확도) 및 품사 태깅 성능 (형태소 단위 F1-measure)

모델	띄어쓰기	품사태깅
입력 문장의 띄어쓰기 정확도가 100%인 경우		
POS tagging using S-SVM [4]	-	98.03
입력 문장의 띄어쓰기를 제거한 경우		
Word spacing using S-SVM + POS tagging using S-SVM (pipeline) [4]	-	96.77
Joint model of word spacing and POS tagging using S-SVM [4]	96.86	96.99
Jointly Learning Model: 띄어쓰기 50만 문장 추가	**97.17**	**97.10**
Jointly Learning model: 띄어쓰기 180만 문장 추가	**97.33**	**97.20**

5. 결론

본 논문에서는 파이프라인 방식의 문제를 해결하기 위해 사용되는 일반적인 결합 학습 방법을 확장하여, 두 작업이 동시에 태깅된 학습 데이터뿐만 아니라 한 작업만 태깅된 학습데이터도 동시에 학습에 사용할 수 있는 결합 학습 모델을 Latent Structural SVM을 확장하여 제안하였다. 실험 결과, 기존의 한국어 띄어쓰기와 품사 태깅 결합 모델의 품사 태깅 성능이 96.99%였으나, 본 논문에서 제안하는 결합 학습 모델을 이용하여 대용량의 한국어 띄어쓰기 학습데이터를 추가로 학습한 결과 품사 태깅 성능이 97.20%까지 향상 되었다.

참고문헌

[1] Yanxin Shi and Mengqiu Wang. A dual-layer CRF based joint decoding method for cascade segmentation and labelling tasks. In Proceedings of the IJCAI Conference, Hyderabad, India, 2007.

[2] Hwee Tou Ng and Jin Kiat Low. Chinese part-of-speech tagging: One-at-a-time or all-at-once? Word-based or character-based? In Proceedings of the EMNLP Conference, Barcelona, Spain, 2004.

[3] Jenny Rose Finkel and Christopher D. Manning. Joint parsing and named entity recognition. In Proceedings of the NAACL Conference, 2009.

[4] 이창기. Structural SVM을 이용한 한국어 띄어쓰기 및 품사 태깅 결합 모델. KCC, 2013.

[5] Joachims T., et al. Cutting-Plane Training of Structural SVMs. Machine Learning, vol. 77, no. 1, 2009.

[6] Yu C. and Joachims T., Learning Structural SVMs with Latent Variables. In Proceedings of the ICML, 2009.

[7] Shalev-Shwartz S., et al. Pegasos: Primal estimated sub-gradient solver for SVM. Mathematical Programming, 127, 1, 2011.

제25회 한글 및 한국어 정보처리 학술대회
25th Annual Conference on Human and Cognitive Language Technology

구두발표 2B

토픽 모델을 이용한 수학식 검색 결과 재랭킹

양선○, 고영중
동아대학교
seony.yang@gmail.com, youngjoong.ko@gmail.com

Reranking Search Results for Mathematical Equation Retrieval Using Topic Models

Seon Yang○, Youngjoong Ko
Dong-A University

요 약

본 논문은 두 가지 주제에 대해 연구한다. 첫 번째는 수학식 검색에 대한 것이다. 웹에는 양질의 수학식 데이터가 마크업 언어 형태로 저장되어 있으며 이를 활용하기 위한 연구들이 활발히 진행되고 있다. 본 연구에서는 MathML (Mathematical Markup Language)로 저장된 수학식 데이터를 일반 질의어를 이용하여 검색한다. 두 번째 주제는 토픽 모델(topic model)로 검색 성능을 향상시키는 방법에 대한 것이다. 먼저 수학식 데이터를 일반 자연어 문장으로 변환한 후 Indri 시스템을 이용하여 검색을 수행하고, 토픽 모델을 이용하여 미리 산출된 스코어를 적용하여 검색 순위를 재랭킹한 결과, MRR 기준 평균 5%의 성능을 향상시킬 수 있었다.

주제어: 수학식 검색, MathML, 토픽 모델, Indri, 재랭킹

1. 서론

웹에 계속 증가되고 있는 대용량 데이터를 어떻게 활용할 것인지에 대한 주제는 계속해서 연구의 대상이 되어 왔다. 특히 텍스트 형태로 저장된 웹 문서들을 의사결정 (decision making), 마케팅 전략 구상 등의 분야에서 효과적으로 사용하기 위한 연구들이 활발히 진행되고 있는데, 감정 분석(sentiment analysis), 의견 마이닝(opinion mining), 상품 리뷰 분석(product review analysis) 등이 이에 속하며 양질의 연구 결과가 계속해서 보고되고 있다.

이와 같이 일반 텍스트 문서에 대한 연구가 활발히 진행되고 있는 반면, 웹에 저장되어 있는 풍부한 수학식 데이터 활용에 대한 연구는 전 세계적으로 아직 초기 단계에 있다고 볼 수 있다. 이미지 형식으로 수식이 저장되던 과거와는 달리 MathML (Mathematical Markup Language)[1] 등의 마크업 언어가 발표되면서, 이러한 형태로 표현된 수학식을 포함하는 웹 문서의 수는 계속해서 급증하고 있다. 그런데, 웹에서의 수학식 저장 방법이 용이해진 데에 반해 그 데이터의 검색 및 활용에 대한 연구는 이제 시작 단계라고 볼 수 있다.

본 논문은 다음과 같이 두 가지 주제에 대하여 연구를 진행한다.

1) 일반 자연어 질의를 이용한 MathML 수학식 검색
2) 토픽 모델(topic model)을 이용한 검색 성능 향상

첫 번째 주제는 수학식 검색에 대한 것으로, MathML에 대해 전혀 모르는 사용자들도 수학식을 검색할 수 있도록 일반 자연어 질의어를 이용하는 데에 중점을 두며, 이를 위해 각각의 수학식 데이터를 한글 문장으로 변환하는 방법을 사용한다. 이 때 하나의 수학식에 대해 일반 문장으로는 표현하는 방법이 여러 가지가 있을 수 있기 때문에, 수학 기호 표현에 대한 동의어 사전의 구축을 통해 질의어로 들어오는 다양한 표현을 정확히 인식하는 데 주력한다. 그 후 Indri 검색 시스템[2]을 이용하여 인덱싱 및 검색을 수행한다.

두 번째 주제는 토픽 모델을 이용하여 검색 성능을 향상시키는 방법에 대한 연구이다. 토픽 모델을 적용함으로써 비감독 학습(unsupervised learning)에 의해 수학식 데이터를 적정 수의 토픽수로 미리 클러스터링(clustering)하며, 이 때 산출된 두 가지 스코어인 문서VS토픽 (여기서 문서는 수학식을 의미함) 스코어 및 단어VS토픽 스코어를 이용하여 Indri 검색 결과를 재랭킹(reranking)하는 방법을 이용한다.

본 논문의 구성은 다음과 같다. 2장에서 MathML 수학식 관련 연구 및 토픽 모델을 정보 검색에 이용하는 연구를 간단히 소개한다. 3장은 수학식 데이터를 한글로 변환하는 과정을 설명하고, 4장에서는 토픽 모델을 이용하여 검색 결과를 재랭킹하는 방법에 대해 설명한다. 5장에서 실험 및 결과를 기술하며, 6장에서 결론 및 향후 연구에 대해 논한다.

2. 관련 연구

MathML이 W3C 에서 제안되면서[1] 웹상의 수학식 표현

이 논문은 2013년 정부(교육과학기술부)의 재원으로 한국연구재단의 지원을 받아 수행된 기초연구사업임 (No. NRF-2013R1A1A2009937)

에 대한 연구가 활발해졌다. Ferreira외[3]는 MathML 수학식을 영어 및 포르투갈어로 변환하는 시스템에 대해 연구하여 AudioMath[4]를 개발하였으며, DesignScience에서는 MathML 수식을 영어로 읽어주는 MathPlayer[5]가 발표되었다. Kim외[6]는 MathML 수식을 여러 클래스로 분류하는 연구를 진행하였다.

MathML이 많이 사용되면서 웹에 있는 MathML 수학식 검색에 대한 연구도 시작되었다. Adeel외[7]는 MathML로 구성된 수학식에 대해 정규 표현을 사용하여 색인어를 생성하였으며, Misutka외[8]는 후위표기를 통해 색인어를 추출하였다. 이준영외[9]는 한글로 입력된 질의어를 MathML 표현으로 변환한 뒤 MathML 수학식을 검색하는 시스템을 제안하였다.

문서 검색에 토픽 모델을 이용하는 연구들도 지속적으로 발표되고 있다. Yi외[10]는 여러 토픽 모델 기법들을 이용한 검색 비교 실험을 수행하였으며, Deveaud외[11]는 정보검색에 토픽 모델 수행을 검색 도중에 수행하는 방법을 제안하였다.

3. 일반 질의어를 이용한 수학식 검색

서론에서 기술하였듯이 본 수학식 검색의 전제는 사용자들이 일반 질의어를 사용하여 웹에 있는 MathML 수학식을 검색할 수 있어야 한다는 점이다. 즉, 수학식을 소리 내어 읽듯이 검색어를 입력함으로써 복잡한 수학 기호를 입력하거나 별도의 수식 입력 툴 사용 없이도 해당 수식을 검색할 수 있도록 한다. 이를 위해 먼저 다양한 수학식 평문 표현을 관찰하고 MathPlayer[5]를 참고로 하여 수학식을 한글 문장으로 변환하는 시스템을 구축하였다. 아래는 그 예를 보여준다.

• 수학식: $b^2 = a^2 + c^2 - 2ac \bullet \cos B$
• MathML 표현:

```
<math display='block'>
  <mrow>
    <msup>
       <mi>b</mi><mn>2</mn>
    </msup>
    <mo>=</mo>
    <msup>
       <mi>a</mi><mn>2</mn>
    </msup>
    <mo>+</mo>
    <msup>
       <mi>c</mi><mn>2</mn>
    </msup>
    <mo>-</mo>
    <mn>2</mn><mi>a</mi><mi>c</mi>
    <mo>&#x22C5;</mo>
    <mi>cos</mi><mi>B</mi>
  </mrow>
</math>
```

• 한글 문장으로 변환된 수학식: “b의 이제곱 이퀄 a의 이제곱 플러스 c의 이제곱 마이너스 2 a c 코싸인 B”

이 때 주의할 점은 하나의 수학식에 대해 다양한 자연어 표현이 있을 수 있다는 점이다. 예를 들어 b^2를 평문으로 작성하면 다음과 같이 다양한 표현이 나올 수 있다.

• “b제곱”, “b의 제곱”, “b의 제곱”, “비의 이제곱”, “비의 2승”, ...

이와 같이 매우 단순한 수학표현에서도 다양한 가짓수의 평문 표현을 관찰할 수 있다. 즉, b를 영어 알파벳 “b”로 표현할 수도 있고, 한글 “비”로 표현할 수도 있으며, 숫자 2에 대해서도 그냥 숫자 자체로 표현하는 경우와 한글로 “이”라고 표현하는 경우를 발견할 수 있었으며, “위첨자 2”에 대해서 “제곱”, “이제곱”, “2제곱”, “2승” 등 다양한 표현이 나왔다. 또한 b 뒤에 “의”라는 조사는 부착된 경우도 생략된 경우도 관찰할 수 있었다. 이런 많은 수학식 평문 표현을 관찰하여 수학 기호에 대한 MathML 표현과 다양한 평문 표현을 연결시킨 수학 기호 표현에 대한 동의어 사전[표 1]을 구축하였다.

표 1. 수학 기호 표현 동의어 사전의 예

구분	수학 기호	MathML 표현	다양한 평문 표현
식별자	a	<mi>a</mi>	a, 에이
	x	<mi>x</mi>	x, 엑스, 엑쓰
	sine	<mi>sin</mi>	sine, sin, 사인, 싸인
	log	<mi>log</mi>	log, 로그
	Γ	<mi>Γ</mi>	gamma, 감마
연산자	+	<mo>+</mo>	+, plus, 더하기, 플러스, 프러스, 쁠, ...
	=	<mo>=</mo>	=, equal, 는, 은, 이퀄, 같다, ...
	∫	<mo>∫</mo>	integral, 인테그랄, 인테그럴, 적분, ...
	∩	<mo>∅</mo>	intersection, cap, 교집합, 교, ...
	Σ	<mo>∑</mo>	sigma, 시그마, 씨그마, ...
숫자	2	<mn>2</mn>	2, 이, two, 투, ...

위의 사전을 이용함으로써 아래의 예처럼 다양한 평문 질의어를 웹에 저장된 수학식 데이터와 최대한 연결시킬 수 있도록 표준화 한다.

• 질의어: “root b제곱 - 4ac”,
“루뜨 비제곱 빼기 4 곱하기 에이 시”,
“루트 b 2승 마이너스 4ac”,

...

• 표준화 이후: “루트 b의 이제곱 마이너스 4 a c”

위의 예처럼 하나의 기호에 대해 여러 표현이 가능한 반면, 여러 기호가 동일하게 표현될 수도 있다. 예를 들어 “R”, “r”, “ℜ”, “ℝ”은 MathML 표현으로는 각각 <mi>R</mi>, <mi>r</mi>, <mi>葶</mi>, <mi>葷</mi>에 해당하지만, 일반 질의어로 입력될 될 때는 네 가지 경우 모두 한글로는 “알” 혹은 “얼”, 그리고 영어 알파벳으로는 “r” 혹은 “R” 로 입력될 것이다. 이와 같이 동일한 읽기로 표현되는 MathML 수식을 검색하기 위해 모든 가능한 MathML 표현 값을 추가하여 질의어를 확장한다.

4. 토픽 모델을 이용한 검색 성능 향상

한글 문장으로 변환된 수학식을 대상으로 Indri 검색 시스템을 이용하여 수학식 검색을 수행한다. 이 때 검색 성능 향상을 위해서 토픽 모델을 이용한다. 한글로 변환된 수학식 데이터에 대해 미리 LDA (Latent Dirichelet Allocation)에 의한 토픽 모델링을 수행하여 수학식VS토픽, 단어VS토픽 스코어를 산출하며, 이 스코어를 이용하여 수식 (1)과 같은 Indri 검색 결과를 수식 (2)와 같이 재랭킹하였다. 아래 수식에서 w는 수학식 d에 포함된 질의어를 의미하며, z는 토픽을 가리킨다.

• Indri 랭킹 스코어 : $\sum_{w} p(w|d)$ (1)

• 재랭킹을 위한 스코어: $\sum_{w}\sum_{z} p(w|z)p(z|d)$ (2)

실제 재랭킹 시에는 위 두 수식을 선형 결합한다. 실험 결과 토픽 수가 2와 3일 때 최종 검색 성능이 향상됨을 확인할 수 있었는데, [표 2] 및 [표 3]은 토픽 수를 각각 2 및 3으로 했을 때 토픽별 스코어가 상위에 랭크된 단어 리스트이다. 여러 토픽에서 고른 스코어를 보여주거나 숫자인 경우를 제외하고, 해당 토픽에 편중 현상을 보이는 상위 20개 단어들이다.

[표 2]의 단어들을 보면, 토픽1 경우 분수 관련한 용어들이 많이 등장하며, 벡터, 행렬, 로그, 미분, 적분 관련된 단어들이 상위 스코어를 가지고 있음을 볼 수 있다. 토픽2에는 함수, 삼각함수, 방정식에 관련된 단어들이 상위에 많이 포함되었다. 그리고 [표 3]의 단어들을 보면, 토픽1 경우 분수 관련한 용어들이 많이 등장하며, 삼각함수, 로그 관련된 단어들이 상위 스코어를 가지고 있음을 볼 수 있다. 토픽2에는 벡터, 행렬, 명제에 관련된 단어들이 상위에 포함되었으며, 토픽3 경우 함수, 미분 적분, 집합 관련 단어들이 많이 포함되었음을 볼 수 있다.

이 결과는 토픽이 2, 3인 경우 토픽 모델링을 이용하여 수학식 데이터를 수학 영역(혹은 단원)별로 분류하고 있음을 알 수 있다. 그리고 이러한 분류 정보가 단어의 가중치를 결정 할 때 최종 성능을 향상시킬 수 있는 수단으로 사용될 수 있음을 실험을 통해 확인할 수 있다.

표 2. 토픽 수 2일 때의 상위 단어 리스트

토픽1	토픽2
분수	괄호열고
끝	괄호닫고
아래첨자	x
분에	이제곱
n	f
y	A
벡터	위첨자
d	바
표	x의
0열	함수
k	B
부터	루트
로그	쎄타
시작	P
0행	싸인
1행	코싸인
C	델타
t	알파
프라임	g
인테그랄	오른쪽이중화살표

표 3. 토픽 수 3일 때의 상위 단어 리스트

토픽1	토픽2	토픽3
분수	아래첨자	f
이제곱	n	A
분에	벡터	d
바	표	위첨자
x의	0열	까지
루트	쎄타	함수
P	c	B
로그	k	부터
삼제곱	1열	t
알파	시작	인테그랄
분모	0행	델타
분자	1행	리미트
파이	프라임	교집합
베타	p	합집합
작거나같다	오른쪽이중화살표	시그마
크거나같다	r	크다
탄젠트	양쪽이중화살표	작다
싸인	q	C
코싸인	가로생략	F
사제곱	콜론	무한대

반면 토픽 수가 4 이상인 경우 최종 검색 성능 향상을 확인할 수 없었는데, 단어의 스코어가 한 토픽에 편중되기 보다는 둘 이상의 토픽에서 고른 스코어를 보여주는 경우가 많았고, 토픽 수가 많아질수록 편중 현상은 더 줄어들고 각 토픽별로 상위에 랭크된 단어리스트에서 특정 수학 영역 발견에 모호함이 많았다. 이는 실험에 사용된 데이터 수가 1,811개로 상대적으로 작아서 토픽 수가 많은 경우 명확한 분류가 어려웠을 것으로 판단된다. 또한 수학식 데이터는 일반 문서보다 길이가 짧고, 사용되는 단어의 집합도 일반 문서들보다 그 원소 수가 작다는 것도, 토픽 수가 많은 경우 분류가 모호해지는 원인이 된 것으로 판단된다.

5. 실험

5.1 실험 데이터

본 실험 데이터로는 고등학교 과정 수학 교재의 수학식을 MathML로 변환한 데이터를 사용하였으며, 특정 단원에 편향되지 않게 로그, 미분, 벡터, 행렬 등 다양한 단원으로부터 균등한 비율로 수학식을 선택하였다. 는 실험 데이터를 요약한 것이다. 시스템 평가 도구로는 일반 검색에서도 많이 사용되는 MRR(Mean Reciprocal Rank)을 사용하였으며, 먼저 전체 수학식 데이터 중 임의의 200개 수학식을 사용자에게 보여준 후 평문으로 자유롭게 질의어로 입력하도록 하였고, 검색 결과에 대해 MRR을 산출하였다. [표 4]는 실험 데이터 수를 요약해서 보여주고 있다.

표 4. 실험 데이터 현황

내용	수
수학식 수	1,811개
중복을 제외한 단어 개수	690개
테스트 질의 수	200개

5.2 검색 성능 평가

먼저 토픽 모델 적용 전 Indri를 이용한 수학식 검색 결과는 [표 5]와 같다. 가중치 방법은 Indri에서 기본적으로 제공하는 language model (LM)과, 추가로 Tfidf 및 Okapi를 사용하였다.

표 5. Indri 검색 결과

실험 방법	MRR
LM	0.2301
Tfidf	0.3432
Okapi	0.3632

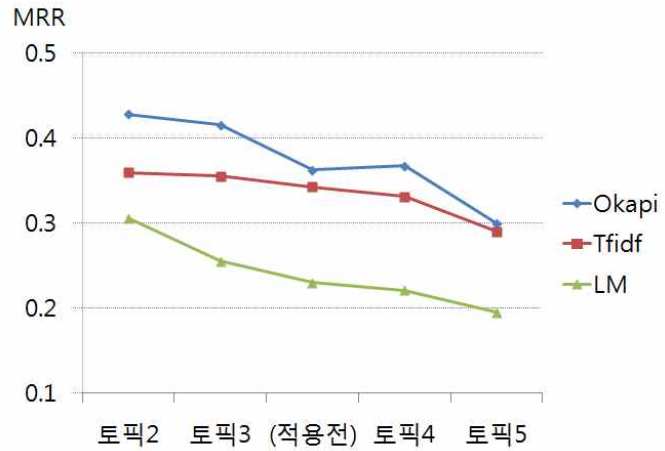
[표 4]를 보면 일반 문서 검색과 달리 LM을 이용하였을 때 성능이 낮았는데, 문서의 길이(즉, 한글로 변환된 각 수학식의 길이)가 일반 문서보다 훨씬 짧으며, LM이 러한 문서길이에 가장 민감하게 영향을 받았기 때문으로 판단된다.

그리고 토픽 모델링 결과를 반영한 후의 향상된 결과는 [표 6]과 같다.

표 6. 토픽 모델을 이용한 재검색 결과 (괄호 안은 토픽 모델 적용 이전 대비 상향된 비율을 나타냄)

실험 방법	MRR (2 토픽)	MRR (3 토픽)
LM	0.3060 (7%)	0.2556 (2%)
Tfidf	0.3602 (2%)	0.3556 (1%)
Okapi	0.4288 (6%)	0.4162 (5%)

[표 6]에서 볼 수 있듯이 2 토픽 경우 평균 5%의 성능 향상을 보였으며, 3 토픽 경우에도 3%에 가까운 성능 향상을 확인할 수 있었다. 요약하자면, 토픽 모델 결과를 이용하여 수학식 검색 순위를 재랭킹함으로써 평균 검색 성능을 성공적으로 향상시킬 수 있음을 알 수 있다.



[그림 1] 토픽 수별 성능 변화

그리고 [그림 1]에서 볼 수 있듯이 토픽 수가 4 이상인 경우는 성능 향상을 얻을 수 없었는데, 그 이유는 앞에서 설명하였듯이 실험에 사용된 수학식 수, 수학식의 길이, 그리고 사용된 단어 수 등에 영향을 받은 것으로 보인다.

6. 결론

본 연구는 수학식 검색 및 토픽 모델 적용이라는 두 가지 주제에 대하여 실험을 수행하였다. 먼저 일반 질의어를 이용하여 수학식을 검색하기 위해 수학식 데이터를 자연어 문장으로 변환하였으며 Indri 시스템을 이용하여 검색을 수행하였다. 또한 검색 성능 향상을 위해 토픽 모델을 적용하였는데, Indri 검색 순위를 재랭킹 하는데 있어서 토픽 모델에서 산출된 스코어를 이용하였다. 이와 같은 재랭킹 결과 토픽 모델 적용 전에 비해서 토픽 수 2인 경우 평균 5%의 성능향상을 확인할 수 있었다.

향후 연구로는 수학식 검색 시스템을 영어 등 다른 언

어에 대해 적용하는 방법에 대해서 연구할 예정이다. 또한 수학식을 한글로 변환한 데이터와 MathML 수학식 자체를 둘 다 사용하여 검색을 수행하는 실험을 계획중이며, 토픽 모델을 이용하여 단어간 유사도를 산출 후 검색 가중치에 결합시키는 실험을 계획 중이다.

참고문헌

[1] http://www.w3.org/Math
[2] http://www.lemurproject.org/indri.php
[3] Helder Ferreira and Diamantino Freitas, "AudioMath: Towards Automatic Readings of Mathematical Expressions," Proceedings of Human-Computer Interaction International, 2005.
[4] http://lpf-esi.fe.up.pt/~audiomath
[5] http://www.dessci.com/en/products /mathplayer
[6] Shinil Kim, Seon Yang, Youngjoong Ko, "Classifying Mathematical Expressions Written in MathML," Journal of IEICE Transactions on Information and Systems, vol.E95-D, no.10, pp.2560-2563, 2012.
[7] Muhammad Adeel, Hui Siu Cheung, Sikandar Hayat Khiyal, "Math GO! Prototype of A Content Based Mathematical Formula Search Engine," Journal of Theoretical and Applied Information Technology, vol.4, no.10, pp.1002-1012, 2008.
[8] Jozef Misutka, Leo Galambos, "Extending Full Text Search Engine for Mathematical Content," Proceedings of Towards Digital Mathematics Library: DML 2008 workshop, pp.55-67, 2008.
[9] 이준영, 양선, 고영중, "평문 질의어 MathML 용어 변환을 이용한 수학식 검색", 한국컴퓨터종합학술대회 논문집, 39권, 1(B)호, pp.312-314, 2012.
[10] Xing Yi, James Allan, "A Comparative Study of Utilizing Topic Models for Information Retrieval," Proceedings of European Conference on Information Retrieval (ECIR 2009), pp.29-41, 2009.
[11] Romain Deveaud, Eric SanJuan, Patrice Bellot, "Are Semantically Coherent Topic Models Useful for Ad Hoc Information Retrieval?," Proceedings of Association for Computational Linguistics (ACL Short Papers 2013), 2013.

개체명 인식을 위한 개체명 사전 자동 구축

전원표○, 송영길, 최맹식, 김학수
강원대학교, IT대학, 컴퓨터정보통신공학전공
{nlpjwp, nlpyksong, nlpmschoi, nlpdrkim}@kangwon.ac.kr

Automatic Construction of a Named Entity Dictionary for Named Entity Recognition

Wonpyo Jeon○, Yeongkil Song, Maengsik Choi, Harksoo Kim
Program of Computer and Communication Engineering,
College of Information Technology,
Kangwon National University

요 약

개체명 인식기에 대한 연구에서 개체명 사전은 필수적으로 필요하다. 그러나 공개된 개체명 사전은 거의 없기 때문에, 본 논문에서는 디비피디아의 데이터로부터 개체명을 효과적으로 추출하여 자동으로 구축할 수 있는 방법을 제안한다. 제안 방법은 엔트리의 '이름'과 '분류' 정보를 사용한다. 엔트리의 '이름'은 개체명으로 사용하고, 엔트리의 '분류'는 각 개체명 클래스와의 상호정보량을 계산하여 엔트리와 개체명 클래스 사이의 점수를 계산한다. 이렇게 계산된 점수를 이용하여 개체명과 개체명 클래스를 매핑한다. 그 결과 76.7%의 평균 정확률을 보였다.

주제어: 개체명 사전, 디비피디아, 상호정보량

1. 서론

개체명 인식은 자연어에서 사용된 인명, 지명, 조직명 등의 정보를 식별하여 해당 클래스로 분류하는 작업을 말한다. 개체명은 대부분의 문서에서 중요한 핵심어 역할을 하기 때문에 자연어처리 연구에 다양하게 응용된다. 특히 도메인에 독립적인 질의응답 시스템(Open Domain Question Answering)에서는 많은 종류의 개체명을 인식할 수 있는 개체명 인식이 필수적이다[1]. 이러한 개체명 인식을 위해서는 개체명 사전이 필요하다. 그러나 개체명 사전을 구축하기 위해서는 많은 인력과 시간이 소비된다. 또한 어렵게 개체명 사전을 구축하더라도 지속적으로 생성되는 개체들을 추가하고, 관리하는 것은 현실적으로 많은 어려움이 있다. 따라서 본 논문은 디비피디아(DBpedia) 데이터를 이용하여 개체명을 효과적으로 추출하여 자동으로 구축할 수 있는 방법을 제안한다.

2. 관련 연구

개체명 사전을 구축하기 위해서는 개체명 클래스 분류를 정의해야 한다. 개체명 클래스의 분류는 연구 분야에 따라 서로 다른 정의를 사용하고 있으며, 그에 대한 많은 연구가 있었다. MUC-6(Message Understanding Conference)에서는 'PERSON', 'LOCATION', 'ORGANIZATION'의 3개 분류를 사용하고 있으며, 이것은 여러 연구에서 대분류 개체명 인식의 기본 클래스로 사용되고 있다[2]. BBN(Bolt Beranek and Newman)은 QA(Question Answering)를 위한 가이드 라인에서 29개의 클래스 계층 구조를 정의하였다[3]. 또한 Sekine는 박물관, 강 등 많은 세밀한 하위 클래스를 포함하고 있는 개체명 클래스 계층 구조를 정의했다[4]. Tkachenko는 위키피디아(WikiPedia) 데이터를 사용한 개체명 인식을 위해 BNN과 Sekine의 클래스 계층 중에서 15개의 메인 클래스와 3개의 보조 클래스를 정의하여 사용했다[5].

본 논문의 내용과 유사한 연구로 배상준[6]의 연구가 있다. 배상준[6]은 위키피디아 데이터를 이용하여 개체명 사전을 자동 구축하는 연구를 하였다. 위키피디아 엔트리(entry) 내에 있는 '분류' 정보를 이용하여 개체명 클래스의 분류체계를 구성하고, 엔트리를 매핑(mapping)시키는 방법이다. 이때, 양질의 분류체계를 얻기 위해 불확실성 측정기법을 사용하여 불확실성이 높은 분류체계를 제거하는 방법을 사용하였다. 그러나 이 방법은 새로운 개체명 분류체계를 만들어 개체명 사전을 만드는 방법이기 때문에 기존의 개체명 분류를 이용하는 연구에는 그래도 사용하기에 어려움이 있다. 그래서 본 논문에서는 정의된 개체명 분류체계에 개체명을 효과적으로 매핑하는 방법을 제안한다.

3. 개체명 인식기

본 연구는 개체명 인식기 연구의 선행 연구로 수행하

*이 논문은 2013년도 정부(교육과학기술부)의 재원으로 한국연구재단의 지원을 받아 수행된 기초연구사업임(2013R1A1A4A01005074) 또한 본 논문은 지식경제부 및 한국산업기술평가관리원의 산업융합원천기술개발사업(정보통신)의 일환으로 수행하였음. [10041678, 다중영역 정보서비스를 위한 대화형 개인 비서 소프트웨어 원천 기술 개발]

였다. [그림 1]은 본 연구팀에서 연구하고 있는 개체명 인식기의 구조를 보여준다.

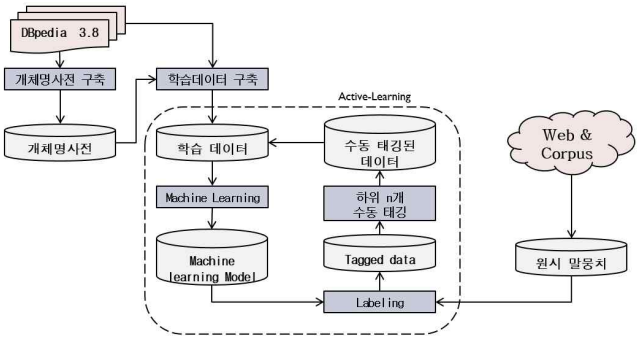


그림 1. 개체명 인식기 구조도

기계학습 기반의 개체명 인식기의 연구를 위해서는 개체명이 태깅된 대용량의 말뭉치가 필요하다. 그러나 국내에는 공개된 개체명 태깅 말뭉치가 거의 없기 때문에, 직접 말뭉치를 구축해야 한다. 말뭉치 구축을 위해 수작업으로 개체명 태깅을 하는데 시간적 인적 자원의 소모가 크므로 개체명 사전이 필수적이다.

4. 개체명 사전 자동 구축

4.1. 개체명 클래스 정의

<표 1>은 본 논문에서 정의한 개체명 클래스이다.

표1. 개체명 클래스 정의

번호	Tkachenko	본 연구	비고
1	PERSON	PERSON	
2	GEOLOGICAL_REGION	LOCATION	의미 확장
3	FACILITY	FACILITY	
4	AUTO	VEHICLE	의미 확장
5	GAME	GAME	
6	PLANT	PLANT	
7	PERIOD	PERIOD	
8	COMPUTER	COMPUTER	
9	DISAMBIGUATION_PAGE	-	삭제
10	GPE	GPE	
11	ASTRAL_BODY	-	삭제
12	ORGANIZATION	ORGANIZATION	
13	WORK_OF_ART	WORK_OF_ART	
14	ANIMAL	ANIMAL	
15	INSECT	INSECT	
16	LANGUAGE	LANGUAGE	
17	LIST_OF_PAGE	-	삭제
18	OTHER_PAGE	OTHER	이름 변경
19	-	EVENT	추가
20	-	THEORY	추가
21	-	FOOD	추가

<표 1>의 개체명 클래스는 Tkachenko가 정의한 개체명 클래스 일부를 수정하여 정의한 것으로, 향후 진행할 개체 관계추출 연구를 위해 기존의 개체명 클래스에서 추가 및 삭제, 수정을 하였다. <표 1>의 의미 확장은 기존 개체명 클래스의 한정적인 의미 때문에 디비피디아의 엔트리를 커버하기 어려운 문제를 보완하기 위한 것이다. 예를 들어 'AUTO'의 경우 자동차를 의미하는데 'VEHICLE'로 확장함으로써 탈 것에 대한 엔트리를 커버할 수 있다.

4.2. 개체명 사전 자동 구축

개체명 사전 자동 구축을 위해 디비피디아 데이터를 사용하였다. 디비피디아는 위키피디아의 데이터를 RDF(Resource Description Framework) 형태의 파일로 저장한 것으로 위키피디아의 거의 대부분의 데이터를 저장하고 있다. 본 논문에서는 개체명 사전 자동 구축을 위해 디비피디아의 데이터 중 엔트리의 '제목'과 '분류'를 사용하였다.

[그림 2]는 개체명 사전 자동 구축의 과정을 보여준다.

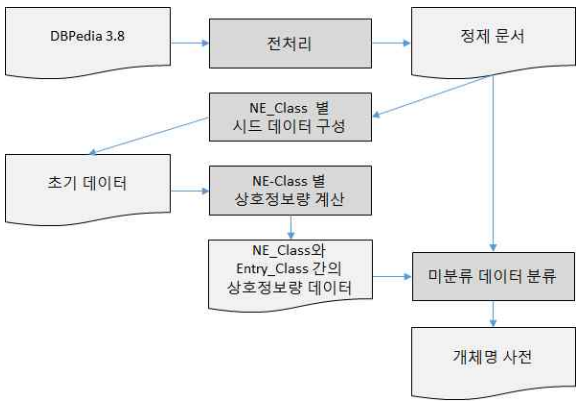


그림 2. 개체명 사전 자동 구축 과정

디비피디아의 데이터로부터 엔트리의 '제목'과 '분류'를 가져온다. 이때 '분류'에 대하여 전처리를 수행한다. '분류'는 여러 어절로 구성될 수 있는데, 본 논문에서는 '분류'의 마지막 어절만을 가져왔다. 이와 같은 방법으로 얻어진 정제 문서로부터 일정 분량의 엔트리를 각각의 개체명 분류로 수작업 분류하여 시드 데이터를 구성한다. 본 논문에서는 각 클래스별로 30개의 엔트리를 시드 데이터로 사용하였다.

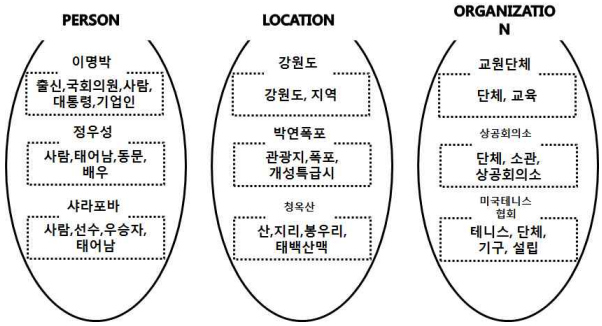


그림 3. 시드 데이터 예

[그림 3]과 같은 초기 데이터로부터 수식 (1)을 이용하여 개체명 클래스와 각 엔트리의 '분류' 정보간의 상호정보량을 계산한다.

$$I(t_i;C_j)=p(t_i,C_j)\log\frac{p(t_i\wedge C_j)}{p(t_i)\bullet p(C_j)} \quad (1)$$

수식 (1)에서 t_i는 엔트리의 '분류'로 [그림 3]에서 점선으로 표현된 "출신", "사람" 등을 의미한다. C_j는 "PERSION", "LOCATION" 등과 같은 개체명 클래스를 의미한다. 이렇게 계산된 값과 수식 (2)를 이용하여 초기 데이터 이외의 엔트리별 점수를 계산하여 개체명 클래스를 매핑한다.

$$Score(E)=\arg max_j\sum_i I(t_i;C_j) \quad (2)$$

이 방법은 엔트리를 구성하는 모든 분류들의 점수를 합산하여 계산하므로 특정 분류 때문에 잘못 분류되는 경우를 처리할 수 있다. 예를 들어 "미국 테니스 협회"의 분류 정보 중 "테니스"가 "ORGANIZATION"이 아닌 다른 클래스와 높은 상호정보량 값을 가지더라도 나머지 분류 정보 "단체", "기구", "설립" 때문에 최종적으로 "ORGANIZATION" 클래스에서 높은 점수가 나온다.

5. 구축 결과

본 논문에서는 개체명 사전 구축을 위해 한국어 디비피디아를 이용하였다. 한국어 디비피디아의 엔트리 중 '분류' 정보를 가지고 있는 엔트리는 186,520개이다. 이중 개체명 사전에 추가할 대상은 단일 '분류' 정보를 가지고 있는 엔트리는 56,031개를 제외한 130,489개로 제한한다. 그 이유는 단일 '분류' 정보를 가지고 있는 엔트리는 각 개체명 클래스와 상호정보량 점수를 구할 수 없기 때문이다. <표 3>은 각 클래스 별로 매핑된 개체명의 수이다.

표 3. 매핑된 개체명의 수

클래스	매핑된 수	클래스	매핑된 수
GPE	8,642	ANIMAL	1,524
PERSON	41,066	INSECT	443
LOCATION	13,924	PLANT	854
WORK_OF_ART	8,730	FOOD	1,445
ORGANIZATION	11,986	FACILITY	10,728
VEHICLE	1,328	LANGUAGE	1,172
THEORY	3,688	COMPUTER	1,166
GAME	3,522	PERIOD	36
EVENT	5,457	OTHER	14,779

구축된 개체명 사전의 신뢰도를 측정하기 위해서는 정답 비교를 위한 개체명 사전이 필요하다. 그러나 공개된 개체명 사전을 찾지 못하였기 때문에 정확한 신뢰도를 측정할 수 없었다. 그래서 차선책으로 분류된 개체명 중 각 클래스 별로 임의로 100개씩 샘플링(sampling)하여 정확률을 측정하였고, 총 5회 반복하여 평균 정확률을 계산하였다. 태깅된 정답이 없기 때문에 재현율은 계산하지 못했다. 매핑된 개체가 36개인 'PERIOD' 클래스의 경우 전체를 대상으로 측정한다. <표 4>는 클래스별로 각 샘플링한 개체명의 평균 매핑 정확률을 나타낸다.

표 4. 평균 매핑 정확률

개체명 클래스	평균 정확률	개체명 클래스	평균 정확률
GPE	0.615	ANIMAL	0.588
PERSON	0.988	INSECT	0.620
LOCATION	0.778	PLANT	0.840
WORK_OF_ART	0.840	FOOD	0.715
ORGANIZATION	0.808	FACILITY	0.858
VEHICLE	0.890	LANGUAGE	0.725
THEORY	0.855	COMPUTER	0.790
GAME	0.695	PERIOD	0.861
EVENT	0.573	OTHER	-

실험 결과 가장 높은 정확률을 보인 것은 PERSON으로 98.8%의 정확률을 보였고, 가장 낮은 정확률을 보인 것은 EVENT로 57.3%의 정확률을 보였다. 전체적으로 평균 76.7%의 정확률을 보였다.

6. 결론

본 논문에서는 개체명 인식기 구축에 필수적인 개체명 사전을 자동으로 구축하는 방법을 제안하였다. 개체명 클래스의 분류로 Tkachenko의 개체명 클래스를 수정하여 총 18개의 개체명 클래스 분류를 만들었다. 개체명 사전의 구축 대상으로 위키피디아의 정보를 저장하고 있는 디비피디아의 정보 중 엔트리 '이름'과 '분류' 정보를 이용하였다. 엔트리의 '이름'과 개체명 클래스를 매핑하기 위해 엔트리를 구성하는 '분류'들과 개체명 클래스 간의 상호 정보량의 합을 사용하였다. 그 결과 개체명 클래스 분류에서 평균 76.7%의 정확률을 보였다.

향후 연구로 개체명 사전의 정확률 향상과 더불어 개체명 인식기, 개체간의 관계추출에 대한 연구를 진행할 예정이다.

참고문헌

[1] Satoshi Sekine, "Named entity: history and future", Technical Report, New York University, 2004.

[2] MUC-6, http://www.cs.nyu.edu/cs/faculty/grishman/muc6.html (2013.09.13 확인)

[3] Annotation guidelines for answer types, http://www.ldc.upenn.edu/Catalog/docs/LDC2005T33/BBN-Types-Subtypes.html (2013.09.13. 확인)

[4] Definition of Sekine`s Extended Named Entity, http://nlp.cs.nyu.edu/ene/version6_1_0eng.html (2013.09.13. 확인)
[5] Maksim Tkachenko, Alexander Ulanov and Andrey Simanovsky, “Fine grained classification of named entities in wikipedia”, Technical report, HP Laboratories, 2010.
[6] 배상준, 고영중, “한국어 위키피디아를 이용한 분류체계 생성과 개체명 사전 자동 구축”, 정보과학회논문지(B), 제16권, 제4호, pp. 492-496, 2010.

한국어 의존 파싱을 이용한 트리플 관계 추출

곽수정[O*], 김보겸[†], 이재성[†]
충북대학교 정보산업공학과[*], 디지털정보융합과[†]
crystalk@cbnu.ac.kr, bogyum@cbnu.ac.kr, jasonlee@cbnu.ac.kr

Triplet Extraction using Korean Dependency Parsing Result

Sujeong Kwak[O*], Bogyum Kim[†], Jae Sung Lee[†]
Dept. of Information & Industrial Engineering[*], Dept. of Digital Informatics and Convergence[†],
Chungbuk National University

요 약

자연언어 문서에서 지식 추출은 QA 시스템을 비롯한 여러 분야에서 필수적이다. 트리플은 가장 일반적인 지식 추출 형식으로 문장 내부의 지식 정보를 주어, 서술어, 목적어의 관계로 표현한다. 본 논문에서는 한국어 의존 파서로 문장을 분석하고, 그 결과에서 트리플을 추출하는 방법을 제안했다. 제안된 트리플 추출기는 21개 문장에서 추출된 78개의 트리플 정답 집합과, 64개의 준정답 집합에 대해서 각각 60.75%와 66.67%의 F-measure 성능을 보였다.

주제어: 트리플 추출, 지식 추출, 관계 추출, 의존 구문 분석기

1. 서론

신문이나 블로그, 위키피디아와 같이 자연언어처리에서 활용 가능한 텍스트 문서가 증가하면서 자연언어 문서에서의 지식 추출에 대한 요구가 증가하고 있다[1]. 추출되는 지식 정보를 이용하면 문서에서 중요한 개체와 그들간의 관계를 파악할 수 있기 때문에 지식추출은 QA 시스템을 비롯한 다양한 시스템의 필수적인 요소가 될 수 있다[2]. 트리플은 가장 일반적인 지식 추출 형식으로 문장 안에 나타난 주어, 서술어, 목적어를 표현한 정보이다[3].

본 논문에서는 한국어 문장 또는 문서에 대하여 의존 파싱을 수행하고, 그 결과를 이용해 트리플을 추출하는 방법을 제안하고자 한다. 2장에서는 트리플 추출과 관련된 논문들을 소개하고, 3장에서는 의존 파싱 결과로 트리플을 추출하는 방법을 설명한다. 4장에서는 설명된 방법으로 구축된 트리플 추출기의 성능을 측정 및 평가하고, 5장에서 결론을 맺는다.

2. 관련 연구

자연언어 문서로부터 지식을 자동으로 추출하고 추론하는 것은 대개 한 문장에서 트리플 정보인 주어, 서술어, 목적어 관계를 추출하는 것으로부터 시작한다[3]. [4]에서는 공개된 파서(스탠포드 파서[5], 링크 파서[6] 등)의 파싱 결과로부터 규칙을 이용하여 트리플을 추출하는 방법을 제시하였다. 또한 [7]에서는 문장에 나타난 모든 중요 단어들을 트리플 관계로 조합하고 이를 SVM을 이용하여 유효한 트리플 관계를 판정하고 추출하였다. [8]은 의존 파서를 통한 구문 분석 결과를 이용해 트리플을 추출하는 방법을 설명한다. 특히 트리플에서 한 요소의 범위에 대한 문제, 능동과 수동 구조의 문장에 대한 처리, 병렬 구조의 문장에 대한 처리, 문장 내부의 종속절에 대한 처리 등 파서의 결과를 이용해 트리플을 추출할 때 생길 수 있는 여러 가지 고려사항에 대해 언급하고 있다. [9]에서는 영어의 술어-논항 구조를 분석하고 이를 정규화한 어휘 패턴으로 생성한 후, 스트링 커널 방법으로 트리플 관계를 추출하였다. 특히 [9]에서는 서술어를 38개의 관계로 한정하여 처리하였다. 이렇게 자연언어 문서에서 트리플을 추출하는 연구가 기존에도 존재하지만 모두 영어를 대상으로 하기 때문에 한국어에 바로 적용하기 어렵다.

본 논문에서는 한국어 의존 파싱의 결과를 이용하여 한국어 자연언어 문서에서 트리플을 추출하기 위한 방법을 제안한다.

3. 트리플 추출

3.1 트리플 추출 과정

의존 파서를 이용한 트리플 추출은 [그림 1]과 같은 과정으로 진행된다.

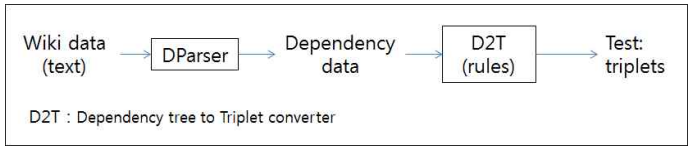


그림 1 트리플 추출 흐름도

먼저 입력 문서가 주어지면 세종 구문 분석 표지[10]를 이용한 ETRI 의존 파서로 문장의 구문 구조를 분석한다. 분석된 결과를 트리플 변환기(D2T)에 입력하면 의존 구조로 나타난 문장의 주어, 서술어, 목적어의 관계가 규칙에 따라 트리플 형식으로 추출된다. 이때 추출되는 주어와 목적어는 조사를 제거하고, 서술어는 기본형으로 표기하

여 저장한다. 이는 트리플을 정규화하여 표현함으로써, 추론 단계에서 트리플 사용을 간편하게 하기 위함이다.

3.2 트리플 추출 규칙

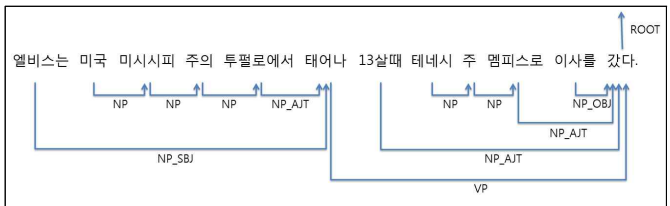
의존 문법에서 주어와 서술어는 각각 SBJ 의존 관계의 tail과 head가 되고 목적어와 서술어는 각각 OBJ 의존 관계의 tail과 head가 된다. 따라서 *_SBJ 레이블과 *_OBJ 레이블이 동일한 서술어를 head로 가지고 있는 경우 *_SBJ의 tail을 주어로, head를 서술어로, *_OBJ의 tail을 목적어로 하는 트리플을 구성할 수 있다. 이때 '*' 는 wild card로 *_SBJ는 NP_SBJ, VP_SBJ 등을 포함한다.

서술어를 설명하는 부사는 서술어와 함께 주어에 대한 정보가 될 수 있기 때문에 *_AJT관계를 *_OBJ 관계와 같은 방식으로 처리한다. 그리고 문장에 동사가 여러 개 있어 주어를 공유한 경우에 동사와 동사는 VP 관계로 레이블되므로 주어가 없는 구에서 VP 관계를 통해 문장 전체의 주어를 찾을 수 있다. 이것을 트리플 추출에 대한 기본 규칙으로 [그림 2]와 같이 정의한다.

1. *_SBJ 레이블과 *_OBJ(혹은 *_AJT) 레이블이 연결된 서술어는 트리플로 묶는다.
2. VP 의존 관계를 통해 백트래킹 후 *_SBJ 의존 관계가 존재하면 해당 명사구를 주어로 사용한다.

그림 2 트리플 추출 기본 규칙

[그림 3]은 트리플 추출의 예로, ①은 기본 규칙 1에 의해 ②,③,④는 기본 규칙 1과 2에 의해 추출된 트리플이다.



추출 트리플

①(엘비스, 태어나다, 투펄로) ③(엘비스, 가다, 멤피스)
②(엘비스, 가다, 이사) ④(엘비스, 가다, 13살)

그림 3 트리플 추출 예

한국어는 주어의 생략이 빈번하게 발생하기 때문에 기본 규칙만 이용해 트리플을 추출하면 문제가 발생한다. 따라서 [그림 4]의 주어 생략 보완 규칙을 정의하여 적용한다.

1. 문장에 주어가 없는 경우 이전 문장의 주어를 사용해 트리플을 구성한다.

그림 4 주어 생략 보완 규칙

4. 실험 및 평가

4.1 평가 집합 구축

트리플 추출기의 성능 평가를 위한 데이터로 엘비스 프레슬리에 대한 위키피디아 문서[11]를 이용했다. 해당 문서는 총 21개의 문장으로 구성되었으며 각각의 문장에 대하여 트리플을 추출하여 총 78개의 정답(Gold answer) 트리플과 64개의 준정답(Silver answer) 트리플 평가 집합을 만들었다.

정답 집합은 원본에서 직접 추출한 트리플 집합으로, 의존 파서를 이용한 트리플 추출기 전체의 성능을 평가할 수 있는 평가 집합이다. 준정답 집합은 정답 집합에서 의존 파서의 오류에 의한 트리플 추출의 오류를 제거한 집합으로, 정답 집합의 82.05%로 구성된 부분 집합이다. 이것을 이용하면 파서의 오류에 의한 트리플 추출기의 성능 저하를 무시하고, 트리플 변환기의 성능만 평가할 수 있다. [그림 5]는 정답과 준정답 그리고 추출기에서 뽑힌 트리플의 차이를 설명한다.

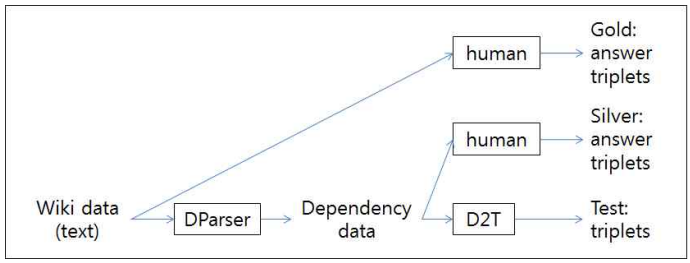


그림 5 정답, 준정답, 출력 결과의 차이점

트리플 정답을 구성할 때, 주어와 목적어의 범위를 정하는 기준이 문제가 될 수 있다. 본 논문에서 제안하는 트리플 추출기는 현재 이 범위 문제를 고려하지 않고 트리플을 추출하고 있으나 추후 확장된 트리플 추출기의 평가를 위해 주어와 목적어의 범위를 "트리플이 의미를 가지게 하는 최소의 범위"로 정의하여 평가 집합을 구성하였다. 예를 들어 "엘비스는 미국 미시시피 주의 투펄로에서 태어났다."라는 문장에서 목적어는 "미국 미시시피 주의 투펄로", "미시시피 주의 투펄로", "투펄로"가 될 수 있다. 그중 "투펄로"는 지명으로 추출하고자 하는 트리플에서 충분히 의미가 되기 때문에, 이를 목적어로 하였다. 다른 예로 "엘비스는 많은 음악 명예의 전당에 올랐다."라는 문장에서 목적어는 "음악 명예의 전당", "명예의 전당", "전당"이 될 수 있는데, "음악 명예의 전당"을 목적어로 하여 트리플을 추출하였다.

4.2 평가

정답 집합과 준정답 집합이 서로 다르기 때문에 트리플 추출기의 성능을 각각에 대해서 측정했다. [표 1]은 정답 집합에 대한 트리플 추출기의 성능이고, [표 2]는 준정답 집합에 대한 트리플 추출기의 성능이다. 이때 기본 규칙만 이용해 추출된 트리플은 38개이고, 기본 규칙과 보완 규칙을 함께 적용해 추출된 트리플은 80개이다.

표 1 정답 집합에 대한 트리플 추출기의 성능

구분	Recall	Precision	F-measure
기본 규칙	35.89%	73.68%	48.26%
기본+보완 규칙	61.53%	60.00%	60.75%

표 2 준정답 집합에 대한 트리플 추출기의 성능

구분	Recall	Precision	F-measure
기본 규칙	43.75%	73.68%	54.90%
기본+보완 규칙	75.00%	60.00%	66.67%

기본 규칙만 이용해 구성한 트리플 추출기는 정답 집합과 준정답 집합 모두에서 정확률이 낮고, 재현율이 높은 것을 확인할 수 있었다. 이것은 기본 규칙만 이용하면 추출되는 트리플의 수가 적기 때문에 나타나는 현상이다. 하지만 주어 생략 보완 규칙을 통해 추가적으로 생성되는 트리플이 정답(혹은 준정답) 집합에 저장되어 있는 경우가 많았다. 따라서 재현율이 떨어지는 정도보다 정확률이 올라가는 정도가 높아 F-measure의 성능이 향상되는 것을 확인할 수 있다.

4.3 트리플 추출기 오류 분석

준정답 집합의 트리플 중 추출되지 못한 트리플은 대부분 병렬 구조 문장의 것이었다. [그림 6]의 예를 보면 “함께”로 연결된 트리플 ‘(엘비스, 하다, 빌 블랙)’과 ‘(엘비스, 하다, 스코티 무어)’를 추출기에서 뽑지 못하였다.

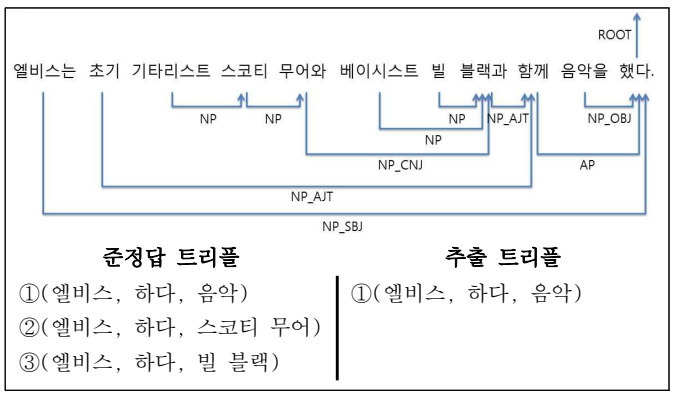


그림 6 병렬 구조 문장 트리플 추출 오류 예

그리고 추출된 트리플 중 정답이나 준정답에 없는 트리플은 의존 파서의 오류를 그대로 트리플로 추출한 것이거나, 주어가 아닌 성분을 수식하는 종속절에서 잘못된 주어를 가지는 트리플이 추출된 경우이다. [그림 7]의 예를 보면 ‘(음반 판매고, 많다, 팝 역사상)’ 트리플이 주어를 제대로 찾지 못해 ‘(엘비스, 많다, 역사상)’과 같이 잘못된 트리플로 뽑힌다.

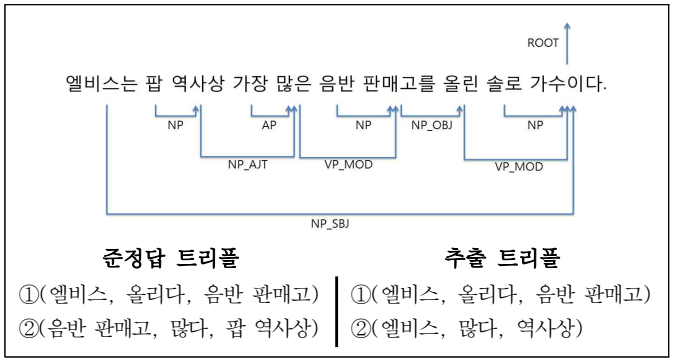


그림 7 종속절에서 추출된 오류 트리플

4.4 트리플 추출기 미고려 사항

현재 트리플 추출기는 주어와 목적어의 범위에 대한 규칙이 없다. 따라서 트리플 추출기 평가 시 이를 고려하지 않기 위해 정답에서 제시한 주어(혹은 목적어)의 head가 추출된 트리플의 주어(혹은 목적어)의 head와 같다면 옳은 것으로 간주했다. 4.1절의 예를 빌어 설명하면, 문장 ‘(엘비스, 오르다, 전당)’과 ‘(엘비스, 오르다, 음악 명예의 전당)’을 같은 것으로 간주하고 있다. 하지만 추후에는 주어와 목적어의 범위에 대한 규칙을 추가하고, 평가할 필요가 있다.

또한, 트리플의 정규화를 위해 목적어의 조사를 생략해 표기하고 있다. 이 때문에 목적어 성분이 문장에서 어떤 역할을 하는지 파악하지 못하는 경우가 있다. [그림 3]의 트리플 ‘(엘비스, 가다, 멤피스)’를 예로 보면 조사 ‘로’가 생략되어 엘비스가 멤피스에서 간 것인지 엘비스가 멤피스로 간 것인지 트리플이 정확한 정보를 가지지 못한다. 따라서 목적어의 역할을 분명히할 수 있는 구분을 트리플에 추가해 표기할 필요가 있다.

5. 결론

본 논문에서는 한국어 의존 파싱 결과를 이용하여 트리플을 추출하기 위한 방법을 제안했다. 트리플 추출기는 수동으로 구축된 정답 집합과 준정답 집합으로 평가했다. 평가를 위해 엘비스 프레슬리에 대한 위키피디아 문서를 이용해 평가 집합을 구성했다. 평가 결과 정답 집합에 대한 F-measure는 60.75%, 준정답 집합에 대한 F-measure는 66.67%의 성능을 보였다. 향후에는 현재 해결하지 못한 병렬 문장에서의 트리플 추출, 종속절에서의 정확한 트리플 추출, 트리플 구성 요소(주어, 서술어, 목적어)의 범위 설정 문제 등을 해결하고, 목적어 성분의 역할을 분명히 할 수 있도록 구분 정보를 추가해야한다. 또한 현재 평가에 사용한 평가 집합보다 더 많은 양의 집합을 이용해 추출기의 성능을 보다 정확히 평가할 필요가 있다. 또한 평가 문장이 적어 확인하지 못했던 트리플 추출 규칙의 부족한 부분을 확인하고 추가할 필요가 있다.

감사의 글

본 연구는 미래창조과학부 및 한국산업기술평가관리원

의 산업융합원천기술개발사업(정보통신)의 일환으로 수행하였음. [100445577, (1세부) 휴먼 지식증강 서비스를 위한 지능진화형 Wise QA 플래폼 기술 개발]

참고문헌

[1] A. Culotta, J. Sorense, "Dependency tree kernels for relation extraction," In Proceeding of the 42nd annual Meeting on Association for Computational Linguistics, 2004.
[2] J. Cowie, W. Lehnert, "Information extraction," communication of the ACM, vol.39, no.1, pp.80-91, 1996.
[3] J. Leskovec, M. crobelnik, N. Milic-Frayling. "Learning sub-structure of document semantic graphs for document summarization," In Proceeding of the 7th International Multiconference Information Society IS 2004, vol.B, pp.18-25, 2004.
[4] D. Rusu, L. Dali, B. Fortuna, M. Grobelnik, D. Mladenic, "Triplet extraction from sentences," In Proceeding of the 10th International Multiconference Information Soociety IS 2007, pp.8-12, 2007.
[5] Stanford Parser web page : http://nlp.stanford.edu/software/lex-parser.shtml
[6] Link Parser web page: http://www.link.cs.cmu.edu/link/.
[7] L. Dail, B. Fortuna, "Triplet extraction from sentences using SVM," In Proceedings of SiKDD, 2008
[8] D. Choi, K. Choi, "Automatic relation triple extraction by dependency parse tree traversing," Poster and Demo, pp.23-24, 2008.
[9] 정창후, 전홍우, 송사광, 홍순찬, 정한민, 최성필, "술어-논항 구조의 어휘 패턴을 이용한 스티링 커널 기반 관계 추출," 정보과학회 논문지: 소프트 웨어 및 응용 제39권, 제12호, pp.927-993, 2012.
[10] 국립국어원, 21세기 세종계획 최종 성과물(2011년 12월 수정판), 2011.
[11] 한국어 위키피디아 web page : http://ko.wikipedia.org/

P 언어를 이용한 한글 프로그래밍

최시영
싸이브레인 연구소
sychoi2100@gmail.com

Programming with Korean Vocabulary by Using P Language

Sea-Young Choi
Cybrain Laboratory

요 약

본 논문에서는 모국어를 이용한 프로그래밍을 위한 지원 방법으로서, 모국어로 된 데이터의 표현, 변수의 모국어 표현, 문법 키워드의 모국어 표현, 모국어 병행표현 등에 대하여 알아본다. 그리고 임의의 다국어를 지원하도록 설계된 P 언어를 이용하여 한글 프로그래밍을 하는 방법을 알아본다. 구체적으로 한글 프로그래밍 환경을 구축하는 방법, 한글 프로그램을 위한 어휘의 선정에 대하여 알아보고, 이를 이용한 간단한 알고리즘의 구현과 art 모듈을 이용하여 그래픽 프로그래밍의 예를 들어보겠다. 그리고 한글 프로그래밍을 위해 P 언어를 사용한 경우의 장점과 단점에 대하여 알아보겠다. 끝으로 한글 프로그래밍의 발전을 위해서는 표준 한글어휘 선정을 위해 학계와 산업계의 통일된 노력이 필요하다는 점과 한글 프로그래밍이 가져다 줄 수 있는 영향과 한글 프로그래밍의 바른 사용 방법으로서 영문과 한글의 병용사용을 제안한다.

주제어: 모국어 프로그래밍, 자국어 프로그래밍, P 언어, 한글 프로그래밍

1. 서론

자연어 분야에서 모국어가 아닌 외국어를 배우는 과정은 오랜 시간과 노력이 필요하다. 이와 비슷하게 프로그래밍 언어의 영역에서도 모국어에 기반하지 않은 프로그래밍 언어를 새로 배우는 과정도 오랜 시간과 노력을 요구한다. 이런 추가적인 학습비용에도 불구하고 자신의 모국어에 기반하지 않는 프로그래밍을 배우는 이들은 해당 프로그래밍 언어가 기반하는 자연어를 모국어로 하는 문화권의 사람들보다 기술적으로 높은 성과를 내기에는 여러 가지 어려움이 존재한다. 비모국어 사용자에게는 프로그래밍에 대한 입문 자체를 어렵게 하고, 숙련된 소프트웨어 개발자들에게도 외국어로 작성된 프로그래밍 소스와 기술문서에 대한 이해에 많은 어려움이 존재하게 한다.

1946년 미국에서 최초의 전자식 일반목적(general purpose)용 컴퓨터인 ENIAC(Electronic Numerical Integrator And Computer)[1]이 만들어진 이후로 많은 후속 컴퓨터 모델들이 미국에서 만들어졌다. 프로그래밍 언어는 이들 컴퓨터 하드웨어를 조작하기 위한 도구로서 탄생된 것이기 때문에, 프로그래밍 언어는 이들 하드웨어들이 제조된 미국에서 많이 만들어지게 되었다. 이런 이유로 대부분의 프로그래밍 언어는 영어에 기반하여서 만들어져 왔다. 프로그래밍 언어는 그 언어를 만든 개인 또는 회사의 모국어에 자연스레 영향을 받게 된다. 구체적으로 프로그래밍 언어 속의 문법어, 데이터 타입, 함수 이름, 언어 기술서등이 모두 해당 언어로 만들어지기 때문이다. 예를 들어 한국인 프로그래머는 영어로 작성된 문서상의 'concat' 라는 함수가 'concatenate' 라는 단어에서 나왔으며, 그 함수의 역할이 문자열을 연결시킨다는 것을 예상하기는 쉽지 않다.

영어에 기반하고 있는 프로그래밍 언어들

초기의 프로그래밍 언어들인 IPL(1954), FORTRAN(1955), LISP(1958)[2], ALGOL(1958), COBOL(1959) 들과, 그 후에 나온 RPG(1959), APL(1962), Simula(1962), SNOBOL(1962),CPL(1963) 등의 언어들은 모두 영어에 기반하고 있다. 역사적으로 학계에서 많이 사용되거나 산업계에서 대중화된 언어들인 BASIC, C, C++, Smalltalk, Prolog, ML, Pascal, Forth, SQL, Ada, Modula, Java, Erlang, Perl, Tcl, Delphi 등의 언어들도 영어에 기반하고 있다. 최근에 나온 Haskell[3], Python[4], Ruby[5], Lua[6], Scheme[7], Javascript, PHP[8], D[9], C# (Microsoft Inc., ISO/IEC/2327)[10] 등의 언어들도 영어에 기반하고 있다. Ruby 언어는 1990년경에 일본의 마츠모토 유키히로가 만든 언어이지만 영어에 기반하고 있다. Prolog, Python, Erlang 등도 비영어권 나라에서 개발되었지만 영어에 기반하고 있다.

본 논문에서는 프로그래머 또는 사회제도에 각각 초점을 맞춘 표현들인 '모국어' (mother language) 와 '자국어' (national language)를 같은 의미로 혼용해서 사용하겠다. 이에 따라 '모국어 프로그래밍', '모국어를 이용한 프로그래밍', '자국어 프로그래밍', '자국어를 이용한 프로그래밍' 들을 모두 같은 의미로 사용하겠다.

2. 관련 연구

많은 프로그래밍 언어가 영어에 기반하고 있기는 하지만, 산업계 중심으로 모국어로 프로그래밍을 하고자 하

는 노력이 부단하게 이루어져 왔다.

2.1. 소프트웨어에 있어서 자국어 지원을 위한 기술들

영어가 아닌 자국어에 기반한 소프트웨어를 사용하고자 하는 기술은 실용적인 이유에서 학계보다는 산업계에서 많이 연구되어온 주제이다. 소프트웨어의 일반 사용자들에게는 무엇보다도 자국어를 포함한 데이터를 표현하는 것이 가장 시급한 문제이었기 때문에 자국어 문자를 표현할 수 있는 문자집합에 대한 기술이 가장 먼저 개발되었다. 그리고 일반 사용자들이 자국어에 기반하여 소프트웨어를 사용할 수 있도록 메뉴, 시간, 통화 등을 표현하는 기술인 소위 '국제화와 지역화'에 관련된 기술이 발달되었다.

2.1.1. 문자집합(character set)

문자집합(character set, charset)은 문자들의 집합을 정의한 것을 말한다. 이와 비슷한 개념으로서 문자인코딩(character encoding)이라는 것이 있는데, 문자인코딩은 이러한 문자들을 부호화하는 방식이다. 그러므로 문자가 한 바이트에서 어느 영역에 할당되는 가 하는 것과 한 개의 문자가 몇 개의 바이트로 표현되는가 하는 문제들은 모두 문자인코딩의 문제이다. 그러나 이 두 가지는 실제에서는 거의 비슷한 의미로 사용되고 있다. 예를 들어 일부 문헌(대표적으로 미국 마이크로소프트사의 문서)에서 사용하는 다중바이트문자셋(multibyte character set) 이라는 표현은 부적절한 표현(misnomer)이다.

컴퓨터 초창기에는 4 bit로 이루어진 이진수를 묶어서 10진법 한자리수를 표현하는 BCD (Binary-Coded Decimal)방식[11] 을 개량해 IBM 회사가 6 bit를 사용해 숫자뿐만 아니라 모든 알파벳 문자와 특수기호를 표현하는 BCDIC (Binary-Coded Decimal Interchange Code) 방식을 개발하여 사용하였다. BCDIC 방식은 통일된 방식이 없이 각 회사별로, 각 제품 별로 다르게 정의되어서 사용되고 있었다. 그 후 미국국립표준협회(The American National Standards Institute, ANSI)가 1963년에 ASCII 표준[12]를 만들어서 이 방식이 널리 사용되게 되었다. 그러나 ASCII 는 영문자만을 지원해서, 한글문자, 한자, 일본문자 등의 동양권 언어들의 문자를 지원할 수 없어서, IBM 과 마이크로소프트사에서는 소위 multi-byte character set을 만들어 우리나라의 KSC 5601-1987 등을 지원하였다. 그러나 동양 3국의 문자뿐만 아니라 현존하는 모든 문자들을 표현하기 위한 필요성이 제기되면서, 마침내 1991년 유니코드협회(Unicode Consortium, Unicode Inc.)에서 유니코드(unicode) 표준[13]이 나왔으며, 이것이 널리 사용되게 되면서 현재로서는 산업계의 표준이라고 할 수 있게 되었다. 유니코드 표준은 1991년8월에 버전1.0 이 나온 이후로 현재까지 2012년9월에 버전 6.2 가 나왔다. 유니코드는 문자셋에 대한 이름이며, 유니코드를 부호화하는 유니코드 인코딩 방식으로서는 UTF-7, UTF-8, UTF-16, UTF-32 등이 있으며, 그 중에서도 UTF-8 이 널리 사용되고 있다.

2.1.2. 국제화와 지역화(Internationalization and localization)

국제화와 지역화는 소프트웨어를 여러 지역 환경에 공통적으로 사용할 수 있도록 만들어 각 지역 환경에 부합하여 변경하는 것을 말한다. 국제화는 보통 I18N 으로 표기하는데, I18N은 'internationalization' 의 첫 글자 'I' 와 마지막 글자 'N' 과 그 사이의 'nternationalizatio' 의 글자수인 18을 결합해서 만든 신조어이다. 이 국제화와 현지화는 주로 응용소프트웨어에 관하여 발전된 기술이며, 각 지역의 문자집합, 날짜/시간형식, 통화표시 방법 등에 관한 내용을 다루고 있다. 국제화와 지역화는 별개로 존재하는 것이 아니라 서로가 상대를 가정하여 존재하며 서로 영향을 미치는 상호 순환하는 과정이다(그림 1 참조)

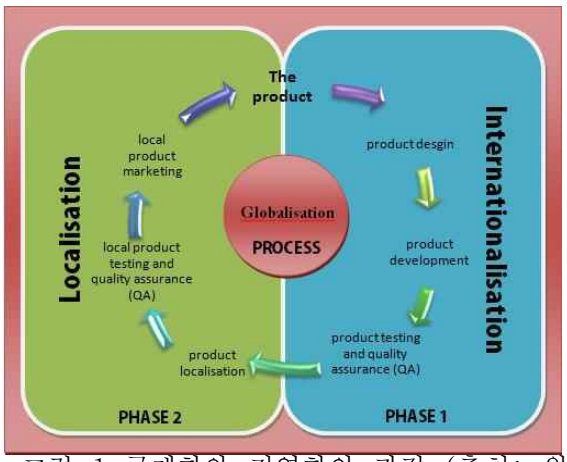


그림 1 국제화와 지역화의 과정 (출처: 위키피디어, IaT_vicky[14])

2.2. 프로그래밍 언어에서의 자국어 지원을 위한 연구

프로그래밍 언어 수준에서 자국어를 지원하고자 하는 필요성과 노력은 많이 있었으나, 명확한 규격이나 구현까지 이루진 것은 많지 않다.

2.2.1. ALGOL 68 의 키워드의 자국어표현

ALGOL 언어는 소위 GAMM-ACM 위원회를 통하여 미국과 유럽의 양 대륙 컴퓨팅 학계의 공동노력으로 이루어진 것이어서, 영어뿐만 아니라, 독일어, 불어, 러시아, 불가리어 같은 유럽의 언어로도 표현될 수 있는 것이 필요하였다. 특히 ALGOL 68 은 ALGOL 전통에 따라 여러 가지 형식의 표현법을 지원하였다. 즉, 표현 언어(representation language) 로서 참조언어(reference language), 출판언어(publication language),하드웨어언어(hardware language)를 지원하고 있다[15][16]. 영어가 아닌 다른 자국어에 기반한 문법 키워드로 구성되는 언어 표현을 지원할 수 있는 여지를 두었고, 실제로 러

시아어로 된 ALGOL 68 규격도 나왔다[17].

2.2.2. 프로그래밍 언어의 특정 자국어 판(localized version)

이들 언어들은 영어에 기반한 특정 프로그래밍 언어의 키워드를 자국어로 변경한 언어들이다. 이들 언어들은 원본 언어들과 분리되어 개발과 배포되어, 원본 언어들의 발전사항을 반영하지 못하는 제약점이 있고, 해당 자국어에 고정화(hardcoded) 되어 해당 자국어 이외로의 번역이 불가하여 해당 자국어를 사용하는 프로그래머만 이해할 수 있는 단점이 있었다. 각 나라별로 여러 가지 언어들이 등장하였다. 중국어의 경우, BASIC 언어에 기반한 中文培基(Chinese BASIC), Python 언어에 기반한 中文파이션[18]등이 있다.

2.2.3. 관련 개념들

Source-to-source translation: 한 프로그래밍 언어로 된 소스를 다른 프로그래밍 언어로 된 소스로 바꾸는 것을 말한다. 예를 들면 Ada 언어로 작성된 소스를 Pascal 언어 또는 C 언어로 바꾸는 경우를 말한다.

Natural Language Processing: 인간이 사용하는 자연어에 대하여 프로그래밍 기법을 통하여 분석하는 것을 말하며, NLP 라고 자주 통칭된다.

2.3. 전산용어의 한글화 노력

국립국어원에서는 1991년부터 2002년까지 각 분야별 “국어 순화 자료집”을 발간하였다. 그중에 1996년도에는 전산 관련 한글 순화 자료집을 발간하였는데, 이것은 “전산기 용어”라는 분야에서 전산 관련 일반적인 용어에 관하여 정하였으며 프로그래밍 언어에 특화되지는 않았다. 2003년에는 각 분야별 국어 순화 자료집을 모아서 “국어 순화 자료집 합본”으로 출간하였다[19]. 이러한 국어 순화의 노력에도 불구하고, 일반 국민에게는 제대로 알려지지 않았으며, 일반 국민들 또한 이에 대한 관심이 극히 낮았다고 한다[20]. 그 이유는 국어순화가 관련 몇몇 전문가 중심으로 이루어져 일반 국민들의 반응을 전혀 살펴보지 않은 채 순화어를 대량으로 만들어 일방적으로 보급하여 일반 국민의 언어의식에 합치되지 않았기 때문이라고 한다[20].

3. 모국어 프로그래밍의 지원방법

모국어 프로그래밍을 지원하기 위한 영역은, 언어 내적인 부분과 언어 외적인 부분으로 나뉜다. 언어 외적인 부분으로서는 해당 언어를 실행하는 실행기(인터프리터 또는 컴파일러)의 사용과, 언어 매뉴얼 등의 언어관련 문서에서 해당 언어를 지원하는가의 여부이다. 언어 실행기의 사용자 메뉴와 오류 메시지 또는 디버그 메시지 등이 해당 언어를 지원하여야 한다. 언어 내적인 부분은 다음과 같은 영역을 지원하여야 한다.

3.1. 모국어 데이터의 표현(입력과 출력)

모국어로 된 데이터를 표현할 수 있는가 하는 점이다. 예를 들어 문자열(string)에서 모국어 문자열을 사용할 수 있는가 또는 모국어로 된 파일이름을 지정할 수 있는가 하는 점이다. 이를 위해서는 모국어 문자집합(character set) 의 구현뿐만 아니라, 모국어 문자의 입력과 출력을 포함한다. 예를 들어 키보드 상에서 직접 한글을 입력시킬 수 있어야 하며, 모니터 화면상으로 한글이 표현될 수 있어야 한다. 다음은 모국어 문자열의 예이다.

“ครสิมาสสหกีำลงจะมา”, “圣诞节快到了”

3.2. 모국어 이름의 변수 사용

모국어에 기반한 문자로 바로 식별자(identifier)를 만들 수 있어야 한다. 예를 들면 다음과 같은 방식의 프로그래밍이 가능하여야 한다.

```
金額 = 100 ;
병의크기 = 갑의크기 + 을의크기 ;
私のリスト = [ 1,2, 3, 4] ;
```

3.3. 모국어로 된 문법어(키워드)

문법어를 이루는 키워드가 모국어로 표현될 수 있어야 한다. 문법 키워드는 파서를 만들 때 파싱(parsing)의 중요한 기준점이 되며 각 언어의 핵심에 해당하는 부분이다. 그러므로 각각의 모국어에 따른 다른 문법 키워드를 지원하기 위해서는 원칙적으로 각각의 모국어에 따른 개별적인 파서를 만들어야 한다는 어려움이 존재한다. 모국어로 된 문법어가 지원된다면, 예를 들어 다음과 같은 방식의 표현이 가능하다.

```
만일에( 갑의수학점수 > 90 )
{
   반복하기( 3번, 출력하기( “축하! ” ) ) ;
   출력하기( “참 잘 했습니다.” );
}
```

이것에는 한 개의 모국어만을 지원하는 경우(이글 2.2.2.참조)나, 특별히 지정된 몇 개의 모국어가 지원 가능한 경우(ALGOL 68), 임의의 모국어를 지원 가능한 경우(P 언어)로 나누어 볼 수 있다.

3.4. 모국어 병행표현(모국어전환)

병행표현은 특정 모국어에 기반한 프로그래밍 소스를 다른 모국어로 변경할 수 있는 가 하는 점이다. 모국어 프로그래밍의 가장 큰 장점이자 단점은 해당 모국어로 작성된 프로그래밍 소스가 해당 모국어 사용자에게는 판

독성이 높아지나, 외국어 사용자에게는 이해가 불가한 소스가 된다는 점이다. 모국어 간의 전환은 이러한 단점을 보완해주는 기능을 하게 된다. 예를 들어, 병행표현이 지원된다면, 아래의 중국어로 된 소스는 한글로 된 다른 소스로 변경할 수 있게 되고, 프로그램 실행시 동일한 결과가 나와야 한다.

```
函数定義 新增评论(自己, 稿件)
{
    寫( "<!--" + 自己 + "-->" );
    寫( 稿件 );
}

함수정의하기 주석달기(내이름, 문건)
{
   쓰기( "<!--" + 내이름 + "-->" );
   쓰기( 문건 );
}
```

4. P 언어를 이용한 한글 프로그래밍

4.1. P 언어란

P 언어는 일반목적(general-purpose)용 동적인 스크립팅(scripting)형 언어이다. P 언어는 멀티패러다임을 지원하며, 특히 객체지향과 함수형 프로그래밍을 동시에 지원한다. P 언어의 특징[21] 으로서는 a) 간결한 문법을 사용하면서도 표현력이 뛰어나고, b) 복잡한 알고리즘을 쉽게 표현할 수 있으며, c) 모든 데이터는 함수 호출을 통해 생성되며, d) 모든 제어구조는 함수로 표현되고, e) 모든 연산자는 함수의 이름에 불과하며, f) 이름과 값이 분리돼 있는, g) 식 중심의 언어(expression-oriented programming language)이다.

4.2. P 언어의 한글 프로그래밍의 지원

P 언어는 자연어 중립적인 언어이며 어느 자연어에 구속된 언어는 아니다. 그러나 P 언어는 처음부터 모든 개별 자연어를 지원하기 위해서 디자인이 된 언어로서, 개별 자연어에 대한 설정 파일만 만들어 주면, 해당 자연어에 기반한 프로그래밍이 가능하게 설계되었다. 그리고 위에서 서술한 모국어 프로그래밍의 4 가지 지원분야를 비교적 충실히 지원한다.

4.3. 한글프로그래밍의 환경구축

ㄱ) 개별 프로그래머가 정한 한글 어휘를 지원하는 방법: P 언어가 실행되는 인터프리터 상에서 P 언어 소스를 입력하는 것과 동일한 방식을 사용하여, 프로그래머가 선정한 한글을 지원할 수 있다. 인터프리터 상에서 다음과 같은 형식의 문장을 입력한다. 여기서 '새로운이름' 은 한글 어휘를 말한다.

새로운이름 = 기본이름 ;

예를 들어, 영어 'if' 라는 표현대신에 한글로 '만약에' 이라는 표현을 사용하고자 하는 경우에는 인터프리터 상에서 다음과 같이 입력한다.

```
만약에 = if ;
```

ㄴ) 설정 파일에 의한 환경 구축: 매번 인터프리터를 실행할 때 마다 프로그래머가 한글 환경을 일일이 구축하는 방식은 효율적이지 못하다. P 언어와 함께 배포되는 각 언어별 설정파일을 이용하여 한글 프로그래밍 환경을 구축할 수 있다. 이 방법을 사용하면 표준화된 한글어휘를 정할 수 있는 장점이 있다.

4.4. 한글 프로그래밍의 모습

P 언어를 사용해서 재귀 함수를 사용하는 '팩토리얼' 이라는 함수를 정의해보면 다음과 같다.

```
함수정의하기(팩토리얼, 숫자,
              만약에( 숫자 == 0 ,
                      1,
                     숫자*팩토리얼(숫자-1) )
) ;
팩토리얼(100);   // 함수사용법
```

객체지향 프로그래밍을 지원하는 파일 사용법은 다음과 같다.

```
철수파일 = 파일( "내파일" ) ;
철수파일.쓰기( "작성자:철수" ) ;
철수파일.닫기( ) ;
```

4.5. art 모듈의 사용

P 언어에서 기본적으로 제공하는 art 모듈 속의 간단한 함수를 이용하여 다음과 같은 그래픽 프로그래밍을 할 수 있다.

```
그림판(400,300);
반복하기( 10,  ( 앞으로(10),
                 원그리기(30),
                 왼쪽으로(65))) ;
```

표 1. art 함수

한글함수	영어함수	하는 일
그림판	canvas	그림판 만들기
앞으로	forward	앞으로 가기

뒤로	backward	뒤로 가기
오른쪽으로	right	오른쪽 방향전환
왼쪽으로	left	왼쪽 방향전환
원그리기	circle	원거리기

그림 2 art 모듈을 사용한 예 (출처 [22])

5. P 언어를 사용한 한글 프로그래밍의 장단점

5.1. P 언어를 사용한 한글 프로그래밍의 장점

ㄱ) 자유로운 한글 어휘의 선택가능: P 언어에서는 문법 키워드(또는 문법 키워드 역할을 하는 함수)가 syntax 수준이 아니라 실행시 이름공간(name space)상의 이름으로 존재하므로, 프로그래머가 임의적으로 원하는 한글 어휘를 사용하여 한글 프로그래밍을 할 수 있다.

ㄴ) P 언어는 다국어 지원이 언어 내재적으로 이루어지므로 다른 지역화 된 언어(이글 2.2.2.참조)들과 다르게, 각 다국어 버전이 독립적으로 존재하지 않고 하나의 버전 안에 존재하게 된다. 그러므로 P 언어의 발전(업그레이드)이 모든 다국어 버전에 그대로 반영되게 된다.

ㄷ)임의의 자국어 지원 가능 및 자국어간의 전환 가능

프로그래머가 표준화된 자국어 어휘를 사용하는 경우, 다른 자국어으로의 전환이 가능하여, 각 자국어 버전간의 소스전환이 이루어질 수 있다.

5.2. P 언어의 단점 및 미비점

P 언어는 위에서 서술한 바와 같이 한글 프로그래밍을 비교적 충실히 지원하지만, P 언어 디자인상의 자유로움으로 인해 언어 내적인 단점과 신규로 만들어진 언어로 인한 언어 외적인 미비점이 존재한다.

ㄱ) 프로그래머의 비표준화된 한글 어휘의 사용 : P 언어는 제어구조가 특정 키워드에 고정화된 언어가 아니므로 프로그래머가 P 언어에서 제공하는 표준화된 한글 어휘를 사용하지 않고 독자적인 한글 어휘를 사용하는 경우에는 가독성이 떨어질 수 있다. 예를 들어 프로그래머는 'if' 기능을 하는 한글어휘인 '만약에' 대신에 '만약', '만일', 참인경우' 등의 표현을 사용하는 경우에는 다른 외국어 버전으로 전환이 어려울 수 있다.

ㄴ) 충분한 라이브러리의 부족: P 언어는 문자열, 함수, 리스트, 배열 등의 다양한 데이터 타입들을 지원하고, 파일, 네트워크, 원도우를 위한 프로그래밍을 지원하지만, 다양한 소프트웨어를 손쉽게 만들기 위해서 필요한 여러 가지 라이브러리(language libraries)가 아직 지원되고 있지 않다.

ㄷ) 실행파일의 생성 지원 미비: 영리 목적의 소프트웨어 제작사의 경우에는 자신의 프로그래밍 소스를 공개하기보다는 기술 보호를 위해서 바이너리로 된 실행파일 형태의 소프트웨어 제작을 선호하는 편이다. P 언어는 현재로서는 스크립트 기반의 언어로서 모든 P 소스는 텍스트 파일 형태로 존재하여 공개되게 되어있다. 그러나 추후 P 언어를 지원하는 컴파일러가 제작이 되면, P 언어에서 바로 바이너리 코드로 만들어진 소프트웨어 제작이 가능할 것이다.

5. 결론

P 언어는 한글만을 위한 한글 전용 프로그래밍 언어는 아니지만, 임의의 다국어를 지원할 수 있는 언어 설계에 의하여 한글 프로그래밍이 가능하다는 것을 알 수 있었다.

한글 프로그래밍이 가능하다고 하여도 구체적으로 프로그래밍에 사용될 한글 어휘에 대한 결정은 언어 외적인 영역에 속한다. 위의 관련 연구(이글 2.3.참조)에서 언급하였던 한글 순화의 미비한 효과에서 알 수 있듯이 언어의 사용자인 일반 국민의 참여 없이 몇몇 전문가들에 의한 작업은 큰 성과를 내기가 어렵다. 프로그래밍 언어의 주된 사용자는 산업계이므로 한글 프로그래밍 어휘의 선정을 위해서는 산업계의 참여와 협조가 절대적으로 필요하다. 그러므로 한글 프로그래밍의 개척과 발전을 위해서는 한글 어휘에 대한 학계와 산업계의 통일된 표준이 존재하여야 할 것이다. 한글 프로그래밍의 발전은 프로그래밍 인구의 저변 확대뿐만 아니라, 더 중요한 영향으로서, 영어적 사고 외에 한국어라는 새로운 언어적 사고의 도입으로 인하여 프로그래밍 이론 분야에서 창의적이고 혁신적인 이론 개발에 기여할 것이다.

한글과 한자를 병행하여 사용할 때 우리말의 표현력과 정확성이 높아지듯이, 프로그래밍 분야에서도 한글 전용 또는 영어 전용보다는 한글과 영어를 적절히 배합하여 사용하는 경우에 좀 더 표현력과 정확성이 높아질 수 있다. 한글 프로그래밍은 기존의 영어 전용 환경에서의 불편함과 비효율성을 해소 하고 모국어의 활용성과 사고적 다양성을 높이기 위한 것이지, 한글 프로그래밍 그 자체를 목적으로 하는 것은 아니다.

참고문헌

[1] http://en.wikipedia.org/wiki/ENIAC

[2] McCarthy, John. "Recursive Functions of Symbolic Expressions and Their Computation by Machine,

Part I", 1960.
[3] http://www.haskell.org
[4] http://www.python.org
[5] http://www.ruby-lang.org
[6] http://www.lua.org
[7] Sussman, Gerry, Hal Abelson, and Julie Sussman, "Structure and interpretation of computer programs." The Massachusetts Institute of Technology 10, 1985.
[8] http://www.php.org
[9] http://www.dlang.org
[10] Information technology-Programming languages-C#, http://standards.iso.org/ittf/PubliclyAvailableStandards/c042926_ISO_IEC_23270_2006(E).zip
[11] Eric Fisher, "The Evolution of Character Codes, 1874 - 1968", http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.96.678&rep=rep1&type=pdf, Accessed 2013
[12] American Standards Association, "American Standard Code for Information Interchange", ASA X3.4-1963, June 1963.
[13] Unicode Consortium, "The Unicode Standard, volume 1.0." Addison-Wesley, Reading, MA, 1991.
[14] http://en.wikipedia.org/wiki/File:Globalisation chart.jpg
[15] Mailloux, Barry James, J. EL Peck, and C. HA Koster. "Report on the algorithmic language ALGOL 68." Numerische Mathematik 14.2, pp. 79-218, 1969.
[16] Van Wijngaarden, A., et al. "Revised Report on the Algorithmic language." Acta Informatica 5, pp. 1-236, 1975.
[17] "Programming language ALGOL 68 - Язык программирования АЛГОЛ 68", GOST 27974-88, 1988.
[18] http://www.chinesepython.org
[19] 국립국어원, "국어 순화자료집 합본", 2003.
[20] 고성환, "국어 순화의 역사와 전망", 국립국어원 새국어생활 Vol.181, pp. 5-6, 2011
[21] 최시영, "새로운 범용 프로그래밍 언어 data-p", 마이크로소프트웨어, pp. 254-255, 2012.10
[22] 최시영, "예제로 배우는 data-p 실전 프로그래밍", 마이크로소프트웨어, p. 257, 2013.2

부록 1. 데이터 타입의 한글이름 예

한글표현	영어표현	데이타유형
없음	none	null
진리값	bool	boolean
숫자	num	number
문자열	str	string
목록	list	list
배열	array	array
레코드	record	record
함수	func	function
매크로	macro	macro
변수	var	variable
상수	const	constant
클래스	class	class
인스턴스	instance	instance
모듈	module	module
자료집	file	file 객체
창	win	window 객체
망연결기	socket	sockect 객체

부록 2. 제어구조의 한글이름 예

한글표현	영어표현	하는 일
만약에	if	조건
동안에	while	조건부 반복
아닌동안에	unless	부조건부 반복
각각에	foreach	순환
경우에	case	택일 선택
계속하기	for	반복
반복하기	loop	단순반복
반환하기	return	함수결과 반환
중단하기	break	순환구조내 중단
다음하기	continue	순환구조내 다음과정
종료하기	exit	전체프로그램중단

포스터발표

영한 기계번역 시스템의 개선을 지원하는 영어 구문 규칙 관리 도구

김성동○, 김창희, 김태완
한성대학교, 컴퓨터공학과
sdkim@hansung.ac.kr, kimch3617@naver.com, madplay@naver.com

English Syntactic Rule Management Tool for Improving English-Korean Machine Translation System

Sung-Dong Kim○, Chang-Hee Kim, Tae-Wan Kim
Hansung University, Dept. of Computer Engineering

요 약

규칙 기반의 영한 기계번역을 위해서는 많은 영어 구문 규칙을 구축하고 관리해야 하는데, 이는 매우 많은 노력과 시간을 요구한다. 이 문제에 대한 해결방안으로 본 논문에서는 영어 구문 규칙의 효율적인 관리를 도와주는 도구를 제안한다. 영한 기계번역 시스템의 성능 개선 과정에서 영어 구문 규칙의 검색과 수정이 빈번하게 이루어지는데, 이러한 작업을 쉽게 할 수 있도록 제안하는 도구는 다양한 키를 이용한 규칙 검색과 규칙 수정 기능을 제공한다. 제안하는 도구는 영어 구문 규칙을 관리하는데 필요한 사람의 노력을 줄여 지속적인 영한 기계번역 시스템의 성능 개선 과정을 보다 손쉽게 할 수 있게 할 것이다.

주제어: 영한 기계번역, 규칙기반 기계번역, 구문 분석, 구문 규칙

1. 서론

규칙 기반 기계번역(rule-based machine translation: RBMT) 방법은 기계번역 방법 중 가장 오래된 방법으로서 언어학 기반(linguistic-based) 방법이라고도 불린다. 규칙 기반 기계번역 시스템을 개발하기 위해서는 많은 규칙을 전문가가 직접 구축해야 하며 이는 많은 시간과 노력을 필요로 한다. 그리고 번역 품질 개선 과정이나 새로운 영역에 적응시키는 작업에는 기존에 구축한 규칙을 지속적으로 개선하는 작업이 요구되는 지식 획득의 병목현상(knowledge acquisition bottleneck) 문제를 가진다. 다른 방법으로 1990년대 이후로 대량의 말뭉치가 구축됨에 따라 이국어 병렬 말뭉치(bilingual parallel corpus)를 이용한 통계적 기계번역 시스템에 대한 연구가 활발하게 진행되었다. 이 방법은 기계번역 시스템을 개발하고 유지하는 비용이 규칙 기반 방법에 비해 적다는 장점을 가지는데 비해 언어 구조나 문장 스타일 등을 적절하게 반영하여 번역하지 못하며 학습 과정이나 번역 에러를 조정하는 것이 어렵고 매우 많은 말뭉치 데이터를 필요로 한다는 단점을 가진다. 이에 비해 규칙 기반 방법은 전문가의 지식을 활용하여 보다 나은 번역을 생성할 수 있으며 예상되는 번역을 생성하기 때문에 에러를 수정하는 것이 통계적 방법에 비해 용이하다. 영한 기계번역이 다루는 두 언어는 매우 상이한 언어이므로 규칙 기반 방법을 이용하여 번역 시스템을 구축하고 이를 보강하는 측면에서 통계적 방법을 적용하는 것이 바람직하다는 판단이다. 이러한 하이브리드 방식의 기계번역 시스템으로는 SYSTRAN의 기계번역 시스템[1], Safaba의 EMTGlobal[2], PROMT의 기계번역 솔루션[3] 등이 있으며 실용화되고 있는 상황이다.

본 논문에서는 차트(chart) 기반의 구문 분석 방법[4]이 적용된 영한 기계번역 시스템의 효율적인 성능 개선 작업을 지원하기 위한 영어 구문 규칙 관리 도구를 제안한다. 규칙 기반의 영한 기계번역 시스템은 많은 영어 구문 규칙을 가지며 각 규칙마다 적절성 점수를 가지고 있어 파싱 트리를 생성하면서 결과 트리에 대한 적절성 점수를 계산한다. 문장 번역 테스트를 통해 번역 품질을 향상시키는 과정에서 영어 구문 규칙을 검색하여 규칙을 보완하고 수정하여 개선하는 과정이 필수적으로 요구되는데, 제안하는 도구는 이러한 과정을 손쉽게 수행할 수 있도록 지원하여 영한 기계번역 시스템의 성능 개선 작업을 도와주는 것을 목적으로 한다.

본 논문은 다음과 같이 구성된다. 2장에서는 기계번역 시스템의 개선 과정에서 사용될 구문 규칙 관리 도구가 갖추어야 할 요구 사항을 분석하고 3장에서는 이러한 요구사항을 수용하는 구문 규칙 관리 도구의 기능을 설명한다. 그리고 4장에서는 논문에서 제안하는 구문 규칙 관리 도구의 효과에 대해 설명하고 앞으로의 과제를 제시하며 논문을 마무리한다.

2. 구문 규칙 관리 도구의 요구 사항

문장 번역 테스트를 통한 영한 기계번역 시스템의 번역 품질 개선을 위한 구문 규칙 개선 작업 과정 다음과 같다. 첫째, 테스트 문장들을 일괄 번역(batch

translation)한 후에 번역 결과를 분석하여 잘못된 번역을 생성한 문장을 수집한다. 둘째, 이들 각각의 문장에 대해 개별 번역(interactive translation)을 수행하며, 이 때 구문 분석 결과인 차트와 파싱 트리(parsing tree)를 출력하면서 적용된 구문 규칙을 함께 출력하도록 한다. 셋째, 이 결과물을 이용하여 번역 오류가 잘못된 파싱 트리의 선택에 의한 것이었는지를 확인한다. 만약 번역 오류의 원인이 이 경우에 해당한다면, 올바른 구문 분석 트리를 생성하기 위한 규칙을 확인하여 이들 규칙이 적용된 구문 분석 트리를 찾고, 결과 트리와 비교하여 잘못된 트리가 선택된 원인을 파악한다. 이 과정에서 구문 규칙을 검색하여 분석하는 작업이 필요한데, 제안하는 도구는 규칙을 검색하여 그 내용을 쉽게 확인할 수 있는 기능을 제공해야 한다. 넷째, 올바른 트리를 생성하기 위한 구문 규칙과 결과 트리 생성에 적용된 규칙을 비교하고, 올바른 트리가 생성될 수 있도록 이들 규칙을 수정한다. 따라서 제안하는 도구는 규칙의 수정 기능을 제공해야 한다. 그리고 규칙이 수정되면 구문 분석기의 규칙 적용 모듈이 수정된 규칙에 맞게 수정되어야 하는데, 제안하는 도구는 규칙 적용 모듈의 수정 기능 역시 제공해야 한다. 그림 1은 영한 기계번역 시스템의 구문 규칙의 예를 보여준다.

```
NP010: NP(%NOUN, cat='NP, OBJP=1, SUBJP=1,
   RefIndicToScore NOUN POSSESIVE 1 0.5,
   score=1.02)
   <- NOUN(COORDED=0)
```

그림 1. 구문 규칙의 예

구문 규칙은 [규칙 이름: 규칙의 좌측(left-hand side) <- 규칙의 우측(right-hand side)]의 형식을 가진다. 규칙의 좌측과 우측은 [문법 기호(행동/조건)]의 형식을 가지는데, "문법 기호"는 NP, VP 등의 구를 나타내는 기호와 NOUN, ADJ 등의 품사를 나타내는 기호를 포함한다. 영한 기계번역 시스템의 구문 분석 모듈은 구문 규칙의 규칙 이름과 우측 기호들과 관련된 C 언어 함수로 구성된다. 예를 들어, 그림 1의 구문 규칙과 관련된 C 언어 함수는 NP010()과 NP010noun() 함수가 있다.

결과적으로 제안하는 구문 규칙 관리 도구는 다음과 같은 기능을 필요로 한다. 첫째, 구문 규칙 검색 기능이다. 구문 규칙의 검색을 쉽게 하기 위해 규칙의 좌측 기호, 우측 기호들, 규칙 이름 등을 키(key)로 하는 검색이 필요하다. 둘째, 구문 규칙의 비교를 위해 규칙의 모든 내용을 보기 좋게 보여줄 뿐만 아니라 여러 개의 규칙을 동시에 보여줄 수 있어야 한다. 이와 함께 규칙에 해당하는 구문 분석 모듈을 보여줄 필요가 있다. 잘못된 구문 분석의 원인을 파악할 때, 규칙을 분석하기 보다는 구문 분석 모듈, 즉 C 언어 함수를 분석하는 것이 더 용이할 수 있기 때문이다. 셋째, 구문 규칙과 해당하는 C 언어 함수의 수정 기능이 필요하다.

3. 구문 규칙 관리 도구의 기능

2장에서 제시한 구문 규칙 관리 도구의 요구사항을 만족시키기 위해서 제안하는 도구는 다음과 같은 기능을 제공한다: 규칙 검색 기능, 규칙과 대응하는 구문 분석 모듈 보여주기 기능, 규칙 수정과 대응하는 구문 규칙 관리 모듈 수정 기능.

본 장에서는 구문 규칙 관리 도구의 기능을 설명함에 있어 사용자 인터페이스의 모습을 제시하고 도구를 사용하는 방법을 중심으로 설명한다.

3.1 구문 규칙 검색 기능

적절하지 않은 구문 분석 트리 선택에 의해 잘못된 번역이 생성되었을 경우, 결과로 선택된 구문 분석 트리를 생성하는데 적용된 구문 규칙을 확인해야 한다. 올바른 번역 생성을 위한 트리가 선택되지 않았던 이유는 경쟁하는 구문 규칙들 중에서 적절한 구문 규칙의 점수가 선택된 구문 규칙의 점수보다 낮거나 적절한 구문 규칙을 적용하기 위한 조건이 맞지 않았기 때문이다. 따라서 경쟁하는 구문 규칙들을 모두 확인하여 규칙의 조건부와 점수를 비교, 분석해야 한다. 그림 2는 구문 규칙 검색 화면의 모습이다.

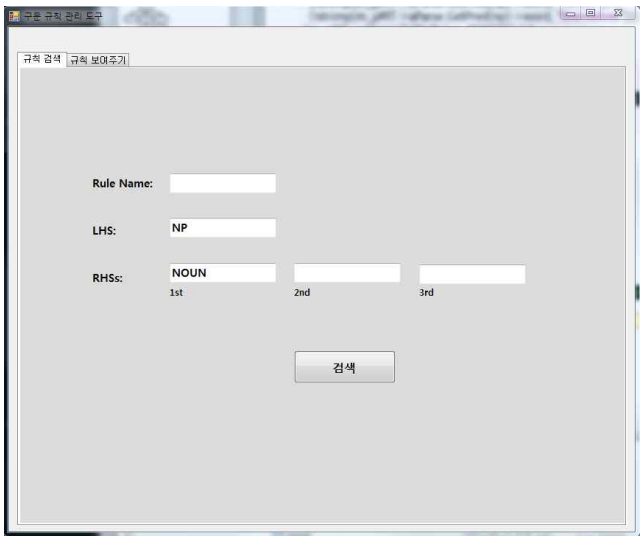


그림 2. 구문 규칙 검색 화면

경쟁하는 구문 규칙을 쉽게 확인하기 위해서 구문 규칙 검색 기능은 2장에서 제시한 것처럼 규칙 이름, 규칙의 좌측 기호(LHS), 우측 기호(RHS)들을 키로 하여 검색할 수 있다. 원하는 키를 입력하고 "검색" 버튼을 누르면 검색 키에 맞는 구문 규칙을 검색하고 결과는 "규칙 보여주기" 탭에서 보여준다. 우측 기호는 하나 이상을 키로 할 수 있으며, 좌측 기호와 우측 기호를 조합하여

키를 구성하여 검색할 수도 있다. 그림 3은 LHS가 NP이고 RHS의 첫 번째 기호가 NOUN인 구문 규칙을 검색하기 위해 키를 설정한 모습을 나타낸다.

3.2 구문 규칙과 구문 분석 모듈 보여주기 기능

검색한 구문 규칙을 보여줄 때, 대응하는 구문 분석 모듈의 C 언어 함수도 함께 보여준다. 그림 3은 규칙 검색 결과를 보여주는 화면의 모습이다.

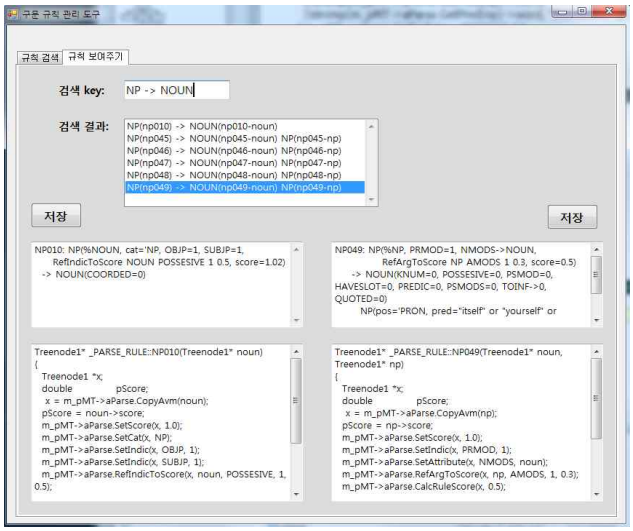


그림 3. 구문 규칙 보여주기 화면

그림에서 "검색 key" 부분은 검색에 사용된 key가 무엇인지를 확인할 수 있게 하며, "검색 결과" 부분은 검색 key와 일치한 구문 규칙들의 목록을 보여준다. 특정한 규칙을 선택하여 클릭하면 먼저 좌측에 구문 규칙과 해당하는 C 언어 함수를 보여주고 다른 규칙을 선택하여 클릭하면 오른편에 보여줌으로써 규칙의 비교를 용이하게 한다. 그림 4는 먼저 "np010" 규칙을 선택하여 좌측에 그 내용이 나타난 후, "np049" 규칙을 선택하여 우측에 내용이 나타난 모습을 보여준다.

3.3 구문 규칙과 구문 분석 모듈 수정 기능

구문 규칙들을 분석하여 잘못된 트리를 선택한 원인을 파악하였다면 구문 분석과 대응하는 C 언어 함수를 수정해야 한다. 원칙적으로는 규칙을 수정하면 대응하는 C 언어 함수를 직접 C 언어 함수를 수정한다. 그림 3에서 제시되는 C 언어 함수를 수정하면 영한 기계번역 시스템의 구문 분석 모듈이 수정되며, 수정된 번역 시스템으로 번역을 수행하여 수정에 의해 올바른 트리를 선택하여 올바른 번역이 생성되는지를 확인하는 과정으로 구문 규칙을 개선하는 작업을 수행한다.

그림 3에서 구문 규칙이나 C 언어 함수를 수정한 후 "저장" 버튼을 누르면 시스템의 규칙과 C 언어 함수가 수정 된다. 화면에 2개의 "저장" 버튼이 있는데 각각 해당하는 위치의 규칙과 함수를 저장한다. 그리고 수정에 대한 기록은 로그 파일에 저장되어 구문 규칙과 모듈에 대한 수정 내용을 나중에 확인할 수 있다.

4. 결론

본 논문에서는 규칙 기반의 영한 기계번역 시스템의 구문 규칙 개선을 통한 지속적인 성능 개선을 지원하기 위한 구문 규칙 관리 도구를 제안하였다. 제안하는 도구는 번역 테스트 과정에서 구문 규칙의 잘못된 적용으로 인해 잘못된 번역이 생성된 경우에 그 원인을 파악하여 올바른 번역을 생성할 수 있도록 기존의 구문 규칙을 수정하는 작업을 쉽게 할 수 있도록 지원한다. 즉 적용된 구문 규칙을 쉽게 검색할 수 있도록 다양한 키를 이용한 검색 기능을 제공하고 구문 규칙과 대응하는 C 언어 함수를 동시에 보여줌으로써 문제가 되는 구문 규칙과 모듈을 쉽게 확인하고 수정할 수 있도록 해준다. 이를 통해 구문 규칙의 개선에 의한 번역 시스템의 성능 개선 과정을 쉽게 한다.

현재는 기존 규칙을 확인하고 수정하기만 할 수 있으나 필요하면 새로운 구문 규칙을 추가하거나 필요 없는 규칙을 제거할 경우를 지원할 수 있도록 구문 규칙 관리 도구가 개선되어야 할 것이다. 구문 규칙이 삭제되면 이에 해당하는 구문 분석 모듈의 C 언어 함수가 삭제되어야 하며 추가되면 C 언어 함수가 새로 생성되어야 한다. 따라서 구문 규칙을 자동으로 C 언어 함수로 변환하는 도구에 대한 연구가 필요하다.

참고문헌

[1] SYSTRAN Translator, SYSTRAN, www.systransoft.com
[2] EMTGlobal, Safaba Translation Innovation, www.safaba.com.
[3] PROMT translator, PROMT, www.promt.com
[4] Terry Winograd, Language a s a Cognitive process: Vol. 1, Syntax, Addison Wesley, 1983.

음식메뉴 개체명 인식을 위한 음식메뉴 사전 자동 구축 방법

구영현[0], 유성준[1)]
세종대학교, 컴퓨터공학과
yhgu@sejong.ac.kr, sjyoo@sejong.ac.kr

Automatic Construction of Restaurant Menu Dictionary

Yeong-Hyeon Gu[0], Seong-Joon Yoo
Sejong University, School of Computer Science

요 약

레스토랑 리뷰 분석을 위해서는 음식메뉴 개체명 인식이 매우 중요하다. 그러나 현재의 개체명 사전을 이용하여 리뷰 분석을 할 경우 구체적이고 복잡한 음식메뉴명을 표현하는데 충분하지 않으며 지속적인 업데이트가 힘들어 새로운 트렌드의 음식 메뉴명 등이 반영되지 않는 문제가 있다. 본 논문[2)]에서는 레스토랑 전문 사이트와 레시피 제공 사이트에서 각 레스토랑의 메뉴 정보와 음식명 등을 래퍼기반 웹 크롤러로 수집하였다. 그런 다음 빈도수가 낮은 음식메뉴와 레스토랑 온라인 리뷰에서 쓰이지 않는 음식메뉴를 제거하여 레스토랑 음식 메뉴 사전을 자동으로 구축하였다. 그리고 레스토랑 온라인 리뷰 문서를 이용해 음식메뉴 사전의 엔티티들이 어느 유형의 레스토랑 리뷰에서 발견되는지를 찾아 빈도수를 구하고 분류 정보에 따른 비율을 사전에 추가하였다. 이 정보를 이용해 여러 분류 유형에 해당되는 음식메뉴를 구분할 수 있다. 실험 결과 한국관광공사 외국어 용례사전의 음식 메뉴명은 1,104개의 메뉴가 실제 레스토랑 리뷰에서 쓰인데 비해 본 논문에서 구축한 사전은 1,602개의 메뉴가 실제 레스토랑 리뷰에서 쓰여 498개의 어휘가 더 구성되어 있는 것을 확인 할 수 있었다. 이와 아울러, 자동으로 수집한 메뉴의 정확도와 재현율을 분석한다. 실험 결과 정확률은 96.2였고 재현율은 78.4, F-Score는 86.4였다.

주제어: 음식메뉴 사전, 개체명 인식

1. 서론

레스토랑 음식 메뉴는 레스토랑이라는 의견 대상의 속성들 중 가장 높은 비율을 차지하며 많은 정보를 포함하고 있다. 대부분의 레스토랑의 경우 여러 가지 음식메뉴를 갖고 있는데 레스토랑에 대해서만 의견 분석을 할 경우에는 레스토랑 자체에 대한 의견 분석만 가능하다. 그러나 음식메뉴라는 레스토랑의 속성을 가지고 의견 분석을 하면 음식메뉴들에 대해 개별적인 긍정/부정 정보를 알 수 있다. 예를 들어 음식 메뉴 속성을 이용하여 분석을 하면, 메뉴1은 부정 의견이고 메뉴2는 긍정 의견인 경우에 각각의 평가 정보를 얻을 수 있지만 레스토랑을 대상으로 분석을 하면 어느 메뉴가 더 평가가 좋은지 알 수 없다.

현재 한국어 음식메뉴사전은 전자통신연구원에서 구축한 한국어 개체명 사전 DB[1]와 한국관광공사에서 구축한 관광용어 외국어 용례사전[2]이 있다. 전자통신연구원에서 개발한 사전은 200개의 엔트리를 가지고 있으며 추상물, 구체물, 표상물, 조직 등으로 분류하였다. 그 중 음식과 관련된 엔트리들은 구체물의 음식으로 분류하여 관리한다. 한국관광공사에서 구축한 사전은 외국 관광객들을 위한 표준화된 레스토랑 메뉴판 작성을 목적으로 개발되었으며 음식 분류정보도 함께 제공된다.

전자통신연구원에서 구축한 사전은 일반적인 목적의 개체명 사전이다. 일반적인 목적의 개체명 사전과는 다르게 레스토랑 도메인의 음식메뉴명은 재료와 조리방법에 따라 많은 파생어들이 있기 때문에 구체적이고 복잡한 음식메뉴명을 표현하는데 충분하지 않다. 그리고 한국관광공사에서 구축한 사전은 음식메뉴명과 함께 음식메뉴의 분류를 제공하지만 실제 분류와 다른 경우가 발생한다. 또한 음식메뉴명은 주기적으로 새로운 음식메뉴가 업데이트 된다는 특징이 있는데 이를 반영하기 위해 지속적인 업데이트를 하려면 시간과 비용이 많이 소요된다.

이러한 문제점들을 해결하기 위해 본 논문에서는 레스토랑 전문 사이트와 레시피 제공 사이트를 이용해 자동으로 레스토랑 음식메뉴 사전을 구성하는 방법을 제안한다.

이를 위해 복수의 레스토랑 전문 사이트들과 레시피 제공 사이트에서 래퍼 기반 웹 크롤러를 이용해 메뉴판 정보, 인기메뉴, 추천메뉴 등과 같은 레스토랑 음식메뉴와 관련된 정보를 주기적으로 추출한다. 그런 다음 빈도수를 측정하여 특정 빈도수 이하의 정보를 제거한다. 그리고 추가적으로 온라인 레스토랑 리뷰와 비교하여 실제로 리뷰어들이 쓰는 어휘인지 분석한 다음 사용 빈도수가 낮은 어휘들은 제거하여 음식메뉴 사전을 자동으로 구축한다.

---

1) 교신 저자

2) 본 연구는 미래창조과학부 및 한국산업기술평가관리원의 산업융합원천기술개발사업(정보통신)의 일환으로 수행하였음. [10044454, 기기 정보뿐 아니라 사용자의 환경/감성/인지 정보에 적응적으로 반응하는 정보기기용 원격 UI 기술 개발]

음식메뉴는 한식, 양식, 일식, 주점 등과 같은 분류 유형 정보를 가지고 있다. 자동으로 구축된 음식메뉴 사전의 엔티티들이 어느 유형의 레스토랑 리뷰에서 사용되는지를 빈도수 기반으로 측정하여 음식 유형 분류 정보를 제공한다.

하나의 음식메뉴는 여러 개의 분류 정보를 가질 수 있는데 이러한 문제를 해결하기 위해 유형별로 음식메뉴가 발견된 비율을 같이 표현하여 확률적으로 어느 유형에 해당하는지를 알 수 있다.

본 논문의 구성은 다음과 같다. 2장에서는 관련연구에 대해 알아보고 3장에서는 본 논문에서 제안하고 있는 음식메뉴 사전 자동 구축을 설명한다. 4장에서는 실험을 통해 기존 음식메뉴 사전과 비교하고, 마지막으로 5장에서는 결론 및 향후 과제를 제시한다.

2. 관련 연구

일반적인 웹 크롤러는 자동적으로 웹서버를 순회하며 웹페이지의 내용을 분석하고, 그 안에 포함되어 있는 URL들을 추출한다. 그런 다음 그 URL들로 하나씩 이동하면서 웹 문서를 수집한다. 이렇게 수집된 다양하고 많은 양의 웹 문서는 검색 엔진에서 이용된다.

반면에 특정한 주제의 문서만을 수집할 수 있는 형태의 크롤러도 있다. Focused Crawler와 Topical Crawler는 크롤러 자체에 document classifier나 수집해야 할 문서의 규칙들을 탑재하여 수집을 한다.[3, 4] 그러나 분류기의 성능이 좋지 못하고 규칙의 양이 불충분하다면 사용자가 원하는 데이터를 수집할 수 없다.

이러한 이유로 인해 수집하고자 하는 특정 사이트의 구조를 분석하여 직접적으로 데이터를 수집해올 수 있는 wrapper에 대한 연구가 진행되었다. Wrapper를 이용한 크롤러는 필요한 정보를 추출하기 위해 규칙을 만들고 자동으로 사이트의 컨텐츠를 수집한다.[5,6,7,8,9,10]

전자통신연구원에서 구축한 한국어 개체명 사전은 음식메뉴명 정보만 제공할 뿐 음식 분류를 제공하지 않는다. 그리고 한국관광공사에서 구성한 음식메뉴명 사전은 음식 분류를 제공하지만 실제 분류와 다른 경우가 발생한다. 예를 들어 '알밥'은 일식으로 분류되어 있지만 리뷰들을 분석해보면 분식, 한식 등에서 더 많이 발견된다. 뿐만 아니라 온라인 리뷰에는 '갈비살'과 '소갈비살' 처럼 같은 음식메뉴인데 다르게 표현되는 경우도 있다.

또한 레스토랑 리뷰 문서의 음식메뉴 어휘들을 살펴보면 기존의 음식메뉴와 관련한 개체명 사전에 없는 경우가 있다. 이러한 이유는 개체명 사전은 일반명사들로 이루어져 있는데 반해 레스토랑에서는 신메뉴가 지속적으로 개발이 된다. 그럼에도 불구하고 개체명 사전을 수작업으로 구축하는 것은 많은 시간과 비용이 발생하기 때문에 업데이트가 지속적으로 이루어지지 않기 때문이다.

3. 레스토랑 메뉴 사전 자동 구축 알고리즘

표 1은 레스토랑 메뉴 사전 구축 알고리즘에 대한 설명이다.

<표 1> 레스토랑 메뉴 사전 구축을 위한 알고리즘

1. 래퍼 기반 크롤러를 이용해 레스토랑 전문 사이트에서 메뉴 정보 추출
2. 래퍼 기반 크롤러를 이용해 레시피 정보 제공 사이트에서 메뉴 정보 추출
3. 추출된 메뉴명의 빈도수 측정
4. Threshold 이상의 빈도수를 갖는 메뉴만 DB에 저장
5. 수집된 레스토랑 리뷰에서 음식메뉴 포함한 리뷰의 빈도수 측정
6. 음식메뉴 분류 정보에 따른 빈도수와 비율 DB에 업데이트
7. 레스토랑 리뷰에서 발견되지 않은 음식메뉴 제거

3.1 래퍼기반 메뉴 정보 추출

기존의 일반적인 웹 크롤러는 모든 문서를 대상으로 하기 때문에 성능이 좋지 못하다. 이러한 문제를 해결하기 위해 수집하고자하는 특정 사이트들의 구조를 분석해 직접적으로 원하는 데이터를 수집하는 래퍼(Wrapper) 기반 웹 크롤러 연구가 있다.

원하는 데이터를 정확하게 수집하는 방법은 해당 사이트의 html code를 이용하여 사이트의 구조를 분석하는 것이다. 그런 다음 원하는 데이터가 있는 부분만 접근할 수 있도록 해야 한다. 사이트의 구조 정보를 토대로 데이터를 수집할 수 있는 프로그램이 wrapper이다. 본 논문에서는 이러한 wrapper 모델을 기반으로 음식점 검색 사이트에서 음식 메뉴를 수집하였다.

레스토랑 전문 사이트는 일반적으로 공통된 구성으로 되어 있다. 레스토랑 전문 사이트는 다수의 레스토랑 s로 이루어져 있다.

$$S = \{s_1, s_2, \ldots, s_n\}$$

레스토랑 s는 레스토랑명, 전화번호, 주소, 추천메뉴 등과 같은 속성 정보 a로 구성되어 있다.

$$A_i = \{a_1, a_2, \ldots, a_o\}$$

레스토랑 S는 p개의 리뷰 r을 가지고 있다.

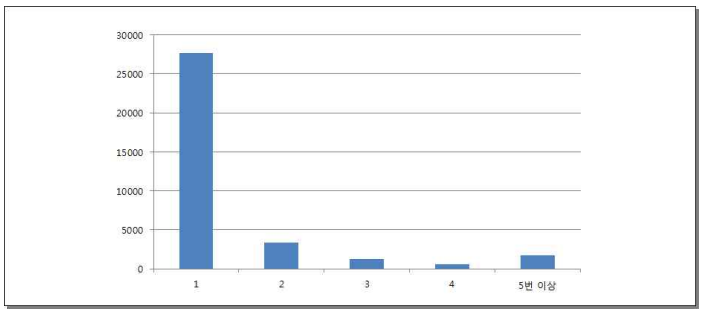
$$R_j = \{r_1, r_2, \ldots, r_p\}$$

메뉴는 주로 레스토랑 사이트 내의 추천메뉴, 인기메뉴, 메뉴판 등 정보의 위치와 구조 정보를 분석해보면 수집이 가능하다. 본 논문에서는 국내의 5개의 레스토랑 전문 사이트 중에서 메뉴 정보를 제공하는 2개의 사이트를 대상으로 수집하였다.

3.2 빈도수 기반 메뉴 선정

수집된 음식메뉴는 모두 34,506개였다. 이렇게 수집된

음식메뉴의 빈도수를 측정하고 빈도수 순으로 정렬하였다. 빈도수를 측정한 이유는 여러 레스토랑의 메뉴에서 빈번하게 나타난 음식 메뉴는 좀 더 일반적인 음식 메뉴일 확률이 높기 때문이다. 그림 1은 추출된 메뉴명의 빈도수를 나타내는 그림인데 빈도수가 1인 메뉴들이 전체의 대다수를 차지한다.



[그림 1] 추출된 메뉴명의 빈도수

수집된 음식 메뉴 중에서 빈도수가 3 이상의 음식 메뉴를 선정하였다. 표 2는 빈도수 순으로 정렬된 음식메뉴의 예이다.

<표 2> 빈도수 순으로 정렬한 상위 20개의 음식메뉴

순위	메뉴명	빈도수	순위	메뉴명	빈도수
1	돼지갈비	522	11	우럭	190
2	삼겹살	499	12	항정살	187
3	갈비탕	318	13	김치찌개	173
4	생삼겹살	300	14	차돌박이	167
5	냉면	285	15	생등심	185
6	갈비살	240	16	삼계탕	165
7	광어	230	17	된장찌개	164
8	회덮밥	221	18	등심	152
9	육회	213	19	생갈비	151
10	닭도리탕	198	20	갈매기살	149

이렇게 빈도수가 높은 음식 메뉴만을 선택한 이유는 빈도수가 낮은 음식 메뉴들은 대다수의 음식점에서 사용하는 일반적인 음식 메뉴가 아니었기 때문이다. 예를 들어 '메뉴1'을 A음식점에서 'A 음식점 메뉴1'이라 부른다고 하자. 두 음식 메뉴는 실제로는 같음에도 불구하고 '메뉴1'은 일반적인 음식 메뉴이기 때문에 빈도수가 높고 'A 음식점 메뉴1'은 A 음식점에서만 사용하기 때문에 빈도수가 낮다. 실제로 레스토랑에 따라 같은 메뉴에 레스토랑 상호를 붙이거나 새로운 이름으로 부르기도 한다. 그러나 온라인 리뷰를 분석한 결과 실제로 사용자들은 리뷰를 작성할 때 이러한 특이하거나 독특한 메뉴명은 잘 사용하지 않는다. 그렇기 때문에 빈도수가 낮은 음식메뉴명은 사전 목록에서 제외하도록 하였다.

3.3 음식메뉴의 분류 정보 추출

수집된 음식 메뉴 중에는 음료수와 술과 같이 음식 메뉴로 부적합한 것들이 있다. 이렇게 부적합한 음식 메뉴들을 제거하고 나머지를 음식 메뉴 사전에 등록한다. 이렇게 구축된 음식 메뉴 사전에도 확인결과 빠진 메뉴들이 존재했다. 이러한 문제를 해결하기 위해 레시피 제공 사이트를 크롤링해서 메뉴를 추가하였다.

음식메뉴는 한식, 양식, 중식과 같은 고유의 분류 정보를 가지고 있다. 그러나 일반적인 분류 체계와 실제 메뉴의 분류 정보가 다른 경우가 있다. 이러한 문제를 해결하기 위해 음식메뉴 사전의 엔티티들이 어느 유형의 레스토랑 리뷰에서 발견되는지를 찾아 빈도수를 구하고 분류 정보에 따른 비율을 사전에 추가하였다.

음식메뉴 M은 음식메뉴명, 기타/양식/일식/주점/중식/카페/한식 리뷰에서의 음식메뉴 빈도수, 전체 리뷰에서의 음식메뉴 빈도수, 기타/양식/일식/주점/중식/카페/한식 리뷰에서의 음식메뉴 비율로 이루어진다.

$$M = \{ food\_name, freq_{etc}, freq_{wst}, freq_{jpn}, freq_{bar}, freq_{chi}, freq_{caf}, freq_{kor}, freq_{tot}, rate_{etc}, rate_{wst}, rate_{jpn}, rate_{bar}, rate_{chi}, rate_{caf}, rate_{kor}, rate_{tot}, \}$$

아래의 표 4는 분류 정보 분석 결과를 포함한 음식 메뉴 사전 중 리뷰에서의 분류 정보에 따른 비율의 예를 보여주고 있다. 이 분류 정보를 이용하면 정보 추출을 통해 음식 메뉴를 추출할 경우 부가적으로 분류 정보도 얻을 수 있다. 표 4의 '알밥'의 경우 일식과 한식에서 비슷하게 분포되어 있는 것을 볼 수 있는데 이렇게 여러 분류에 해당되는 음식메뉴도 이 방법을 이용해 구분할 수 있다.

<표 3> 카테고리 분석 정보가 포함된 음식메뉴 사전의 예

구분	기타 리뷰 비율	양식 리뷰 비율	일식 리뷰 비율	주점 리뷰 비율	중식 리뷰 비율	카페 리뷰 비율	한식 리뷰 비율
갈비탕	0.0	0.7	0.7	0.7	0.3	0.7	97.0
미소라멘	0.0	1.1	85.1	11.7	0.0	0.0	2.1
모듬회	0.0	0.0	20.0	28.0	0.0	0.0	52.0
안창살	0.0	2.1	0.0	2.1	0.0	0.0	95.7
매운탕	0.0	0.0	4.6	37.1	1.7	2.2	54.5
낙지볶음	0.0	2.4	0.8	1.6	0.0	3.3	91.9
해물탕	1.3	8.5	1.3	19.6	19.0	2.6	47.7
알밥	1.0	0.5	43.4	4.0	0.0	2.0	49.0
감자탕	0.3	0.9	1.5	0.3	0.6	0.3	96.0

4. 실험

실험을 위해 레스토랑 온라인 리뷰를 래퍼 기반 웹 크롤러를 이용해 수집했다. 크롤링 된 리뷰 중 1,000개의 리뷰를 뽑아 다시 문장 단위로 나누었다. 수작업으로 분류를 한 결과 음식 메뉴를 포함한 리뷰 문장이 1,129개였고 음식 메뉴를 포함하지 않은 리뷰 문장은 2,393개였다.

음식 메뉴는 품사 중에서 명사에 해당되기 때문에 분

류한 리뷰 문장을 형태소 분석한다. 그런 다음 명사만 추출하고 음식 메뉴 사전과 비교하여 일치하는 음식메뉴를 추출하여 리뷰 문장의 음식메뉴를 얼마나 정확히 추출하는지를 실험하였다. 그 결과 True Positive는 778, False Positive는 0, False Negative는 351, True Negative는 2,393이었다. <표 5>는 리뷰를 형태소 분석한 다음 명사를 음식메뉴 사전과 비교해 성능을 평가한 결과이다.

<표 4> 온라인 레스토랑 리뷰의 형태소 분석결과를 이용한 사전 성능 평가

추출 결과	실제 Class		Total
	Positive	Negative	
Positive	778	0	778
Negative	351	2,393	2,744
Total	1,129	2,393	3,522

온라인 리뷰의 특성상 띄어쓰기가 제대로 되어 있지 않은 문장이 많기 때문에 추출이 되지 않는 문장들이 있었다. 이러한 문제점을 해결하기 위해 형태소 분석을 하지 않고 문장의 공백을 제거한 다음 Java의 indexOf() 메소드를 이용해 문자열 비교를 해 음식메뉴 사전에 있는 엔티티를 포함하는 문자열을 뽑도록 하였다. 이 방법은 포함 관계인 음식 메뉴가 모두 추출되는 문제가 있다. 예를 들어 '바지락칼국수'는 '바지락칼국수', '칼국수', '국수'가 모두 추출된다. 이러한 문제점을 막기 위해 여러 개의 음식 메뉴가 추출되는 경우 가장 문자열이 긴 음식 메뉴를 추출하도록 한다.

이 방법으로 온라인 레스토랑 리뷰 문장의 음식메뉴를 얼마나 정확히 추출하는지를 실험하였다. 그 결과 True Positive는 884, False Positive는 35, False Negative는 245, True Negative는 2,358이었다. <표 6>은 리뷰를 음식메뉴 사전과 문자열 비교를 하여 성능을 평가한 결과이다.

<표 5> 온라인 레스토랑 리뷰의 문자열 비교를 통한 사전 성능 평가

추출 결과	실제 Class		Total
	Positive	Negative	
Positive	884	35	919
Negative	245	2,358	2,603
Total	1,129	2,393	3,522

좀 더 쉽게 두 가지 방법을 비교하기 위해 F-Score를 구하였는데 형태소 분석기를 이용한 방법은 81.6이었고 문자열 비교를 한 방법은 86.4로 상대적으로 나은 성능을 보였다.

그러나 문자열 비교를 이용한 음식 메뉴 추출 방법은 '죽', '난', '전', 떡' 등과 같이 음식 메뉴명의 문자열 길이가 짧은 경우에는 잘못된 음식메뉴명을 추출하기도 한다. 또한 '라면', '파이', '스프' 등도 잘못된 음식 메뉴명을 추출하는 주원인이었다. <표 7>은 문자열 비교를 통해 음식 메뉴명을 잘못 추출하는 경우의 예이다.

<표 6> 잘못 추출된 음식 메뉴명의 예

'고향의 맛이라 생각하고 먹을수도 있겠지만 그게아니**라면** 먹기가 좀 부담스럽습니다.'
'와이**파이**가 잘 터지네요.'
'이 집 에**스프**레소가 맛있네요.'

그 밖에도 외국 음식명은 외래어 표기 방법에 따라 다양하게 표현되어 추출의 성능을 나쁘게 하였다. 예를 들어 'cake'는 '케잌', '케이크', '케익' 등 레스토랑이나 리뷰어에 따라 다르게 표현된다.

5. 결론

본 논문에서는 레스토랑 전문 사이트와 레시피 제공 사이트에서 각 레스토랑의 메뉴 정보와 음식명 등을 웹 크롤러로 수집하였다. 그런 다음 빈도수가 낮은 음식메뉴와 레스토랑 온라인 리뷰에서 쓰이지 않는 음식메뉴를 제거하여 레스토랑 음식 메뉴 사전을 자동으로 구축하였다. 그리고 레스토랑 온라인 리뷰 문서를 이용해 음식메뉴 사전의 엔티티들이 어느 유형의 레스토랑 리뷰에서 발견되는지를 찾아 빈도수를 구하고 분류 정보에 따른 비율을 사전에 추가하였다. 이 정보를 이용해 여러 분류 유형에 해당되는 음식메뉴를 구분할 수 있다.

성능 측정을 위해 구축된 음식 메뉴 사전을 이용해 레스토랑 리뷰로부터 음식 메뉴를 추출하였다. 추출 방법은 형태소 분석 방법과 문자열 비교 방법 두 가지를 비교하였는데 실험 결과 형태소 분석 방법의 정확률은 100, 재현율은 68.9, F-Score는 81.6이었고 문자열 비교 방법의 정확률은 96.2, 재현율은 78.4, F-Score는 86.4로 상대적으로 문자열 비교 방법의 성능이 더 좋았다.

이렇게 구축된 음식 메뉴 사전으로 레스토랑 리뷰 분석을 통한 인기메뉴 추출이나 의견대상에 대한 의견 검출 등을 할 수 있다.

참고문헌

[1] 전자통신연구원 한국어 어휘 사전 DB, http://www.itec.re.kr/itec/sub02/sub02_01_1.do?t_id=5674#
[2] 한국관광공사 관광용어 외국어 용례사전, http://kto.visitkorea.or.kr/kor/translation/translation/main.kto
[3] S. Chakrabarti, M. van den Berg, and B. Dom, "Focused Crawling : A New Approach to Topic-Specific Web Resource Discovery", Computer Networks, Vol.31, No. 11-16, pp.1623-1640, 1999
[4] Ziyu Guan, Can Wang, Chun Chen, Jiajun Bu, Junfeng Wang, "Guide Focused Crawler Efficiently and Effectively Using On-line

Topical Importance Estimation", In Proc. of ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 757-758, 2008
[5] S. Chakrabarti, Mining the Web. Discovering Knowledge from Hypertext Data, Morgan Kaufmamm, pp. 257-287, 2003
[6] J. Cho, H. Garcia-Molina, and L. Page, “Efficient Crawling through URL Ordering”, Computer Networks, Vol.30, No.1-7, pp. 161-172, 1998
[7] Chia-Hui Chang, Mohammed Kayed, Moheb Ramzy Girgis, and Khaled Shaalan, “A Survey of Web Information Extraction Systems”, IEEE Transaction on Knowledge and Data Engineering, Vol.18, No. 10, pp.1411-1428, 2006
[8] Jaeyoung Yang, Tae-Hyung Kim, and Joongmin Choi, “An Interface Agent for Wrapper-Based Information Extraction”, In Proc. of the International Confenrence on Principles of Practice in Multi-Agent Systems(PRIMA'04), pp.291-302, 2004
[9] Claudio Bertoli, Valter Crescenzi, and Paolo Merialdo, “Crawling programs for wrapper-based applications”, In Proc. of IEEE International Conference on Information Reuse and Integration(IRI'08), pp.160-165, 2008
[10] Stephen Soderland, Claire Cardie, and Raymond Mooney, "Learning information extraction rules for semi-structured and Free Text", Machine Learning, Vol. 34, No.1-3, pp.233-272, 1999

주식 관련 기사 분류 및 긍정 부정 판단을 통한 종목 추천 시스템

이유준○, 박정우, 전민재, 최준수, 한광수
국민대학교 컴퓨터공학부
yj_lee@kookmin.ac.kr, jungwooc@kookmin.ac.kr, mj_jeon@kookmin.ac.kr,
jschoi@kookmin.ac.kr, kshahn@kookmin.ac.kr

Stocks Recommending System through Classifying News Articles by Positive or Negative Decision

Yoojun Lee○, Jungwoo Park, Minjae Jeon, Joonsoo Choi, Kwangsoo Hahn
Kookmin University

요 약

주식 시장에서 거래되고 있는 증권은 MACD(Moving Average Convergence Divergence), Stochastic 등의 보조 지표를 이용하는 기술적 분석을 통하여 매수/매도 시점을 결정한다. 주식 시장의 객관적인 자료를 통하여 분석하는 기술적 분석 방법은 주식 시장 외적인 요소를 반영하는데 있어 한계점이 존재한다. 본 논문에서는 기술적 분석 방법에 기사를 종목별로 분류하고 기사의 긍정 및 부정을 판별하는 문서 분류 기법을 적용하여 주식 외적인 요소를 반영하는 시스템을 제안한다.

주제어: Genetic algorithm, Stocks recommendation, Opinion mining, Document classification

1. 서론

주식 투자에 있어 기술적 분석은 과거의 주가와 거래량을 이용해 주식의 추세를 분석하여 미래의 주가를 예측하는 방법이다[1]. 그러나 기술적 분석을 통한 방법은 완벽히 신뢰할 수 있는 방법은 아니다. 가령 특정 회사에서 새로운 기술을 개발했거나 국가적 위기 상황과 같은 주식 시장 외적인 요소들은 기술적 분석 방법만으로 반영 할 수 없다. 최근 인터넷의 보급화로 인해 모든 소식들을 인터넷 기사로 접할 수 있게 되었다. 이러한 기사들은 신문사나 기자에 따라 주관적인 특성을 가지고 있다. 이런 기사들은 주식 시장에 직접적 혹은 간접적으로 영향을 준다.

현대 세계에서는 엄청난 양의 정보들을 생성하고 있다. 응용 분야의 광범위한 범위에서 방대한 규모의 데이터들이 계속해서 수집되고 있는데, IBM에서는 매일 2.5 퀸틸리언 바이트(2.5 quintillion byte, 2.5×10^{18} byte)의 데이터가 발생함을 언급하고 있다[2]. 이처럼 인터넷 공간에는 사람이 처리할 수 없을 정도로 많은 기사들이 존재한다. 이와 같이 대용량 데이터 처리의 중요성이 부각되면서 대용량 데이터의 효율적인 저장, 관리, 검색, 분석 및 활용을 위한 연구가 활발히 이루어지고 있는 실정이다.

본 논문에서는 뉴스 기사에서 관심있는 종목에 대한 기사를 선별하여 해당 기사가 주가에 미치는 것을 주식 추천 시스템에 반영하고자 한다. 기술적 분석 방법을 통해 객관적인 데이터를 기반으로 하여 추천 종목의 신뢰성을 높이고, 기사들을 각 종목별로 분류하여 기사의 긍정 및 부정을 판단하는 방법을 통해 주관적인 데이터를 확보하여 주식 시장 외적인 요소를 반영하는 증권의 매수/매도 종목 추천 시스템을 제안한다.

2. 관련 연구

2.1 기술적 분석과 유전 알고리즘

본 연구에서는 MACD 지표를 사용하여 기술적 분석을 한다. MACD 지표를 이용하여 유전 알고리즘을 구성하는 방법과 이를 통해 종목을 추천하는 방향을 제시한다.

2.1.1 기술적 분석(Technical analysis)

기술적 분석에서 많이 사용되고 있는 지표인 Slow stochastic과 MACD는 실무에서 많이 사용되고 있으며 신뢰도가 높은 편이다.

MACD는 단기 EMA(Exponential Moving Average, 지수이동평균)와 장기 EMA의 차이이다. 보편적으로 단기 EMA의 기간은 12일을 사용하고, 장기 EMA의 기간은 26일을 사용한다. MACD signal은 k(보편적으로 9일)일간의 EMA를 나타낸다. 또한 MACD oscillator는 MACD와 MACD signal과의 차이를 나타낸다.

MACD Line	단기 EMA - 장기 EMA
MACD Signal	k 기간의 EMA
MACD Oscillator	MACD - MACD Signal

표 1 MACD 계산 공식

본 연구에서는 MACD 지표를 계산하는데 사용되는 기간을 보편적으로 사용하는 기간으로 고정하지 않고, 각 종목에 적합한 날짜를 직접 구하여 계산하고자 한다. MACD 지표에서는 단기, 장기, k 총 3개의 기간이 필요하다.

2.1.2 유전 알고리즘

유전 알고리즘은 진화의 원리를 문제 해결에 이용하는 방법론이다. 다윈의 진화론과 같이 다산, 생존 경쟁, 변이, 자연 선택, 진화의 과정을 반복하여 진행하게 된다. 유전 알고리즘은 NP-hard 문제에 있어 효과적이다. 하지만 문제 해결을 위한 효율적인 알고리즘이 존재하는 경우는 비효율적인 방법이 될 수도 있다. 보편적인 유전 알고리즘은 하나의 해를 염색체라고 표현하고 복수개의 염색체를 운영하여 최적의 해를 구한다. 일반적인 유전 알고리즘은 아래와 같은 방법으로 진행된다[3].

최초 난수를 발생시켜 n개의 유전자인 해집단을 생성하고, 해 집단의 적합도를 평가하게 된다. 적합도란 각각의 유전자를 이용해 MACD를 구하고 지표가 제공하는 매매시점을 이용해 특정 기간 동안 모의 투자를 하여 수익률을 계산한다. 적합도를 평가한 후 품질 비례 선택과 엘리트 보전 전략을 이용해 교배 할 두 개체를 선택하게 된다. 이렇게 선택 된 두 개체를 교차 연산과 변이 연산을 통해 새로운 두 개체를 생성하게 된다. 생성 된 개체가 기존 해집단의 개체보다 적합도가 높게 되면 대치 연산을 통해 개체를 대치하게 된다. 위와 같은 과정을 반복하여 최적의 해를 찾게 된다[4].

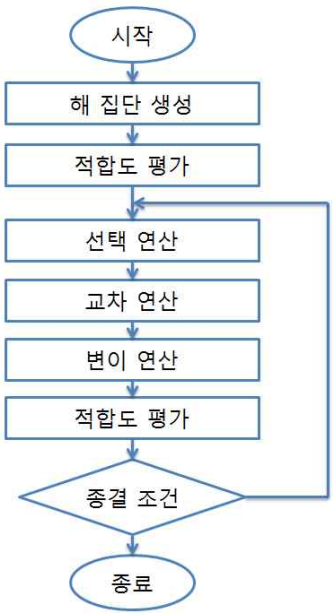


그림 1 유전 알고리즘 흐름도

단기 기간, 장기 기간, k 기간을 각각 8bit로 표현하여 하나의 24bit 유전자로 표현이 된다. 유전자를 이용하여 날짜를 변수로 한 최적의 MACD 지표를 구하게 된다.

x1 (단기)	1 <= x1 <= 256
x2 (장기)	1 <= x2 <= 256
x3 (k)	1 <= x3 <= 256

표 2 유전자 표기 방식

2.2 문서 분류

인터넷의 보급화와 사용 증가에 따라 웹상의 전자 문서의 양이 기하급수적으로 증가하고 있다. 전자 문서가 양적으로 크게 늘어남에 따라 이러한 수많은 문서를 사람이 분류하는 것은 거의 불가능해 졌다. 이에 따라 문서를 알맞게 분류하는 것을 도와주는 도구에 대한 요구가 점차 중요해 지고 있다.

문서 분류란 미리 정의되어 있는 두 개 이상의 범주(Category)에 대하여, 새로운 문서 집단이 입력되었을 때 미리 학습된 범주와 입력 문서 간의 유사도 비교를 통해 입력된 문서에 대한 범주를 결정해주는 기법이다[5]. 이러한 문서 분류는 다양한 분야에 적용되어 활용되고 있다. 스팸 메일 필터링, 뉴스기사 필터링, 뉴스기사의 주제 선정, 웹 문헌 분류 등 여러 분야에서 활발하게 연구되어지고 있다.

2.3 오피니언 마이닝

오피니언 마이닝(Opinion mining)은 감정 분석(Sentiment analysis)과 비슷한 의미로 사용된다. 텍스트에 포함된 내용이 주관적(Subjective)인지 객관적(Objective)인지 먼저 판별하고, 주관적이면 극성(Polarity)을 분석하여 내용이 긍정적(Positive)인지 부정적(Negative)인지 판별한다. 이러한 오피니언 마이닝은 대량의 정보 속에서 유용한 정보를 찾아내는 특징을 가지고 있다. 오피니언 마이닝은 특징(Feature) 추출, 의견 분류, 요약 및 표현 3가지 단계로 이루어진다.

특징 추출 단계에서는 중요한 정보로 판단되는 특징들을 추출해 낸다. 해당 특징의 어휘 정보도 추출된다. 의견 분류 단계에서는 추출된 특징과 어휘가 어떤 의미로 사용되었는지 판단 및 분류한다. 요약 및 표현 단계에서는 의견 성향이 밝혀진 의견정보들을 요약하여 전달하는 단계이다[6]. 이와 같은 오피니언 마이닝의 단계별 수행을 위해 다양한 방식으로 연구되고 있으며 자연어처리 기술 기반에서 통계적 기법에 이르기까지 여러 분야의 기술이 접목되고 있다. 기존에 긍정 및 부정을 판별하는 방법은 다음과 같다.

첫째로 SVMs(Support Vector Machines) 방법이 있다. 이는 미리 사전에 긍정 및 부정으로 분류된 학습 데이터(Training Sets)로 텍스트의 긍정 및 부정 의견을 분류하는 방식이다. 둘째로 N-grams or Part of Speech 방법이 있다. N-grams 단어 구조로 긍정 및 부정을 찾는 방식으로 문장을 Bi-gram decomposition으로 분리한다. 셋째로 Lexicon-based Approach 방법이 있다. 사전에 미리 정의된 긍정 및 부정 bag of words(1-grams or Uni-grams)를 이용하여 텍스트에 포함된 긍정 및 부정 단어의 출현 빈도로 긍정과 부정을 판별하는 기법이다. LIWC(Linguistic Inquiry and Word Count)나

POMS(Profile of Mood States) 같은 사전을 이용할 수 있는데 한국어에서는 사용할 수 없다. 이 방법에서는 긍정/부정 코퍼스(Corpus)를 만드는 것이 관건이다. 넷째로 Linguistic Approach 방법이 있다. 텍스트의 문법적인 구조를 파악하여 극성을 판별하는 기법이다. 주로 Lexicon-based Approach 방식과 함께 사용한다. 문맥(Context) 등을 파악하여 극성을 파악한다.

오피니언은 어떤 단위로 볼 것인가에 따라 3가지로 분류될 수 있다. 문서 전체적으로 의견을 종합하는 document level, 문서를 문장 단위로 나누어서 개별 문장의 의미를 파악하는 sentence level, 문장 내부의 어떤 개체의 속성에 대한 의견을 파악하는 entity and feature/aspect level로 분류 할 수 있다.

3. 기사 분류 및 분석 시스템

기술적 분석은 객관적인 주식 데이터를 기반으로 분석하기 때문에 주식 시장 외적인 요소들을 반영하는데 한계점이 있다. 하지만 실제 주식 시장에서는 해당 주식 종목에 대한 기사로 인해 주가에 영향을 미치게 된다. 본 논문에서 제안하는 기사 분석 시스템은 이러한 단점을 보완하여 안정적인 주식 종목 추천을 목표로 한다. 전체 시스템은 아래와 같다.

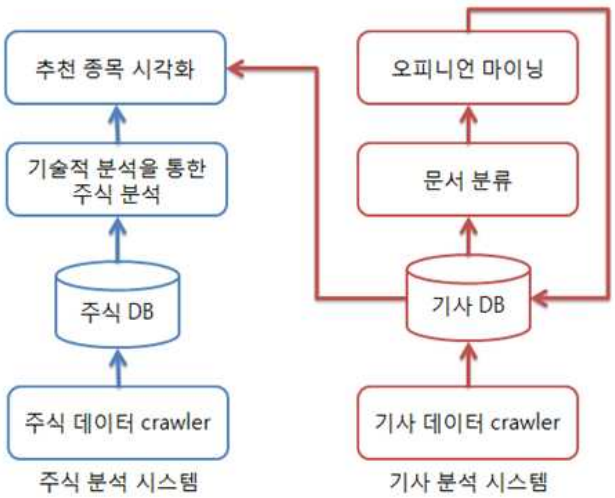


그림 2. 추천 종목 시스템의 구조

기사 분석 시스템은 실시간으로 신문 기사를 crawling하고 이렇게 수집된 기사는 데이터베이스에 저장 되고, 문서 분류 컴포넌트에 입력된다. 주식 시장과 관련된 기사이기 때문에 다음과 같은 형식으로 문서가 분류된다. 기사 분류 시스템은 주식 종목당 2개의 카테고리로 분류가 된다. 하나는 주식 시장에 영향을 미치지 않는 기사 카테고리, 나머지 하나는 주식 시장에 영향을 미치는 기사 카테고리이다. 시스템 운용자가 처음 두 개의 카테고리 별로 기사를 학습시키고, 시스템에 입력되는 기사에 대해 분류를 하고 분류된 기사는 학습 된 데이터에 추가한다. 이렇게 분류되어 학습된 기사 데이터가 많이 쌓이게 되면 분류의 정확도는 높아지게 된다.

문서 분류 단계가 끝나면 오피니언 마이닝을 통하여 기사의 성향을 분석한다. 해당 기사가 주식 시장에 반영되었을 경우 그 영향력을 분석하는 단계이다. 크게 긍정적, 부정적, 중립의 3가지 성향이 존재하게 된다. 오피니언 마이닝에서는 단어에 가중치를 두어 문장에 대한 점수를 계산하는 방법을 사용한다. 시스템 운용자가 단어 사전을 만들고 각 term들에 대한 가중치를 작성하게 된다. 각 term들에 부여된 가중치는 성향을 나타내게 되는데 부정의 부정, 긍정의 부정, 부정의 긍정 등 성향이 반대가 되는 경우를 위하여 긍정은 '+' 부호를 사용하고 부정은 '-' 부호를 사용하게 된다. 부호의 곱을 통하여 위와 같은 경우도 정확한 분류가 가능하게 된다.

문장에 대한 점수를 계산하기 위해서는 형태소 분석을 통해 문장 구조를 분석하고 각각의 관련된 문장 성분간의 가중치를 더하고 부호를 곱하는 방식으로 성향 분석을 진행한다. 기사에 대한 성향이 결정되면 기존의 주식 분석 시스템에 해당 기사에 대한 점수를 반영하게 된다.

4. 결론

본 논문은 유전 알고리즘과 기술적 분석을 사용하여 국내 주식 시장에서 매수와 매도를 추천하는 종목 추출 방법과 주식 시장 외적인 요소를 반영하는 방법에 대해 다루었다. 주식 시장에 영향을 미칠 수 있는 기사들을 문서 분류 단계에서 추출하여 신뢰성을 높이고, 매수 및 매도를 추천하는 종목에 기사의 성향을 분석하여 계산된 가중치를 적용한다. 따라서 기술적 분석 방법을 통해 구해진 신뢰도가 높은 데이터에 주식 시장 외적인 요소를 반영하여 더욱 안정적인 종목 추천을 제공하는 시스템으로서 긍정적인 가능성이 있음을 확인할 수 있었다. 뉴스 기사와 같은 매체는 주관적인 성격을 가지고 있기 때문에 기술적 분석으로 얻어지는 객관적인 데이터보다는 신뢰성이 낮은 편이다. 향후 기사 분석을 통해 나온 가중치를 주식 분석 시스템에 적용할 때의 적정 비율을 연구하여 시스템의 안정성을 증대시킬 것이다.

참고문헌

[1] Edwards, Robert D., Magee, John, Bassetti, "Technical Analysis of Stock Trends", American Management Association, pp.4-7, 1998.

[2] 양혜영, 빅데이터를 활용한 기술기획 방법론, 한국과학기술기획평가원, 제14호, 2012.

[3] Goldberg, D.E, Genetic Algorithm in Search, Optimization, and Machine Learning, Addison-Wesley, pp.10-17, 1989.

[4] 문병로, 쉽게 배우는 유전 알고리즘, 한빛미디어, pp.59-81, 2008.

[5] Fabrizio Sebastiani, "Machine Learning in Automated Text Categorization", ACM Computing Surveys, Vol.34 ,No.1, pp.1-47, March 2002.

[6] 양정현, 명재석, 이상구, "상품 리뷰 요약에서의 문맥 정보를 이용한 의견 분류 방법", 정보과학회 논문지, 제36권, 제4호, pp.254-262, 2009.

학생 답안 정보를 활용한 반자동 정답 템플릿 구축 도구

장은서○, 강승식
국민대학교 컴퓨터공학부
{akdangz, sskang}@kookmin.ac.kr

Semi-automatic Answer Template Constructor using Student Responses

Eun-Seo Jang○, Seung-Shik Kang
Kookmin University, School of Computer Science

요 약

대학수학능력시험이나 학업성취도평가와 같은 중요한 시험은 다단계의 복잡한 과정을 거쳐 채점되며, 대규모의 인적·물적 자원이 투입된다. 컴퓨터 시스템을 활용하여 채점 과정에 소모되는 자원을 절감하고 정확성을 향상시키는 등 채점 효율의 향상을 목표로 하는 연구가 진행 중에 있다. 하지만 채점하려는 문항에 대한 정보 없이 시스템이 자동으로 정·오답을 판단하는 것은 불가능하며, 채점자가 문항에 대한 정보를 일정한 형식으로 가공하여 시스템에 제공하여야 한다. 본 논문은 채점하려는 학생 답안을 정보로 이용하여 시스템이 인지할 수 있는 정답 템플릿 형식으로 구축하는 방법을 소개한다. 또한 소개한 방법으로 작성된 정답 템플릿을 이용하여 자동 채점을 수행하였을 때의 채점 결과 및 통계 자료를 수록하였다.

주제어: 자동 채점, 정답 템플릿, 한글 처리, 자연어 처리

1. 서론

자연어 처리 기술을 이용하여 특정 문항의 학생 답안들을 자동으로 채점하는 방법은 관련 학계의 주요 연구 주제 중 하나이다. 특히 학업성취도평가 또는 대학수학능력시험과 같은 시험은 학생의 진로를 결정하는 중요한 시험이며, 학생 뿐 아니라 학부모 및 교사 등의 관심이 집중된다. 그렇기 때문에 학생 답안의 채점 또한 복잡한 다단계의 과정을 거쳐 신중하게 진행되며 이를 위해 대규모의 인적·물적 자원이 투입된다. 이러한 채점 과정의 일부를 자동화된 컴퓨터 시스템으로 대체하여 비용의 절감과 함께 채점의 정확도와 같은 효율을 향상시키는 방법이 연구 중에 있다.[1]

하지만 채점 시스템이 문제를 자동으로 인식하여 학생 답안의 정·오답 여부를 가리는 것은 현실적으로 불가능하며, 채점자가 문항을 채점하는 데 필요한 정보를 시스템이 인식할 수 있는 형식으로 가공하여 제공하여야 한다. 하지만 채점자는 채점 대상 과목의 전문가이지만 컴퓨터 시스템을 다루는 데에는 미숙할 수 있으며 시스템이 인식할 수 있는 형식의 정답 템플릿을 작성하는 데 불필요한 시간이 소모될 수 있다.

본 논문은 채점하려는 학생 답안을 먼저 분석하여 문항 채점에 필수적인 정보와 채점자가 부가적으로 작성한 채점 옵션, 모범 답안 정보, 항목 별 부여 점수 등의 정보들을 포함하는 완성된 채점 정보를 시스템이 인식 가능한 형식의 정답 템플릿 파일로 자동 구축해 주는 방법을 소개한다. 또한 소개한 방법을 이용하여 9개 문항에 대한 정답 템플릿을 생성하는 실험을 진행하였으며 이에 대한 채점 결과 및 통계 자료들을 수록하였다.

2. 관련 연구

2.1 자동 채점 시스템

자동 채점 시스템이란 특정 문항에 대한 학생 답안들을 각각 검사하여 점수를 부여하는 작업을 자동으로 수행하는 시스템이다. 그러나 채점 시스템이 문항을 분석 및 인식하여 학생 답안의 정·오답 여부를 가리는 것은 불가능하며, 채점자가 문항의 채점 기준들을 시스템에 제공해야 한다. 이러한 문항 정보는 자동 채점 시스템마다 상이하며 본 논문에서는 현재 개발 중인 KASS 2.0 자동채점 시스템(가제)에 적용할 수 있는 형태의 문항 정보 형식인 정답 템플릿을 반자동으로 구축할 수 있는 방법을 소개한다.

2.2 정답 템플릿

KASS 2.0의 정답 템플릿은 채점 옵션 정보, 모범 답안 정보, 고빈도 답안 수동 채점 정보, 개념기반 채점 정보의 4가지 기본 정보와 채점자의 선택에 따라서 추가적으로 단서어 정보를 저장하는 답안 정보 파일이다.[2]

채점 옵션 정보는 각 학생 답안의 전처리 옵션들을 저장한다. 선택 가능한 옵션으로는 특수 기호 제거, 띄어쓰기 오류 교정, 맞춤법 오류 교정, 불용어 제거가 있다. 선택한 전처리 옵션은 정답 템플릿을 만드는 과정뿐 아니라 실제 채점이 될 때에도 적용되며 각 문항의 특성에 따라 알맞은 옵션을 부여하여야 한다.

모범 답안 정보는 문항의 예시 정답이라고 할 수 있는 하나 이상의 문자열 집합이며, 모범 답안 점수는 문항에서 받을 수 있는 최고점을 의미한다. 고빈도 답안 정보는 특정 고빈도 답안 유형이 출현했을 때 부여할 점수 정보를 포함한다. 학생 답안들 중에서 정답으로 인정할 수 있는 핵심 개념을 공통적으로 포함하는 다양한 유형을 개념 정보로 만들어 처리할 수 있으며 개념기반 채점 정보는 이러한 개념 정보들을 저장한다. 단서어 목록은 정답 작성을 위해 필수적으로 포함되어야 하는 단어들의 목록이며 답안에 단서어 목록의 단어가 하나도 등장하지 않는 경우 해당 답안을 오답 처리한다.

3. 정답 템플릿 반자동 구축 도구

3.1 학생 답안 정규화

학생 답안을 분석하는 작업에 앞서 문항의 채점 요건에 맞게 학생 답안을 정규화하는 작업이 선행되어야 한다. 맞춤법, 띄어쓰기 또는 구두점 사용의 오류를 검사해야 하는 문항이 있는 반면, 그렇지 않은 문항도 존재한다. 채점자가 문항의 특성에 맞게 정규화 옵션을 설정하고 학생 답안을 불러오도록 하면 시스템은 불러온 학생 답안을 대상으로 정규화 작업을 수행한다. 표 1은 학생 답안을 정규화하는 예제를 보여준다.

표 1 학생 답안 정규화 예제

모범 답안	다만 그 성음을 바탕으로 이치를 다한 것 뿐이다.
학생 답안	Ⓐ: 다만 그 성음을 바탕으로 이치를 다한것뿐이다.
특수기호 제거	다만 그 성음을 바탕으로 이치를 다한것뿐이다
띄어쓰기 교정	다만 그 성음을 바탕으로 이치를 다한 것 뿐이다

정규화가 완료된 학생 답안들은 유형 별로 빈도를 조사한 다음 빈도 순으로 내림차순 정렬되어 채점자에게 보여진다. 학생 답안 유형 목록은 유형 별 빈도와 누적 빈도를 제공하며, 채점자는 이 목록을 이용하여 문항의 채점을 위한 부가 정보를 기입한다.

3.2 부가 정보 입력

채점자가 입력 가능한 부가 정보는 모범 답안 목록 및 점수, 고빈도 답안 부여 점수, 개념 생성 대상 및 개념 출현 시 부여 점수, 단서어 목록이 있다.

채점자는 문항 별로 제공되는 채점 기준을 이용하여 모범 답안 목록과 모범 답안 점수를 부여하고 고빈도 답안 유형 중에서 선택적으로 수작업 채점을 수행한다. 누적빈도 정보를 참고하여 어느 정도의 학생 답안이 채점될 수 있는지 확인할 수 있다. 학생 답안 유형 중에서 개념 정보를 생성할 필요가 있는 유형을 선택 및 생성하고 필요에 따라 단서어 목록을 작성한다.

표 2는 정답 템플릿 구축 도구가 학생 답안을 분석 및 제공하는 답안 유형 목록을 검토하여 채점에 필요한 정보를 작성한 예시를 보여준다. 표 2의 예제를 통해 ans_2~ans_5를 수작업 채점하였으며 ans_2, ans_4, ans_5를 이용하여 개념을 생성하는 것을 알 수 있다.

표 2 부가 정보 기입 예시

개념 생성	답안 유형	빈도	누적 빈도	점수
	ans_1	f_1	$\sum_{i=1}^{1} f_i$	
√	ans_2	f_2	$\sum_{i=1}^{2} f_i$	s_2
	ans_3	f_3	$\sum_{i=1}^{3} f_i$	s_3
√	ans_4	f_4	$\sum_{i=1}^{4} f_i$	s_4
√	ans_5	f_5	$\sum_{i=1}^{5} f_i$	s_5

3.3 정답 템플릿 구축

정답 템플릿은 json 형식으로 구조화하였고 정규화 옵션, 모범 답안 정보, 고빈도 답안 수작업 채점 정보, 개념 정보와 채점자의 선택에 따라 단서어 정보가 저장된다. 채점자가 부가 정보의 입력을 모두 마친 다음 정답 템플릿 구축 명령을 내리면 시스템은 학생 답안 분석 결과와 입력된 부가 정보를 결합하여 정답 템플릿을 작성하게 된다.

모범 답안과 고빈도 답안 채점은 문자열 완전 일치 방법을 이용하기 때문에 각 목록의 문자열들을 정답 템플릿에 저장하고, 각각 부여된 점수를 저장한다. 단서어 또한 문자열 정보만 필요하기 때문에 채점자가 기입한 문자열 집합을 그대로 파일에 저장한다. 개념은 생성 대상인 답안 유형의 문자열을 형태소 분석하여 각 어절을 어휘 토큰과 문법 토큰으로 분리하여 저장한다.

4. 실험 및 결과

4.1 실험 환경

자동 채점을 수행하기 위하여 KASS 2.0 을 이용하고 정답 템플릿은 자동 채점 시스템이 인식할 수 있는 형식으로 저장한다. 정답 템플릿 생성 및 자동 채점 실험을 수행하기 위해 9개 문항의 학생 답안들을 이용했다. Q_1~Q_5는 1개 어절로 정답을 서술할 수 있는 문항이고, Q_6~Q_9는 1개 문장으로 정답을 서술할 수 있는 문항이다. 정답 템플릿은 개발된 반자동 생성 도구에 의해 만들어진 그대로 별도의 수정 없이 사용하였다.

4.2 실험 결과 및 평가

동일 기준으로 생성했을 때의 문항 별 차이를 알아보기 위해 1개의 모범 답안을 입력하였고 빈도 10 이상인 답안 유형을 수작업 채점하였으며 빈도 3~9인 답안 유형을 개념으로 생성하였다. 표 3은 위의 기준을 이용하여 정답 템플릿을 생성한 결과이다.

표 3 정답 템플릿 생성 결과

문항	개수			
	전체	모범답안	고빈도답안	개념
Q_1	3013	1	6	6
Q_2	3018	1	10	11
Q_3	3013	1	10	8
Q_4	3023	1	8	4
Q_5	3035	1	4	4
Q_6	3013	1	20	35
Q_7	3012	1	5	7
Q_8	3010	1	17	23
Q_9	3010	1	27	25

표 4는 생성한 정답 템플릿을 이용하여 실제 자동 채점을 수행한 결과이다. 실험 결과, 비교적 간단한 문항인 Q_1~Q_5 문항의 경우 학생 답안의 대부분이 모범 답안과 고빈도 답안 정보를 이용한 완전일치 채점에서 처리됨을 확인하였다. 1개 문장으로 이루어진 Q_6~Q_9 문항의 경우에는 Q_1~Q_5 문항보다 개념 기반의 채점 빈도가 늘어났으며

미판단 답안의 비율 또한 약간 증가하였다. 미판단 답안의 비율은 Q_1~Q_5 문항의 경우 평균 약 1.3%이고 Q_6~Q_9 문항은 평균 약 4.1%이다.

표 4 자동 채점 수행 결과

문항	개수				
	전체	모범답안 완전일치	고빈도답안 완전일치	개념 기반	미판단 답안
Q_1	3013	2806	109	25	73
Q_2	3018	2603	296	66	53
Q_3	3013	2611	318	33	51
Q_4	3023	2249	722	21	31
Q_5	3035	1774	1229	23	9
Q_6	3013	67	2619	199	128
Q_7	3012	2004	858	30	120
Q_8	3010	807	1982	91	130
Q_9	3010	816	1945	130	119

5. 결론 및 향후과제

본 논문은 자동 채점 시스템의 원활한 동작을 위해서 학생 답안을 분석 및 정보를 가공하고 채점자가 작성한 부가 정보를 이용하여 문항 채점 정보 파일인 정답 템플릿을 생성하는 방법을 소개하였다. 또한 소개한 방법으로 프로그램을 작성하여 9개 문항에 적용하는 실험을 진행하였다. 생성된 정답 템플릿을 이용하여 채점한 결과 1어절~1문장으로 구성된 학생 답안의 약 96.5%를 채점하였다.

하지만 여전히 미판단 답안이 존재하고 이로 인해 채점 작업을 한 번 이상 중복 수행해야 하는 문제가 있다. 향후 정답 템플릿 생성 과정을 보완하여 한 번의 채점 사이클로 모든 학생 답안을 처리할 수 방법에 대한 연구를 진행할 예정이다. 현재 시행중인 다양한 시험에 자동 채점 시스템을 적용하여 보다 효율적인 채점 작업이 가능할 것으로 기대한다.

참고문헌

[1] 성태제, 양길석, 강태훈, 정은영, “학업성취도평가 서답형 문항 컴퓨터 채점화 방안 탐색,” 한국교육과정평가원 연구보고서 RRE 2010-1, 2010.

[2] 박일남, 노은희, 심재호, 김명화, 강승식, “한국어 서답형 자동채점을 위한 정답 템플릿 기술 방법,” 제 24회 한글 및 한국어 정보처리 학술대회, pp.138-141, 2012.

[3] 박은아, 이문복, “컴퓨터기반 문제해결 능력 평가 시범 적용과 향후 과제,” 한국교육과정평가원 연구보고서 ORM 2012-92, 2012.

[4] 강승식, 한국어 서답형 문항(단어-구 수준) 자동채점 프로그램 개발, 한국교육과정평가원 연구보고서, 2012.

[5] 강승식, 한국어 형태소 분석과 정보 검색, 홍릉과학출판사, 2002.

[6] 노은희, 심재호, 김명화, 김재훈, "대규모 평가를 위한 한국어 서답형 자동채점 프로그램 개발의 전망,” 한국교육과정평가원 연구자료 ORM 2012-92, 2012.

[7] S. Dikli, "An overview of automated scoring of essays," The journal of technology, learning, and assessment, pp.1-35, 2006.

상품평 분석을 통한 상품 평가 요약 시스템

김제상O,정군영, 권인호, 이현아
금오공과대학교, 컴퓨터소프트웨어공학과
oiu124@naver.com, gunyoung20@naver.com, kih6412@naver.com, halee@kumoh.ac.kr

Product Review Summarization through Review Sentence Analysis

Je-Sang KimO, Gun-young jung, In-ho Gwan, Hyun-Ah Lee
Kumoh National Institute of Technology, Dept. of Computer Software Engineering

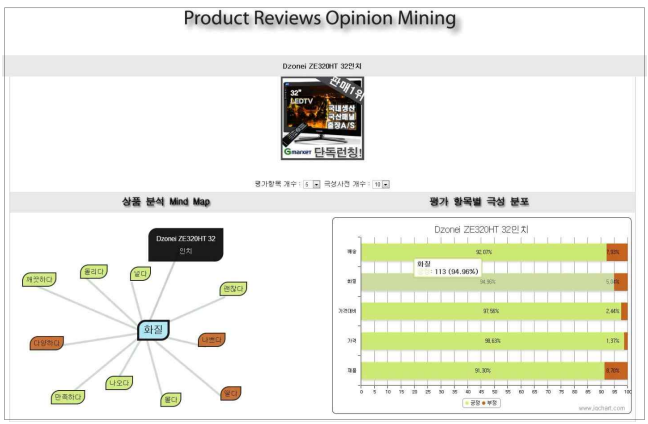
요 약

다수의 상품평 요약은 인터넷 쇼핑몰 고객에게 편의를 제공할 수 있다. 본 논문에서는 상품평 요약 시스템의 성능 향상을 위한 방안을 제안한다. 시스템은 크게 상품평의 평가 항목 추출과 극성 사전 생성, 극성 판별 단계로 구성된다. 평가 항목 추출에서는 외부 연관도의 영향력을 줄이고, 극성 사전 생성에서는 단어 거리 평균을 적용한다. 제안한 방식을 사용하였을 때 평가 항목에 대한 문장의 극성 판별 시 90.8%의 정확율을 보였다.

주제어: 평가 항목, 상품평 요약, 극성 판별, 감성 분류

1. 서론

본 논문에서는 상품평을 자동으로 요약하기 위한 시스템을 제안한다. 본 시스템에서는 평가 항목, 즉 '색상', '크기' 등의 상품의 다양한 특성별로 상품을 긍정과 부정 중 어느 극성(polarity)인지 판별하여 사용자가 원하는 정보를 알아보기 쉽게 제시한다. 본 시스템은 상품평을 수집하고 정형화한 뒤, 상품평 내 어휘 정보와 웹 검색을 활용하여 평가 항목을 추출하고, 언어 처리기법을 이용하여 평가 항목별 평가 어휘와 평가 어휘의 극성을 자동으로 추출하여, 평가 항목별 극성을 [그림 1]에서와 같이 그래프 형태로 요약하여 출력한다. 각 단계는 상품평과 웹 검색 결과를 활용하여 모두 자동으로 처리된다.



[그림 1] 상품평 요약 시스템
(평가 항목별 평가어휘 출력 / 평가 항목별 극성 분포)

2. 관련 연구

상품평의 의견 추출에서는 세부적인 상품 특징(예를 들어, 스커트의 경우 '사이즈', 디지털 카메라의 경우 '액정')에 대한 기존 구매자의 평가가 구매 예정자의 구매 결정에 중요한 판단 기준이 될 수 있어 특징 기반 요약이 필요하다. 국내에서 상품 특징 단어를 추출하는 방법은 k-Structure[1]를 이용한 방법과 PMI-IR 방식[2]을 이용한 방법 등이 있다. [3]과 [4]에서는 평가 항목을 추출하기 위해서 특징 추출에 사용되는 통계적 기법으로 PMI를 이용하며, [5]에서는 ME-NAS모델을 사용하여 상품평의 속성과 감정 데이터를 추출한다. 이러한 연구들은 평가 항목에 따라 달라지는 평가 어휘의 극성을 처리 하기 위한 연구들으로, 특징 기반 요약 방법을 사용하고 있다.

[6]는 도메인별로 평점이 포함된 상품평을 수집하여 상품평의 평점 데이터를 활용하여 극성 사전을 생성하고, [7]은 평가 항목에 의존적이지 않은 소량의 시드(seed) 극성 사전을 이용하여 자동 확장 방식으로 극성 사전을 생성한다. [5]는 어절 유니그램 모델을 제안하여 극성 사전을 생성한다. 도메인별 평점을 활용할 경우 평점 데이터의 불분명성이 문제가 되고, 어절 유니그램 및 바이그램 이상의 구를 포함한 확률 토픽 모델을 활용한 [5]는 amazone.com의 상품평을 활용한 실험 모델이다.

본 연구는 [3]과 [7]의 방법에 기반하여, 개선된 극성 사전의 자동 구축과 평가 요약을 목표로 한다.

3. 상품 평가 요약 시스템

상품 평가 요약 시스템은 크게 상품평 수집과 정형화, 상품 평가 항목 추출, 극성 사전 생성, 극성 판별의 네 부분으로 구성된다. 대상 상품평은 웹에서 수집되며, [7]의 구어체 보정과 정형화를 수행한다. 다음에서는 본

논문에서 제안하는 평가 항목 추출과 극성 사전 생성 방식을 설명한다. 최종적인 극성 판별은 [7]와 동일한 방식을 사용한다.

3.1 상품 평가 항목 추출

[3]의 경우, 상품 평가에 의미없는 단어의 영향력을 줄이기 위해 웹문서에서의 공기 빈도를 외부 연관도로 사용한다. 이에 대한 결과 분석에서 외부 연관도로 인하여 정확하지 않은 단어가 평가 항목으로 추출되는 경우가 발견되었다. 본 연구에서는 외부 연관도의 순위별 편차를 줄여 평가 항목 추출의 정확도를 향상시키고자 한다. 수식 (1)에서는 [3]의 PMI-RTF의 외부 연관도에 log를 취하여 외부 연관도 값의 편차를 감소시킨다.

$$PMI-RTF(t_i c_{name}) = \frac{f_{review \in c}(t_i)}{MAX(f_{review \in c})} \times \frac{\log(f_{web}(c_{name}, t_i))}{\log(f_{web}(t_i))} \quad (1)$$

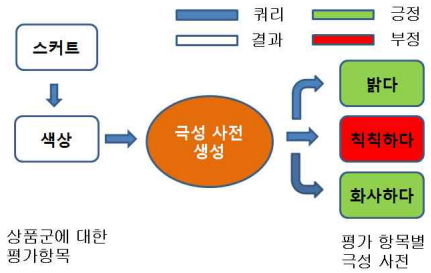
3.2 극성 사전 생성

극성 사전의 생성을 위해서 [7]을 참조하여 수동으로 극성을 분류한 소량의 시드(seed) 집합에 기반한 극성 사전 자동 확장을 수행한다. [7]에서는 긍정 어휘와 부정 어휘에 대해 극성 어휘와의 거리 합의 역수를 사용하여 극성 점수를 계산한다. 이 경우 극성 어휘가 등장할 때마다 $Score(e,p+)$와 $Score(e,p-)$가 점차 감소하여 0에 가까워지는 결과를 얻게 된다. 이로 인하여 사용빈도가 높은 단어가 확장 사전에서 제외되어, 충분한 크기의 확장 사전을 얻기 어렵다. 본 연구에서는 사전 확장의 효율성을 높이기 위해 평가 항목 e와의 거리 값의 평균 $\overline{dist(e,p+)}$와 $\overline{dist(e,p-)}$을 사용한 수식 (2)와 (3)을 제안한다. 수식 (2)와 (3)의 결과 중 임계치(threshold)가 0.2 이상인 결과 어휘에 한해서 긍정 어휘 또는 부정 어휘로 판단한다.

$$Score(p,+) = \frac{f(p+)}{f(p+)+f(p-)} \times \frac{1}{\overline{dist(e,p+)}} \quad (2)$$

$$Score(p,-) = \frac{f(p-)}{f(p+)+f(p-)} \times \frac{1}{\overline{dist(e,p-)}} \quad (3)$$

생성된 극성 사전은 [그림 2]과 같이 평가 항목에 대한 극성을 가진 평가 어휘들의 쌍으로 생성된다. 예를 들어, “색상” 항목에 대해서 “밝다”, “화사하다”는 긍정 어휘, “칙칙하다”는 부정 어휘로 분류된다.



[그림 2] 극성 사전 생성의 입출력

4. 실험 및 평가

시스템 구축에는 형태소 분석기[8], 네이버 웹 문서 검색 Open API를 활용했다. 실험을 위한 데이터는 온라인 쇼핑몰[9]의 데이터를 수집하였으며, 실험 데이터에 대한 상세한 정보는 [표 1]와 같다.

[표 1] 실험 데이터 상세 정보

카테고리 명	상품평 (개)	문장 수 (문장)
전자사전	5,364	13,631
스커트	3,681	6,849
LED TV	3,258	6,416
여성 가방	3,391	7,652
여성 티셔츠	1,751	3,571
자전거	3,148	7,334
Total	20,593	45,453

[표 2]는 평가 항목 추출을 위한 기존의 PMI-RTF[3]와 본 논문의 결과를 비교한다. 결과에서 본 논문의 개선된 PMI-RTF이 높은 정확률을 보이는 동시에, 정답 평가 항목들이 상위 등수에 더 밀집해 있는 것을 확인할 수 있다.

[표 2] ‘전자사전에 대한 평가 항목 상위 20위’

	개선한 PMI-RTF	기존 PMI-RTF	외부 연관도	내부 연관도
1	기능	기능	mp3	배송
2	배송	사전	추천	기능
3	사전	가격	전자	제품
4	가격	mp3	기능	디자인
5	제품	동영상	사전	사전
6	디자인	샤프	딕플	가격
7	생각	추천	배터리	생각
8	상품	전자	동영상	마음
9	사은품	제품	샤프	물건
10	동영상	배터리	아이리버	상품
11	아이	영어	가격	사은품
12	샤프	사용	영어	동영상
13	사용	아이리버	라디오	선물
14	마음	딕플	구매	아이
15	발음	아이	메모리	만족
16	중국어	누리안	사용	발음
17	영어	구매	누리안	중국어
18	선물	하나	하나	사용
19	누리안	메모리	터치	샤프
20	하나	상품	아이	성능

평가 항목 추출의 정확도를 평가하기 위하여, 스커트 상품에 대한 상품평에서 수동으로 평가 항목을 추출하여 정답 평가 항목을 구축하였다. 평가 항목 자동 추출 상위 20위에 대한 평가에서 [3]은 평균 28.7%의 재현율을 보였으나, 본 논문의 방식은 36.7%의 재현율을 보여, 외부 연관도의 순위별 외부 연관도 값의 편차를 줄인 것이 효과적임을 알 수 있다.

개선된 방법으로 추출된 평가 항목을 이용한 극성 판별 결과는 [표 3]와 같다. 평가에서는 상품평 500문장에 대하여 평가 항목별로 극성 정확도를 수동으로 평가하였다. 극성 판별의 정확도는 약 89.9%로, [7]의 결과인 81.8%의 정확도와 비교하여 봤을 때 7.1% 정도 높은 수

치를 확인 할 수 있다.

[표 3] 극성 판별 시스템 평가

상품	평가항목	정확률
위드이폰 (스커트)	배송	93.33
	재질	75.00
	사이즈	82.14
TIVA (LED TV)	배송	81.81
	화질	93.44
	가격	95.74
누리안 (전자사전)	기능	92.68
	가격	96.87
	디자인	98.14
평 균		89.90

5. 결론

본 논문에서는 특징 기반 요약에 특화된 단일 문서인 상품평에 대하여 분석하는 방법을 제안하였다. 제안한 특징 기반 요약 방법을 온라인 쇼핑몰의 상품평 뿐만 아니라 다양한 장르의 단일 문서의 댓글에 적용시킨다면 효과적인 데이터 마이닝 기법으로 활용될 수 있을 것으로 기대된다. 향후 연구로는 비정형 데이터들의 비문법적 표현의 추가적인 보정과 분석된 데이터에 대한 자동 학습이 필요하다.

참고문헌

[1] "k-Structure를 이용한 한국어 상품평 단어 자동 추출 방법",강한훈, 유성준, 한동일,정보과학회논문지: 소프트웨어 및 응용 제 37권 제 6호(2010.6)

[2] “PMI-IR을 이용한 국내 소셜커머스 상품 평가”, 임지연,김이준, 성균관대학교,2011 한국컴퓨터종합학술대회 논문집 Vol.38, No.1(C)

[3] “효율적인 상품평 분석을 위한 어휘 통계 정보 기반 평가 항목 추출 시스템”, 이우철, 이현아, 이공주, 정보처리학회논문지 16-B권 6호 , 2009.12, 497~502p

[4] “상품리뷰요약을 위한 대체어 자동추출”, 안미희, 백종범, 이수원, 한국컴퓨터종합학술대회 논문집 Vol.39, No.1(B). 2012

[5] “상품평에서 속성과 감정을 나타내는 어절 n-gram 추출을 위한 MaxEnt 확률 토픽 모델”, 이영록, 박성배, 이상조, 2012년 가을 학술발표논문집 Vol.39, No.2(B)

[6] “상품평 극성 분류를 위한 특징별 서술어 긍정/부정 사전 자동 구축”, 송종석, 이수원, 정보과학회논문지 소프트웨어 및 응용 제 38권 제 3호(2011.3)

[7] “상품평의 언어적 분석을 통한 상품 평가 요약 시스템”, 이우철, 이현아, 이공주, 정보처리학회논문지 17-B권 1호, 2010.2, 93~98p

[8] 한나눔 한국어 형태소 분석기, JHanNanum-0.8.4-kover, http://kldp.net/projects/hannanum/, 2010/7/31

[9] 비교 쇼핑몰 Basket - http://www.basket.co.kr/

LDA를 이용한 트윗 유저의 연령대, 성별, 지역 분석

이호경○, 천주룡, 송남훈, 고영중
동아대학교
hogay88@gmail.com, balendia@gmail.com, nh.song.89@gmail.com, youngjoong.ko@gmail.com

Analyzing ages, gender, location on Twitter using LDA

Ho-Kyung Lee○, Ju-Ryong Chun, Nam-Hoon Song, Youngjoong Ko
Donga University, Computer Engineering

요 약

요즘 많은 사람들은 트위터를 통해 짧은 문장의 트윗을 작성하여 자신의 의견이나 생각을 표현한다. 사람들이 작성한 트윗은 사용자의 연령, 성별, 지역에 따라 다른 특성이 담겨있다. 이러한 정보를 이용하여, 기업에서는 연령대, 성별, 지역에 따라 각기 다른 마케팅 전략을 세울 수 있을 것이다. 본 논문에서는 트위터 사용자들의 트윗을 분석하여 연령대, 성별, 지역을 예측하려 한다. 네이버 오픈사전의 자질, 한국전자통신연구원(ETRI)의 개체명 사전을 이용한 자질 및 한국어 형태소 분석, 음절 단위의 bigram을 클래스별 의미 있는 자질로 선택하고 LDA를 이용하여 예측된 확률분포를 활용하여 분류한 결과, 연령 72%, 성별 75%, 지역 43%의 납득할만한 예측 정확도 결과를 얻게 되었다.

주제어: Twitter, LDA(Latent Dirichlet Allocation), 연령대, 성별, 지역

1. 서론

최근 들어, 소셜 네트워크 서비스(Social Network Service)가 확산 되면서 사람들이 자신의 의견, 생각, 개인적인 경험을 공유하고 표현할 수 있게 되었다. 트위터(twitter)는 블로그의 인터페이스와 미니홈페이지 기능, 메신저의 기능을 한데 모아놓은 소셜 네트워크 서비스이다. 트위터는 하나의 트윗(tweet)을 140자 이내로 제한하고 있으며, 그 트윗은 사용자의 의견이나 생각을 포함하고 있다. 트위터 뿐만 아니라, 전 세계적 SNS에 대한 관심 및 스마트폰 보급률의 증가에 따라 SNS를 사용하는 사용자들의 수가 증가하고 있으며 또한 트위터와 관련된 많은 연구가 수행되고 있다.

그리고 이러한 트위터 분석 연구는 트위터 사용자의 성별, 연령대, 거주지역 그리고 기업에서는 관심있는 보험상품을 조회하는 패턴을 파악하여 고객의 요구사항을 분석하고 신규 보험 상품 개발 등에 이용되는 사례에 적용 되고 있다.

사람들은 개인마다 자신만의 특성이 드러나는 글쓰기 방식을 가지고 있다. 이러한 개개인의 글쓰기 방식 특성을 이용하여 연령대를 기준으로 집단의 문체를 분석, 판별하고자 하는 연구들이 많이 이루어지고 있다. 이들 연구는 같은 연령대의 사람들은 비슷한 시기에 동일한 공감되는 일들을 겪으면서 다른 연령대와 구별되는 성향이 글속에 잘 나타나게 된다는 점에서 착안한 것이다. 연령대를 분류하는데 좋은 자질로는 이모티콘 자질이 있다. 10/20대의 경우 30/40대에 비해 이모티콘의 사용 빈도가 높다. 성별을 분류하는데 에는 호칭에 관한 단어들이 좋은 자질로 이용된다. 그리고 지역에 따른 좋은 자질로는 각종 지역 명칭이나 사투리 사전을 이용 할 수 있다.

본 논문에서는 네이버의 오픈 사전, ETRI 개체명 사전과 같은 자원을 활용하여 연령대, 성별, 지역을 분류하는데 자질들을 선정하여 사용하였으며 트윗 길이 정보[2], 자음 및 모음으로 구성된 한글 이모티콘 개수 정보[3] 또한 자질로 이용하였다. 연령대, 성별, 지역별로 구성한 자질들을 LDA를 활용하여 분포를 계산하고, 계산된 분포를 기반으로 트위터 사용자의 연령대, 성별, 지역을 분류하는 방법을 제안한다.

본 논문의 구성은 다음과 같다. 2장에서는 제안하는 방법과 연관된 관련 연구에 대해 기술하고 3장에서는 제안하는 기법의 전체적인 구성과 각 부분에 대해 설명한다. 4장에서는 LDA를 이용한 실험방법 및 결과를 살펴보고 5장에서는 결론 및 향후 연구에 대해 말한다.

2. 관련 연구

트위터에 관한 초기 연구들은 트윗 행태와 이용자에 관한 기초통계 분석 위주의 연구들이었으나, [1]에서는 한국어 트위터에서 연령대에 따라 문체가 어떻게 달라지는가를 비교적 적은 규모의 자질을 통해 분석하고 예측하였다. [1]에 더하여 [2]에서는 트윗의 문체뿐 아니라 내용을 다루고 있는 부분들을 계량화 할 수 있게 자주

이 논문은 2013년 정부(교육과학기술부)의 재원으로 한국연구재단의 지원을 받아 수행된 기초연구사업임(No. NRF-2013R1A1A2009937)

1) 각 사용자의 한 트윗당 길이 정보

2) 한글 모음 및 자음으로 구성된 이모티콘 개수
예) ㅎㅎㅎㅎ, ㅋㅋㅋㅋ

쓰이는 n-gram 방식과 같은 자질들을 추가로 구성하여 연령대 및 성별 예측을 하였다.

다른 트윗 분석 연구로는 다양한 시간 스케일에서 주제를 추출할 수 있는 NMF(Non-negative Matrix Factorization) 클러스터링 기법을 적용하여 트위터의 트랜드를 [3]에서 분석하였고, [4]에서는 n-gram을 이용한 자질 추출 기법과 슬라이딩 윈도우(sliding window)를 이용하여 자질을 추출하여 트윗을 분석하였다. [5]에서는 100여개에서부터 많게는 5,700가지의 자질들의 분포를 토대로 연령대를 예측하는 방법을 제안하였다.

마지막으로 [6]에서는 LDA를 기반으로 트위터 데이터를 분석하여 토픽의 변화 시점 및 패턴을 파악하는 연구를 진행하였다.

3. 제안 방법

본 논문에서는 한국어 형태소 분석 결과[가]와 음절 단위 bigram[나]을 기본 자질로 사용하고, 여기에 연령대, 성별, 지역을 분류하는데 좋은 정보를 주는 자질들을 추가로 사용한다. 한국어 형태소 분석 정보는 명사 정보만을 이용한다. 연령대, 성별, 지역을 분류하기 위해 여러 자질들을 기반으로 LDA를 이용하여 분포를 추정한다. 추정된 분포를 활용하여 사용자가 작성한 여러 트윗의 연령대, 성별, 지역을 분류함으로써 효과적으로 사용자의 연령대, 성별, 지역을 분류하는 방법을 제안한다.

3.1 데이터 구축

트위터 사용자들의 개인 프로필 및 국내 트위터 관련 사이트를 이용하여 트위터 사용자의 연령대, 성별, 거주지역에 따른 트윗의 가장 최근의 데이터만을 수집한다. 성별은 남, 여로 분류, 연령대는 10대~20대, 30대~40대로 분류, 그리고 거주지역은 수도권, 충청도권, 전라도권, 경상도권으로 분류하여 모든 트위터 사용자를 균등하게 수집한다. 트윗 하나당 문서 하나로 구성하여 외국어로 된 트윗, 반복된 트윗, 광고성 트윗을 제거하는 정제과정을 거친다.

3.2 자질 선택 및 추출

표 1. 자질 분류

구분		예
연령대 분류 자질 [다]	**[다-1] 채팅어, 유행어, 신조어**	쩐다, 헐랭, 멘붕, 근자감, 귀요미, 본좌, 득템 ...
연령대 분류 자질	**[다-2] 정치, 경제, IT, 연예 뉴스**	4대강 국회 부동산 금리 보조금 KT 오자룡 ...
[다]		
연령대 분류 자질 [다]	**[다-3] 트윗 길이 정보**	밥 먹자!/S0, 태안 근흥초등학교에서 ...(중략)... 맛보았습니다. 새삼 SNS의 위력을 절감했습니다./S1
	[다-4] 자음 및 모음 한글 이모티콘 개수 정보	ㅎㅎㅎㅎ ㅋㅋㅋㅋㅇㅋㅇㅋ/E0
성별 분류 자질 [라]	**[라-1] 남녀 호칭, 특수기호**	언니, 오빠, 누나 형, 누님, 형님 ... ※, ★, ☆, ♥, ♡, ♠, ♤, ▲, △, ◈ ...
지역별 분류 자질 [마]	**[마-1] 지역명, 관광지명 및 사투리**	수원, 인천, 대구, 광주, 청주, 창원, 청담동, 이태원, 종로, 해운대, 사하구, 월미도, 북한산, 지리산 ... 가구웁는, 개갈, 산찬하다, 부앙부앙하다, 천지빼가리, 포분들리다 ...

*** S0 : 트윗의 길이 5자 이하**
*** S1 : 트윗의 길이 81자 이상**
*** E0 : 자음 및 모음 한글 이모티콘 개수 10개 초과**

연령대 자질으로는 10대~20대와 30대~40대를 구분할 수 있는 자질로 네이버 오픈 사전을 이용하여 [다-1]에 수록된 단어들 중 추천수 10이상만 추출, 또한 네이버의 기사 중 정치, 경제, IT, 연예 뉴스[다-2]들의 주제문의 명사만을 추출하여 총 1354개의 단어로 이루어진 연령대 자질 사전을 구축하였다. 10대~20대와 30~40대를 구분할 수 있는 또 다른 자질로 연령대마다 다를 것이라고 판단되는 [다-3]를 자질로 이용하였다. 또한 'ㅋ' 나 'ㅎ' 또는 'ㅠㅠ'와 같은 [다-4]를 자질로 이용하였다.

[다-3]는 트윗의 길이를 5자 이하 'S0'태그, 81자 이상은 'S1'태그를 부착하여 사용하였으며 [다-4]는 트윗의 자음 및 모음 한글 이모티콘 개수가 10개 초과하는 트윗에 'E0'태그를 부착하여 사용하였다.

성별 자질 선택으로는 남, 여를 구분할 수 있다고 판단되는 [라-1]를 자질로 하여 총 82개의 성별 자질 사전을 구축하였다.

지역 자질 선택으로는 한국전자통신연구원(ETRI)의 개체명 사전에서 각 지역을 구분할 수 있다고 판단되는 수도권, 충청도, 전라도, 경상도의 지역명 및 관광지명을 추출하고 또한 네이버 오픈사전의 사투리 사전에서 충청

도, 전라도, 경상도의 사투리를 추천수 10이상만 추출하여 총 8255개의 [마-1]을 구축하였다.

3.3 LDA의 적용 및 계산

사용자의 각 트윗에서 제안한 자질들을 추출하고 LDA를 적용하여 분포를 계산하였다. LDA에서 연령대, 성별은 토픽을 2개로, 지역은 토픽을 4개로 설정하였다. 아래는 연령대별 각 주제에서 가중치 기준으로 상위 N 개의 단어의 목록의 예이다.

표 2. 연령대별 자질 사전 토픽 결정 예

토픽1	토픽2
학교	기업
출첵	퇴근
멘붕	부동산
헐	회사
아이돌	맛집
문상	회장
...	...

본 논문에서는 LDA를 이용하여 계산된 토픽-단어 분포를 기반으로 트윗의 토픽을 결정한다. 예를 들어, 연령대별 토픽 분포에서 토픽1에 10대~20대의 자질이, 토픽2에 30~40대의 자질이 각각 많이 등장하였다고 판단되면 토픽1은 10대~20대, 토픽2는 30~40대라고 직접 결정한다.

표 3. 문서별 가중치에 따른 토픽 예측 예

	문서	토픽1	토픽2
사용자	문서1	**3.244**	2.756
	문서2	0.352	**1.648**
	문서3	**21.528**	6.472
	문서4	**5.616**	2.384
	문서5	**5.000**	0.000
	대소 비교	4	1

사용자의 토픽은 LDA를 적용하여 나온 문서당 토픽 분포 가중치의 결과를 이용하여 예측한다. 한 트위터 사용자의 토픽을 예측하는 방법은 다음과 같다.

1. 한 트위터 사용자당 트윗 문서에서 토픽들의 가중치를 대소 비교한다.

2. 개수가 많이 나온 토픽을 선택하여 그 트위터 사용자를 예측한다.

예를 들어 한 사용자가 n개의 트윗 문서 $\{d_1, d_2, \cdots, d_n\}$를 가질 때, j번째 토픽 t_j 에 대한 문서 d_i의 스코어를 $Score_i(t_j)$라고 하면, $Vote_i(t_j)$는 수식 (1)과 같이 정의한다.

$$Vote_i(t_j) = \begin{cases} 1 & \text{if } j = (\text{argmax}_x Score_i(t_x)) \\ 0 & else \end{cases} \quad (1)$$

그리고 그 사용자는 최종적으로 수식 (2)에서 산출된 토픽 t_y 에 매핑시킨다

$$\text{argmax}_y \sum_{i=1}^{n} Vote_i(t_y) \quad (2)$$

즉, 앞에서 클래스 별로 결정한 토픽과 한 트위터 사용자의 트윗 문서에서 예측한 토픽을 비교하여 정답을 판단한다.

4. 실험 및 결과

각 클래스 별로 형태소 분석기반 자질, bigram 자질을 baseline로 설정하고 본 논문에서 제안한 연령대, 성별, 지역별로 분류된 자질 정보 및 사전을 조합하여 트위터 사용자를 예측하였다.

예측 정확도의 계산 방법은 기본적으로 f1-measure을 사용하였고 각 클래스 별 micro-평균으로 계산하였다.

4.1 데이터 집합

말뭉치는 최근 트위터를 활발히 이용하는 100명의 트윗 사용자당 190~200개씩, 총 19567개의 트윗으로 구성하였다. 그리고 개인 프로필 및 국내 트위터 관련 사이트에 기록된 것을 기준으로 10대, 20대, 30대, 40대별로 25명씩, 남자 51명, 여자 49명, 지역을 수도권 30명, 충청도권 21명, 전라도권 22명, 경상도권 27명으로 각 클래스 별로 균등하게 구성하여 트윗을 수집하였다.

4.2 실험 결과

실험은 한 트위터 사용자를 예측하는 결과이다.

표 4. 실험 방법에 따른 연령대 예측 결과

방법	10대/20대	30대/40대	micro
[가]	0.429	0.586	0.520
[나]	0.505	0.515	0.510
[다]	0.632	0.512	0.580
[다] + [다-3]	0.568	0.661	0.620
[다] + [다-3] + [다-4]	0.750	0.625	0.700
[가] + [다] + [다-3] + [다-4]	**0.763**	**0.659**	**0.720**
[나] + [다] + [다-3] + [다-4]	0.775	0.592	0.710

[가] : 형태소 분석 명사 자질 [baseline1]

[나] : bigram 자질 [baseline2]
[다] : 연령대 자질 사전
[다-3] : 트윗 길이 정보
[다-4] : 한글 이모티콘 개수 정보

연령대 예측 결과는 형태소 분석 명사 자질과 연령대 자질 사전, 트윗 길이 정보, 한글 이모티콘 개수 정보를 모두 조합한 방법이 가장 높은 72%의 정확도로 예측할 수 있었다.

표 5. 실험 방법에 따른 성별 예측 결과

방법	남	여	micro
[가]	0.598	0.434	0.530
[나]	0.608	0.592	0.600
[라]	**0.771**	**0.725**	**0.750**
[가] + [라]	0.452	0.603	0.540
[나] + [라]	0.432	0.667	0.580

[가] : 형태소 분석 명사 자질 [baseline1]
[나] : bigram 자질 [baseline2]
[라] : 성별 자질 사전

표 6. 실험 방법에 따른 지역별 예측 결과

방법	수도권	충청도	전라도	경상도	micro
[가]	0.286	0.244	0.333	0.222	0.270
[나]	0.320	0.324	0.340	0.333	0.330
[마]	**0.233**	**0.421**	**0.379**	**0.633**	**0.432**
[가] + [마]	0.413	0.308	0.235	0.171	0.320
[나] + [마]	0.381	0.316	0.233	0.171	0.300

[가] : 형태소 분석 명사 자질 [baseline1]
[나] : bigram 자질 [baseline2]
[마] : 지역 자질 사전

그에 반해 성별 예측 결과와 지역별 예측 결과는 형태소 분석 명사 자질이나 bigram자질과 함께 적용한 자질 사전을 모두 조합한 방법의 예측 정확도가 생각보다 높지 않았다. 성별이나 지역별 예측에서는 필요하다고 판단되는 자질이 한정적이기 때문에 많은 자질이 포함된 형태소 분석 명사 자질 및 bigram 자질은 역효과가 나지 않았나 판단된다.

따라서 성별 예측 결과에서는 성별 자질 사전만을 사용한 방법이 가장 높은 75%의 정확도를 보여주었고 지역별 예측 결과에서는 지역별 자질 사전만을 사용한 방법이 약 43%의 정확도로 가장 높게 예측되었다.

baseline로 적용했던 형태소 분석 명사 자질과 bigram 자질은 토픽을 2개로 나눈 연령대, 성별 예측 결과는 50%가 조금 넘는 일반적인 정확도를 보여주며 마찬가지로 토픽 4개로 나눈 지역별 예측 결과는 25%가 조금 넘는 정확도를 보여주었다. 본 논문에서 제안한 각 클래스별 자질 정보와 사전을 이용하였을 때 확실히 예측 정확도가 높아지는 것을 볼 수 있었다.

5. 결론

본 논문에서는 각 클래스 별 특성에 맞는 자질 사전을 구축하여 LDA를 이용, 트위터 사용자를 예측하였다. [2]에서 제시한 자질 구축 방법 및 SVM을 이용한 예측과 대응하여, 학습을 하지 않고도 각 클래스별로 판단되는 자질을 추출하여 LDA를 이용함으로써 납득할 만한 정확도를 예측하였기 때문에 의미 있는 연구가 되었다고 생각한다. 그리고 향후 연구로는 트위터의 리트윗(Retweet)[1], 리플라이(Reply)[2]기능을 이용하여 사용자간의 관계를 이용하여 정확도를 높이고 나아가 사용자의 직업, 관심사 등의 토픽을 늘려 예측 할 수 있는 시스템을 구축하는 연구를 진행 할 예정이다. 이러한 연구들이 활발히 진행된다면 마케팅 분야에 적절히 활용 될 수 있을 것이라 생각된다.

참고문헌

[1] 김상채, 박종철, “한국어 트윗의 문체 기반 자질 분석을 통한 연령대 예측” , HCI 2012 학술대회
[2] 김상채, 박종철, “문체 분석을 활용한 한국어 트위터 사용자의 연령대 및 성별 예측”, 2012 한국컴퓨터종합학술대회 논문집 Vol.39, No.1(B)
[3] 하용호, 임성원, 김용혁, “내용기반 트윗 클러스터링을 통한 트랜드 분석”, 2012년 가을 학술발표논문집 Vol.30, No.2(B)
[4] 홍초희, 김학수, “트윗 분류를 위한 효과적인 자질 추출”, 2011 한국컴퓨터종합학술대회 논문집 Vol.38, No.1(A)
[5] J. D. Burger, J. Henderson, G. Kim and G. Zarrella, “Discriminating Gender on Twitter”, Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing, pp. 1301-1309, 2011.
[6] 전설아, 허고은, 정유경, 송 민, “트위터 데이터를 이용한 네트워크 기반 토픽 변화 추적 연구”,정보관리학회지 30(1), 285-302. 2013.

1) 리트윗(Retweet)이란 자신의 스트림에 나타난 다른 사용자의 트윗을 자신의 팔로워들에게 전파하는 기능

2) 리플라이(Reply)란 자신의 스트림에 나타난 다른 사용자의 트윗에 대한 사용자의 답변을 작성한 트윗을 사용자에게 보내는 기능

문장 길이 축소를 이용한 구 번역 테이블에서의 병렬어휘 추출 성능 향상

정선이, 이공주
충남대학교 정보통신공학과
syjeong@cnu.ac.kr, kjoolee@cnu.ac.kr

Performance Improvement of Extracting Bilingual Term from Phrase Table using Sentence Length Reduction

Seon-Yi Jeong, Kong-Joo Lee
Dept. of Information and Communication Engineering, Chung-Nam University

요 약

본 연구는 대량의 특정 도메인 한영 병렬 말뭉치에서 통계 기반 기계 번역 시스템을 이용하여 병렬어휘를 효과적으로 추출해 낼 수 있는 방법에 관한 것이다. 통계 번역 시스템에서 어족이 다른 한국어와 영어간의 문장은 길이 및 어순의 차이로 인해 용어 번역 시 구절 번역 정확도가 떨어지는 문제점이 발생할 수 있다. 또한 문장 길이가 길어짐에 따라 이러한 문제는 더욱 커질 수 있다. 본 연구는 이러한 조건에서 문장의 길이가 축소된 코퍼스를 통해 한정된 코퍼스 자원 내 구 번역 테이블의 병렬어휘 추출 성능이 향상될 수 있도록 하였다.

주제어: 기계 번역, 구 번역 테이블, 용어 추출

1. 서 론

최근 통계기반의 기계 번역에 대한 연구가 활발히 진행되고 있다. 통계 기반의 기계 번역 프로그램인 MOSES[1]는 두 언어의 병렬 코퍼스를 이용하여 GIZA++[2]로부터 두 언어의 어휘의 번역 정보가 담긴 구 번역 테이블을 생성한다. 구절 테이블은 통계 기반 기계 번역에서 자동 번역을 구축할 시 어휘의 번역 정보로 이용된다. 그러므로 구 번역 테이블의 정확도 및 추출 성능은 통계 기반 기계 번역 시스템의 성능과도 직결된다. 본 논문에서는 통계 기반의 기계번역 프로그램인 MOSES를 이용하여 추출된 구 번역 테이블의 용어 추출 성능을 향상시키기 위한 연구를 시도하였다.

한영 기계번역의 경우 단어 번역(word translation) 과정뿐만 아니라 재배열(reordering)과정에서 긴 문장으로 인한 오류가 발생한다. 통계 정보만을 이용하는 단어 번역 단계의 경우 문장의 길이 및 어순 정보는 구 번역 테이블 내 적합한 번역어의 선택에 영향을 미칠 수 있다. 'SOV' 어순의 영어의 경우 문장의 길이가 주어와 서술어를 제외한 내용어들을 중심으로 길어진다고 할 때 서술어에 해당하는 영어의 동사와 한국어의 동사가 극단적으로 멀어질 수 있는 경우가 발생하게 된다. 이 경우 단어가 아닌 구절의 번역 정확도는 떨어질 수 있다. 이러한 문제점을 보완하기 위해 코퍼스와 동일한 도메인의 단어 및 구절 정보를 이용하여 문장 내 병렬 어휘 정보를 삭제하고 문장 길이를 축소하여 유효 번역 단어 간 거리를 줄여 용어 추출 성능을 높이고자 하였다. 본 연구는 기계 번역에서 사용되는 구절 번역 테이블을 활용한 용어 추출 시 용어 및 대역어 추출 성능을 향상시키기 위한 연구이다. 최종 평가 데이터로 구절 번역 테이블로부터 추출된 용어 셋을 사용함을 밝힌다.

논문 구성은 2 장에서 관련 연구를 설명하고 3장에서 본 논문에서 제안한 문장 길이 축소방법에 관해 설명한다. 4 장에서는 평가를 위한 실험환경 및 평가 환경을 설명하고 평가 셋에 대한 기존 결과와 제안 방법의 평가 결과를 보여준다. 5 장에서는 논문 내용을 결론짓고 향후 연구 방향에 대해 논의한다.

2. 관련 연구

통계 기반 기계 번역의 성능 향상을 위해 긴 문장에 대한 문제 인식을 통해 문장을 줄이거나 분할하려고 시도

표 1 영한 예시 문장

no.	길이	문 장
1	33	Recently, with the rapid spread of high-speed communication network the *population* using online photo print service is *rocking up* photograph taken by digital camera is transmitted via the internet and printed and delivered.
2	24	최근 초고속 통신망이 빠르게 보급됨에 따라 디지털 카메라로 찍은 사진을 인터넷에 전송한 후 사진을 인화하여 배송 받는 온라인 사진 인화 서비스를 이용하는 *인구*가 *급증하고 있다*.

했던 연구는 [3, 4, 5]가 있다. [3]의 연구에서는 통계 기반의 기계 번역 시스템에서 20 어절 이상의 긴 문장들을 보다 정확히 분석하기 위해 복수개의 의미 있는 절로 문장 분할을 시도하였다. 문장 내 분할 지점을 인식하기 위해 SVM을 사용하여 분할 지점의 특성을 학습한 후 분할 지점을 탐색한다. [4]의 연구에서는 입력 문장이 길어지면 통계 기반의 기계 번역 성능이 떨어짐을 지적하고 이를 완화하기 위해 긴 문장을 같은 의미의 짧은 문장들로 분할하여 기계 번역의 성능을 향상 시킬 수 있도록 하였다. 분할 방법으로는 변환 규칙을 학습하는 변환 기반 문장 분할 방법을 사용하였다. [5]에서는 긴 문장이 기계 번역에 미치는 영향에 대한 연구를 수행하였다. [5]의 연구에서는 문장의 길이가 길수록 기계 번역에서 발생하는 변형이 많아짐을 지적하고 문장의 길이가 길수록 번역의 성능이 떨어짐을 보였다. 이와 같이 문장 길이에 대한 문제 인식과 기계 번역 성능간의 연구는 존재하지만 긴 문장이 단어 번역 과정 및 구절 테이블에 미치는 영향에 관한 연구는 이루어지지 않고 있다.

3. 병렬 어휘 정보 삭제에 의한 문장 길이 축소

본 장에서는 구절 번역 테이블의 용어 추출 성능 향상을 위한 코퍼스의 문장 길이 축소 방법 및 효과를 설명한다.

3.1. 특정 도메인

기계 번역에서 이용할 문서 자원에서 특정 도메인 특성을 가지는 병렬 말뭉치의 경우 동일한 도메인 특성의 어휘정보가 말뭉치에 상당량 분포하게 된다. 이때 번역 어휘 쌍이 병렬 문장 쌍에서 동시에 존재하게 된다.

3.2 문장 길이 축소

같은 도메인을 공유하는 병렬 코퍼스의 경우 번역 어휘 쌍이 병렬 문장 쌍에서 병렬적으로 발견된다. 이러한 병렬 어휘 쌍을 이용하여 병렬 문장 쌍의 길이를 축소할 수 있다. 표 1은 병렬 코퍼스의 일부 정렬된 문장쌍이다.

문장 쌍 내부에 단일 어절 및 다중 어절의 어휘가 병렬적으로 존재한다. 이 중 단일 어절을 제외한 다중 어절 어휘의 길이를 1로 대치한다면 문장의 길이 축소 효과를 볼 수 있다. 문장 1의 경우 'high-speed communication network'와 'online photo print service', 'digital camera'가 길이 축소를 위한 병렬 어휘로 선택될 수 있다. 이에 해당하는 문장 2의 번역 어절은 '초고속 통신망', '디지털 카메라', '온라인 사진 인화 서비스'이다. 예시와 같이 문장 1과 문장 2에서 병렬 어휘로 대응된 2어절 이상의 어절을 하나의 어절로 축소시킴으로써 전체적인 문장 길이 축소 효과를 기대할 수 있다.

문장 축소 과정은 다음과 같다.

- 정렬된 병렬 어휘가 존재하는 사전을 준비한다.
- 코퍼스 내 모든 병렬 문장 쌍 내 존재하는 번역 어휘 쌍 을 탐색한다.
- 탐색된 병렬어휘는 다음과 같다.

$$Phs = \{Phs_i : i = 1 \cdots n\}$$

$$Pht = \{Pht_i : i = 1 \cdots n\}$$

- 탐색된 번역 어휘 쌍인 모든 Phs, Pht 에 대해 각 언어의 문장에서 '[MARK]'로 대치한다. (token length = 1)
- 원시 언어 문장 S의 길이가 $Len(s)$, 번역 언어 문장 T의 길이가 $Len(t)$ 이고 병렬 어휘 Phs_i, Pht_i의

각각의 길이가 $Len(Phs_i)$, $Len(Pht_i)$ 일 경우
S의 길이는

$$Len(\bar{s}) = Len(s) - \sum_{i=1}^{n}(Len(Phs_i)-1)$$

T의 길이는

$$Len(\bar{t}) = Len(t) - \sum_{i=1}^{n}(Len(Pht_i)-1)$$

로 축소된다.

문장 길이 축소를 위한 병렬 어휘 쌍은 기존의 대역어 사전을 사용하거나, 동일한 도메인의 MOSES의 초벌결과도 사용이 가능하다.

문장 길이 축소의 목적은 유효 단어 간 상이 어순에 의한 거리 차를 줄여 구절 테이블의 용어 추출 성능을 높이는 것이다. 또한 문장 구조 보존을 위해 병렬 어휘는 문장 내에서 삭제하지 않고 길이 1의 태그를 이용해 대치하였다. 표 1의 예시에서 다중 어절 병렬 어휘를 이용할 경우 문장길이는 각각 28, 20으로 축소된다.

문장 길이 축소에 의한 기대 효과는 다음과 같다. MOSES/GIZA++가 허용하는 다중 어절(phrase)의 최대 길이는 7이다[1]. GIZA++는 문장 내 7의 단어 길이 제한 내에서 두 언어의 구절의 적합한 번역어를 탐색한 뒤 통계정보를 이용하여 테이블을 만든다.

표 1의 문장 2의 '인구가 급증하고 있다'를 구 번역 테이블의 유효 다중 어절이라고 가정할 때 다음과 같은 번역어 후보들을 얻을 수 있다.

a	인구가 급증하고 있다 ||| the|DT **population|NN** using|VBG *online|NN photo|NN print|NN service|NN*
b	인구가 급증하고 있다 |||**population|NN** using|VBG *online|NN photo|NN print|NN service|NN* is|BE
c	인구가 급증하고 있다 ||| using|VBG *online|NN photo|NN print|NN service|NN* is|BE **rocking|VBG**

표 2 길이 축소 이전의 구 번역 테이블 예

e	인구가 급증하고 있다 ||| **the|DT population|NN** using|VBG *[MARK]|dummy* is|BE **rocking|VBG up|RB**
f	[MARK] 이용하는 인구가 급증하고 있다 ||| **the|DT population|NN** using|VBG *[MARK]|dummy* is|BE **rocking|VBG up|RB**

표 3 길이 축소된 문장의 구 번역 테이블 예

표 2의 결과를 보면 구 번역 테이블이 추출하는 엔트리 a, b, c 모두 어순 차와 내용어의 길이가 길어짐에 따라 '인구'의 유효 번역어인 'population'과 '급증하고 있다'의 유효 번역어인 'rocking up'을 동시에 포함하지 않고 있다. 구절 번역 테이블에서 이러한 엔트리들은 용어 추출 과정에서 정확한 번역어를 표시한 엔트리가 아니기 때문에 구 번역 테이블에서 용어를 추출할 때 정확도를 떨어뜨릴 수 있다. 반면 표 3의 e, f는 유효 거리 7이내에서 적합한 번역 어휘들을 모두 포함할 수 있게 된다. 이러한 과정을 통해 문장 길이가 축소된 코퍼스를 사용한 구절 테이블에서 노이즈 감소 효과와 새로운 용어 추출 효과를 얻을 수 있을 것이다.

4. 실 험

본 연구에서는 실험에 사용할 코퍼스로 '전자 뉴스' 영한 병렬 코퍼스를 사용하였다. 이 코퍼스의 특성은 전문적인 기술 용어 및 개체명(Named Entity)을 나타내는 어휘가 상당량 분포한다. 또한 다양한 길이의 문장이 존재한다. 실험에 사용한 코퍼스의 특성을 표 4에 나타내었다.

표 4 실험에 사용한 코퍼스 특성

코퍼스	문장수	평균 길이
한국어 (Baseline)	10000	21.16
영어 (Baseline)	10000	36.16
한국어(SLR)	10000	20.59
영어 (SLR)	10000	34.01

본 연구에서는 가장 코퍼스 특성에 부합되는 병렬어휘 정보를 사용하기 위해 같은 도메인의 다른 코퍼스로부터 얻은 MOSES 구 번역 테이블로부터 추출된 병렬어휘 정보를 사용하였다.

표 5 병렬어휘 사전 특성

엔트리	22040
참조된 병렬 어휘	3951
참조된 병렬 어휘 길이 평균 (영어)	2.32
참조된 병렬 어휘 길이 평균 (한국어)	1.22

표 5의 병렬 어휘 사전으로부터 코퍼스 내 병렬 문장에서 일치하는 병렬 어휘가 나타날 경우 모두 길이 1의 토큰으로 대치하고 길이를 축소할 수 있도록 하였다. 실험 결과 실험 코퍼스에 대해 참조된 병렬 어휘는 3951개로 나타났다. 참조된 병렬 어휘 길이의 평균은 영어와 한국어 각각 2.32와 1.22로 나타나는데 다중 어절의 영어 용어의 대역어가 한국어에서 단일 어절로 번역되거나 한국어 띄어쓰기 문제로 인해 영어에 비해 한국어 문장의 길이 축소 효과가 미비하게 나타났다. 표 6에 참조된 번역 어휘의 일부를 제시하였다.

표 6 참조된 병렬 어휘의 일부

embedded linux	임베디드 리눅스
bio industry	바이오산업
role-playing game	롤플레잉게임
offline company	오프라인 기업
wireless internet user	무선 인터넷 사용자
application software	응용 소프트웨어
location information	위치정보
web agency	웹에이전시
wireless lan market	무선랜 시장
bluetooth module	블루투스 모듈
embedded linux operating system	임베디드 리눅스 운용체계
shooting game	슈팅게임
online content	온라인 콘텐츠

실험 과정을 그림 1에 나타내었다.

실험에 사용한 구 번역 테이블의 추출 도구는 MOSES의 GIZA++를 사용하였다. 영어의 POS정보를 사용하기 위해 MOSES의 factored 모델을 이용하였다. 베이스라인 실험은 영어 쪽에만 POS정보가 부착된 한영 병렬 코퍼스를 GIZA++를 이용하여 구 번역 테이블을 추출하고 명사구 추출 필터를 이용하여 병렬 어휘를 추출한다.

명사구 추출 필터는 품사 정보를 이용하여 명사구만을 용어로 추출할 수 있도록 하였으며 구 번역 테이블의 확률 정보를 이용하여 번역 후보를 필터링 하였다.

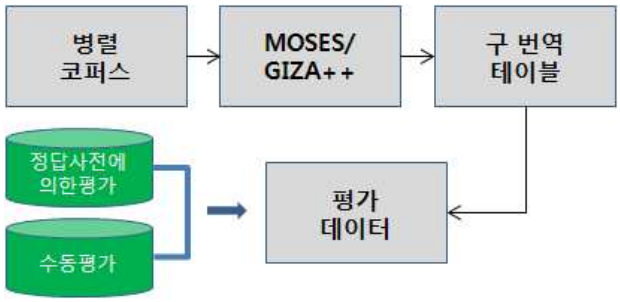


그림 3 데이터 추출 및 평가

제안한 방법의 실험은 기존의 코퍼스를 문장 길이 축소 방법을 이용하여 10000 문장쌍의 길이 축소가 적용된 새로운 병렬 코퍼스를 얻는다. 이 병렬 코퍼스 또한 영어 쪽에만 POS정보가 부착되어 있다. 베이스라인의 POS정보와 제안한 방법의 POS정보는 동일하다. 이 새로운 병렬 코퍼스와 기존 코퍼스를 합쳐 20000 문장쌍의 병렬 코퍼스를 생성한다. 생성된 20000 문장쌍의 병렬 코퍼스를 GIZA++를 통해 구절 번역 테이블을 얻고 명사구 추출 필터를 이용하여 병렬 어휘를 추출한다.

5. 평가 및 결과

추출된 병렬 어휘에 대한 평가는 기존 사전 용어들과 네이버 지식백과[6] 및 두피디아[7]의 웹 정보로 구축해 놓은 도메인 정답사전을 이용하였다. 그러나 정답 사전의 어휘는 제한되어 있으므로 추출된 모든 용어에 대한 정확한 정확률(precision) 값과 재현율(recall) 값을 얻을 수 없다. 그러한 문제의 보완을 위해 정답사전에 의한 평가와 정답 사전에 없는 엔트리는 수동 평가 결과로 평가를 진행하였다. 평가 대상이 될 엔트리는 코퍼스 내에서 빈도가 높아 통계 정보만으로도 정확한 번역 결과를 얻을 수 있는 어휘는 제외하고 빈도 20 이하의 엔트리들 중 명사구 필터를 통해 추출된 용어로 구성되었다. 추출 된 용어 가운데 길이 축소를 위해 참조된 병렬 어휘 사전에 포함되는 엔트리는 제외하였다.

표 7 Baseline과 제안 방법의 구 번역 테이블 크기 비교

구 번역 테이블	크 기
Baseline	43859
SLR	58757

표 8 추출된 병렬 어휘 결과

	Baseline	SLR
추출된 병렬어휘	2289	2443
다중 어절	1545	1767
단일 어절	744	676

표 9 정답사전에 의한 단일 어절 용어 추출 평가 결과

	Baseline	SLR	Baseline +SLR
Accuracy	62% (364/582)	58% (300/514)	52% (462/893)

표 10 정답사전의 의한 다중 어절 용어 추출 평가 결과

	Baseline	SLR	Baseline +SLR
Accuracy	66% (48/73)	73% (52/71)	62% (71/115)

표 11 수동평가에 의한 다중 어절 용어 추출 평가 결과

	Baseline	SLR	Baseline +SLR
Accuracy	68% (68/100)	76% (76/100)	72% (144/200)

실험 결과 Baseline과 제안방법(SLR)의 구 번역 테이블의 크기는 표 7과 같다. MOSES/GIZA++로부터 생성된 구 번역 테이블을 명사구 추출 필터를 이용하여 평가 데이터로 쓰일 용어를 추출하였다. 추출된 용어는 표 8과 같다. 추출된 용어 중 단일 어절과 다중 어절의 평가를 다르게 하기 위하여 데이터를 구분하여 평가하였다. 단일 어절의 경우 정답사전에 의한 평가만을 수행하였고 다중 어절의 경우 정답 사전에 의한 평가 엔트리가 적어 정답 사전에 의해 평가된 엔트리를 제외한 엔트리 중 임의로 뽑은 100개를 수동 평가 하였다. 정답 사전의 단일 어절 평가 결과 추출한 baseline과 SLR에서 각 0.62의 과 0.58의 정확률을 보이는 582개와 514개의 용어가 추출되었다. 다중 어절의 경우 Baseline보다 SLR에서 더 많은 정확한 엔트리를 추출하여 추출한 엔트리의 정확률 값이 0.73으로 0.66의 Baseline보다 높음을 보였다. 수동 평가의 경우 임의의 100개의 용어를 수동 평가 한 결과 각각 0.68 및 0.76의 정확률을 보였다. 자동 평가 방법에 의해 병렬 어휘를 평가한 경우 수동 평가에 의한 방법보다 정확도가 떨어짐을 알 수 있다. ‘전자 뉴스’ 코퍼스로부터 추출된 어휘는 다양한 상용적인 번역어를 포함하고 있다. 이 대역어가 정답 사전에 존재하는 대역어와 일치하지 않은 경우 정확도를 떨어뜨리게 된다. 예를 표 12에 나타내었다.

표 12 정답 사전과 매칭되지 않은 병렬 어휘

어휘	추출된 대역어	사전의 대역어
home automation system	홈오토메이션 시스템	홈자동화시스템
video memory	비디오메모리	비디오저장장치
video conference	화상회의	영상회의
security level	보안성	보안 수준

평가에 사용된 데이터 셋은 기존 방법과 SLR의 방법으로 뽑은 용어들 중 서로 중복되지 않는 용어들이 다수 포함되어 있다. 그러므로 Baseline의 방법 외에 SLR의 방법을 사용할 경우 Baseline에서 얻을 수 없는 단일 어절 및 다중 어절 용어들을 확보할 수 있음을 증명하였다. Baseline의 방법과 SLR의 방법을 병행하여 사용할 경우 Baseline만을 사용할 경우보다 한정된 코퍼스로부터 좀 더 풍부한 어휘 추출을 기대할 수 있다. 다만 단일어절과 다중어절 Baseline과 SLR의 데이터 특성이 다르고 실험 결과의 정확도 및 재현율 등의 값을 고려할 때 각 방법과 다중 어절 및 단일 어절 데이터 셋을 추출하는 과정에서 명사구 추출 필터에 최대한 정확도가 좋은 용어 선별을 위한 각 데이터에 맞는 휴리스틱 적용이 요구된다.

6 . 결 론

문장 길이가 축소된 코퍼스를 이용하여 통계 기반의 기계 번역 시스템의 구절 테이블의 용어 추출 성능을 높이는 실험과 그에 따른 결과를 제시하였다. 그 결과 베이스라인에서 얻을 수 없는 코퍼스의 용어가 다량 추출되었고 그의 정확도도 나쁘지 않음을 실험 결과에서 알 수 있다. 실험 결과를 통해 통계 기반이 기계 번역 시스템에서 한정된 코퍼스 자원을 가지고 보다 풍부한 어휘 정보의 번역을 요구할 때 제안한 방법이 도움을 줄 수

있을 것으로 기대한다. 또한 실험에 나타난 정확도와 재현률 값의 보완을 위해 용어 추출 시 사용되는 필터에 각 데이터 특성에 맞는 휴리스틱 적용이 필요하다.

참고문헌

[1] http://www.statmt.org/moses/
[2] http://www.statmt.org/moses/giza/GIZA++.html
[3] 김유섭, “지지 벡터 기계를 이용한 긴 문장의 효과적인 분할”, 10-19, 한국정보기술학회논문지, 2007.
[4] 이종훈, 이동현, 이근배, “통계적 기계 번역을 위한 변환 기반 문장 분할 방법”, 한국정보과학회 언어공학연구회 학술발표논문집, 276-281,2007.
[5] 조희영, 서형원, 김재훈, “문장 길이가 한영 통계기반 기계번역에 미치는 영향 분석”, 한국정보과학회 학술발표논문집 34(1C), 199-203, 2007.
[6] http://terms.naver.com/
[7] http://www.doopedia.co.kr/

빅데이터 기반의 오피니언 마이닝을 이용한 기업 가치 평가 시스템 개발

이정태[O1], 천민아[1], 임상우[1], 전병석[1], 김재훈[2], 한영우[3]
한국해양대학교[1], 한국해양대학교[2], 한국예탁결제원[3]
{make8286[O], dkahffk0218, dlatkddn, dmdtnsky}@naver.com, jhoon@hhu.ac.kr, kim@mail.re.kr

Developing Corporate Valuation System with Opinion Mining Based on Big Data

Jung-Tae Lee[O1], Mina Cheon[1], Sang-Woo Lim[1],
Byung-Seok June[1], Jae-Hoon Kim[2], Yeong-Woo Han[3]
[1]Korea Maritime University, [2]Korea Maritime University, [3]Korea Securities Depository

요 약

빅데이터(Big Data)는 현재 생산되고 있는 데이터 중 그 규모가 방대하고, 생성 주기가 짧으며, 수치 데이터 뿐 아니라 텍스트 이외의 멀티미디어 등 비정형화된 데이터를 포함하는 대규모 데이터를 말한다. 빅데이터를 처리하여 가치 있는 정보를 추출하는 방법에 관한 연구가 활발하게 진행되고 있으며, 이를 바탕으로 빅데이터가 다양한 분야에서 활용되고 있다. 현재 국내 주식시장에서도 빅데이터를 이용하여 기업의 투자에 활용하고 있다. 이 논문에서는 인터넷의 증권과 관련된 뉴스를 수집하여 수집된 뉴스와 주가 지수를 이용하여 기업 뉴스 평가 시스템을 개발하는 방법을 제안한다.

주제어: 빅데이터, 오피니언 마이닝, 기업 뉴스 평가

1. 서론

빅데이터(big data)는 디지털 경제의 확산으로 규모를 가늠할 수 없을 정도로 많은 정보와 데이터가 생산됨에 따라 중요 키워드로 떠오르고 있다. 빅데이터란 정보 기술의 발전과 IT의 일상화가 진행되고 있는 현재 생산되고 있는 데이터 중 그 규모가 방대하고, 생성 주기가 짧으며, 형태도 수치 데이터뿐 아니라 텍스트 이외의 멀티미디어 등 비정형화된 데이터를 포함하는 대규모 데이터를 말한다[1]. 빅데이터가 등장함에 따라 데이터를 저장, 활용, 확산 및 공유하는데 그쳤던 과거와 달리 현재는 축적된 데이터 자원을 이용한 분석과 추론을 통해 해당 데이터로부터 가치창출을 하는 방향으로 데이터의 이용 흐름이 바뀌고 있다[2]. 이런 흐름 속에 빅데이터에 관한 분석기법까지 비약적으로 발전하면서 사회의 움직임을 더욱 정확하게 분석하고 예측할 수 있게 됨에 따라 빅데이터는 국토 보안, 의료, 마케팅, 증권업계 등 다양한 분야에서 활용되고 있다.

현재 국내 주식시장에서는 빅데이터로 뉴스를 선택하여 이를 분석해서 투자에 활용하고 있다. 모 증권사의 조사 결과에 따르면 일반적으로 투자자들은 한 기업에 대한 뉴스가 증가하면 관련 기업에 크게 관심을 갖게 되고 이는 주식 거래량과 주가 변동에 영향을 미치는 것으로 나타났다[3]. 이런 현상에 주목하여 특정 기업의 뉴스를 수집 후 오피니언 마이닝(opinion mining) 결과와 주가지수 정보를 기반으로 기업 가치를 평가하는 시스템을 개발하는 것을 목표로 한다.

이 시스템은 특정 기업의 뉴스를 수집하는 웹 크롤링 에이전트와 수집된 기업뉴스에 대한 평가를 수행하기 위한 오피니언 마이닝 엔진, 기업 평가 시 활용할 수 있는 가중치를 조정할 수 있는 기능과 평가된 결과를 조회 할 수 있는 화면, 오피니언 마이닝 결과와 주가지수 정보를 기반으로 기업 가치를 평가하는 예측 엔진으로 구성된다.

이 논문의 구성은 다음과 같다. 먼저 2장에서는 이 시스템을 개발하는데 필요한 관련 연구를 살펴보고, 3장에서는 제안하는 기업 평가 시스템에 대해 설명하고, 마지막으로 4장에서 결론 및 향후 연구를 기술한다.

2. 관련 연구

2.1 오피니언 마이닝

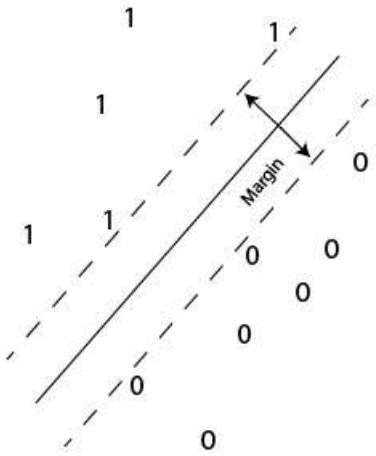
빅데이터를 분석하는 기술과 방법들은 기존 통계학과 전산학에 사용되던 데이터 마이닝(data mining), 기계학습, 자연 언어 처리, 패턴 인식 등이 있다[4]. 오피니언 마이닝은 텍스트 마이닝(text mining)의 한 분야로 소셜 미디어(social media) 등의 정형/비정형 텍스트의 긍정, 부정, 중립의 선호도를 판별하는 기술이다[5]. 일반적으로 오피니언 마이닝은 특정 주제에 대한 사람들의 주관적인 의견들을 모아 문장 단위로 분석한다. 문장 분석에서는 사실과 의견 부분을 구분한 후, 의견 부분을 토대로 긍정과 부정으로 나눈 뒤 그 강도를 측정한다.

오피니언 마이닝은 대체로 크게 특징 추출(feature extraction), 의견 분류(opinion classification), 요약 및 표현 등의 가지 세 단계로 이루어진다. 첫째, 특징 추출 단계는 오피니언 마이닝을 수행함에 있어 유용한 정보라고 판단되는 여러 특징(feature)들을 정의하고 추

출해 내는 단계이다. 이때 단순히 특징만을 추출하는 것이 아니라 주제어에 대하여 긍정적인인지 부정적인지에 대한 극성 판별도 함께 이루어진다. 둘째, 의견 분류 단계에서는 대상의 특징을 중심으로 문서를 요약하고 추출된 특징과 의견을 나타내는 어휘가 주제어에서 어떤 의미로 사용되었는가에 대한 판단 및 분류를 하는 단계이다. 셋째, 요약 및 표현단계에서는 주제어에 대한 사용자의 의견들 중 의견 성향이 밝혀진 정보들을 요약하여 전체 정보의 내용을 효율적으로 사용자에게 보여주는 단계이다[6-9].

2.2 SVM

SVM(Support Vector Machine)은 감독 기계 학습 기법(supervised learning)의 한 종류로써 데이터를 분류하는데 좋은 성능을 보인다. SVM이 좋은 서능을 보이는 가장 큰 이유는 일반적으로 학습 시에는 매우 많은 특징을 다루어야 하는데, SVM의 경우 특징의 수에 의존하지 않는 over-fitting 방지 알고리즘을 가지고 있어 많은 수의 특징 공간을 다룰 수 있다. SVM의 기본 원리는 많은 통계 학습 모델 중에서 가장 높은 성능 결과 값을 가지고 있는 모델로, 두 개의 클래스의 구성 데이터들 가장 잘 분리할 수 있는 결정함수(hyperplane)를 찾는 것이다[10-12].



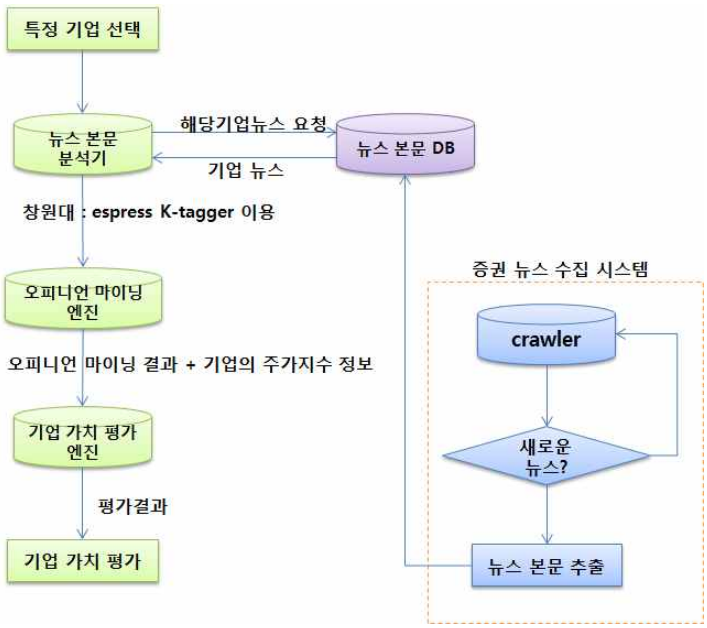
(그림 1) 선형 SVM의 결정함수[13]

이 논문에서는 긍정/부정의 2가지 클래스를 판별하고자 하며 Weka의 SVM[15]을 사용한다. 실험 데이터의 추출은 증권 뉴스 기사들을 모아서 사람이 직접 분류하여 데이터를 수집한다. 성능 평가를 위한 자질 단어의 수집도 함께 이루어졌다. 내용어의 수집은 뉴스 기사 상에서 형태소 분석을 통해 의미 있는 단어들을 선별하였으며, 문서에서 지대한 영향을 끼칠 수 있는 감정단어 선별은 국립국어원에서 제공하는 세종 말뭉치의 형용사를 기반으로 감정단어 사전을 구축하였다.

3. 기업평가 시스템

3.1 전체 시스템

이 논문에서 제안하는 전체 시스템의 동작 구조는 (그림 2)와 같다. 먼저 특정 기업의 뉴스를 수집하는 웹 크롤링 시스템을 통해 증권 사이트에서 관련 뉴스들을 수집한다. 새로운 뉴스를 수집했을 경우 html 태그를 제거하여 뉴스 본문을 추출한 후, 추출한 본문 데이터베이스에 저장한다. 사용자가 특정 기업을 선택하게 되면, 뉴스 본문이 저장되어 있는 데이터베이스에 접속하여 해당 기업의 뉴스들을 창원대에서 제공하는 형태소분석기인 Espresso-K-Tagger[16]를 이용하여 형태소 분석을 수행한다. 수행된 결과를 바탕으로 수집된 기업뉴스에 대한 평가를 위해 오피니언 마이닝 엔진을 이용하여 오피니언 마이닝을 수행하고, 그 결과와 주가지수 정보를 기반으로 기업 가치를 평가하는 예측 엔진을 사용하여 그 결과값을 화면에 조회할 수 있도록 한다.



(그림 2) 전체 시스템

3.2 오피니언 마이닝 엔진

수집한 뉴스의 본문을 형태소 분석하여 이를 긍정/부정으로 분류한다. 이를 위해 문서를 긍정/부정으로 나눌 때 일반적으로 사용하는 기계학습 알고리즘 중 SVM을 이용한다. 감정단어는 세종말뭉치를 기반으로 '좋다', '나쁘다'는 감정단어의 유의어를 기반으로 확장시켜 수집하고 내용어는 형태소 분석을 통해 분석한 내용 중 불용어를 제외한 전체 문서의 분석결과를 토대로 감정단어를 제외한 명사, 형용사, 부사, 동사의 단어들을 수집한다. 이렇게 수집한 감정단어, 내용어, 감정단어+내용어를 토대로 SVM을 이용하여 문서를 분류한다. 오피니언 마이닝에서 감정단어와 관련된 자질들의 가중치가 중요하므로, 감정단어의 경우 $TF-IDF$를 이용하여 다른 자질보다 가중치를 더 높게 부여한다.

$TF-IDF$는 문서에 단어 t가 나타난 단어의 빈도수 tf_t와 문서의 역문서 빈도수인 idf_t를 이용하여 가중치를 구하는 방법이다. idf_t를 구하는 식은 (1)과 같다.

$$idf_t = \log_2 \frac{N}{df_t} \quad (1)$$

여기서 N은 전체 문서의 수이며, df_t는 단어 t가 출현한 문서의 빈도수이다.

$TF-IDF$는 여러 문서로 이루어진 문서군이 존재할 때 단어 t가 특정 문서에서 얼마나 중요한 것인지를 나타내는 통계적 수치다. $TF-IDF$를 구하는 방법은 식 (2)와 같이 tf_t와 idf_t를 곱하는 것으로 구할 수 있다.

$$TF-IDF = tf_t \times idf_t \quad (2)$$

각 감정 단어 자질 t마다 출현 횟수를 이용하여 식 (3)을 이용하여 가중치 W_t를 부여한다. 이는 각각의 긍정과 부정의 의미에 대하여 많이 출현한 단어의 의미가 더 강하므로 높은 가중치를 부여하고, 전체에 비례하여 평준화시키는 것이다.

$$W = \frac{t_{cnt}}{N_{\max cnt}} \quad (3)$$

t는 감정 단어 자질이며 t_{cnt}는 해당 단어가 뉴스에 출현한 횟수이며 $N_{\max cnt}$는 감정 자질의 단어들 중 최대로 출현한 횟수다. 각 감정 자질의 가중치는 식 (2)와 (3)을 이용하여 구한다[14].

3.3 기업 가치 평가 엔진

오피니언 마이닝의 결과와 수집한 주가지수 정보를 바탕으로 기업 가치를 평가한다. 해당 기업과 관련된 수집된 뉴스의 오피니언 마이닝 수행결과 긍정 뉴스가 부정 뉴스보다 더 많을 경우 기업의 가치를 높게 평가하는 것을 기본으로 하며, 여기에 수집한 주가지수도 평가 항목으로 적용한다. 대체로 우량기업의 경우 뉴스의 평가가 기업 가치에 미미한 영향을 미치는 경우가 많으므로 [15], 이 경우에는 해당 기업이 가지고 있는 주가지수에 더 많은 가중치를 줘서 평가하도록 한다.

4. 결론 및 향후 연구

오피니언 마이닝은 기본적으로 영화나 특정 제품의 상품평 등 사람들의 주관이 들어간 내용에 더 적합하기 때문에, 객관적인 사실이 대부분의 내용인 뉴스의 오피니언 마이닝에는 한계점이 존재한다. 따라서 차후 뉴스를 수집할 때 해당 뉴스의 댓글들을 따로 수집하여 이를 평가 항목에 추가하는 방향으로 연구를 진행하면 지금 구성하고 있는 시스템보다 더 좋은 결과를 낼 수 있을 것으로 예상된다.

참고 문헌

[1] 정용찬, "빅데이터", 커뮤니케이션북스, pp.2, 2012.

[2] 한국정보화진흥원, 빅데이터 전략 연구센터, "새로운 미래를 여는 빅데이터 시대", pp.19, 2013.

[3] 김수진, "[증시閑담] 뉴스 빅데이터, 주식을 분석해라", Chosun Biz, http://biz.chosun.com/site/data/html_dir/2012/12/05/2012120500308.html, 2012.

[4] 정병권 외 2명, "미래사회와 빅 데이터(Big data) 기술", IT기획 시리즈, 정보통신산업진흥원, pp. 20-21, 2012.

[5] 최종후, "빅 데이터 시대가 도래한다", Korea University Sejong Campus Magazine, vol.20, pp.9, 2012.

[6] Namrata Godbole, Manjunath Srinivasaiah, and Steven Skiena, "Large-Scale Sentiment Analysis for News and Blogs," Int'l AAAI Conference on Weblogs and Social Media (ICWSM 2007), 2007.

[7] E. Boiy, P. Hens, K. Deschacht, and M. Moens, "Automatic Sentiment Analysis in On-line Text," Proceedings of the ELPUB2007 Conference on Electronic Publishing, June 2007.

[8] J. Yi and W. Niblack, "Sentiment Mining in Web-Fountain," Proceedigns of the International Conference on Data Engineering (ICDE'05), pp. 1073-1083, 2005.

[9] T. Nasukawa, J. Yi, "Sentiment analysis: capturing favorability using natural language processing," Proceedings of the K-CAP-03, 2nd International Conference on Knowledge Capture, pp. 70-77, 2003.

[10] Y. Bao, and N. Ishii, "Combining Multiple K-Nearest Neighbor Classifiers for Text Classification by Reducts." Proceeding of the fifth International Conference on Discovery Science, pp. 340-347, 2002.

[11] 이재식, 이종운, "사례기반 추론을 이용한 한글 문서분류 시스템", 경영정보학연구, 제12권, 제2호, 2002.

[12] 김진상, 신양규, "베이지안 학습을 이용한 문서의 자동분류", Journal of the Korean, Data & Information Science Society, Vol. 11, No. 1, pp 19-30, 2000.

[13] Chris Thornton, "Machine Learning - Lecture 15 Support Vector Machines", http://www.sussex.ac.uk/Users/christ/crs/ml/lec08a.html, 2011.

[14] 황재원, 고영중, "감정 분류를 위한 한국어 감정 자질 추출 기법과 감정 자질의 유용성 평가", 인지과학, 제19권, Vol.4, pp. 506-507, 2008.

[15] Weka, http://www.cs.waikato.ac.nz/ml/weka/

[16] 창원대학교 AIR 연구실, Esspreso K Tager, http://air.changwon.ac.kr/blog/2012/01/04/esspreso-pos-tagger-for-korean/

CopyCheck: 한글문서 표절검사 소프트웨어

박소영○, 장은서, 권도형, 강승식
국민대학교 컴퓨터공학부
park-soyeong@nate.com, akdangz@kookmin.ac.kr, kdhlook@naver.com, sskang@kookmin.ac.kr

CopyCheck: Korean Document Plagiarism Detection System

So-Yeong Park○, Eun-seo Jang, Do-Hyung Kwon, Seung-Shik Kang
School of Computer Science, Kookmin University

요 약

본 논문에서는 대학의 과제물이나 학위 논문 또는 회사의 입사지원서, 자기소개서와 같은 문서에 대하여 표절검사에 활용할 수 있는 소프트웨어인 CopyCheck를 설계 및 개발하였다. CopyCheck는 표절검사 방법을 빠른 검사와 정밀 검사를 두어 보다 사용자가 편리하게 사용할 수 있도록 하였다. 표절검사를 진행한 후, 전체보기와 구간보기, 표절구간 시각화의 3가지 방법을 통해 사용자가 다양한 방법으로 표절 문서를 파악할 수 있도록 도와준다. 또한, 표절검사 결과를 저장할 수 있도록 하여 사용자가 언제든지 다시 볼 수 있도록 하였다.

주제어: 표절, 한글 처리, 표절검사, 한글문서 표절검사 소프트웨어

1. 서론

21세기 정보화시대가 도래하면서 컴퓨터의 발달이 급속도로 진행되고 있다. 이로 인한 매체의 활성화로 누구나 손쉽게 정보를 공유하고 가져올 수 있게 되어 표절이 보다 빈번하게 이루어지고 있다. 특히 학교 안에서의 과제, 논문 등의 표절이 빈번하게 일어나고 있으며, 더 나아가 회사의 입사지원서 또는 자기소개서와 같은 문서에 대해서도 표절이 빈번하게 일어나고 있다. 이로 인해 표절은 사회 문제로 더욱더 크게 대두되는 추세가 지속되고 있다. 이러한 결과로 문서간의 표절 검사에 대한 필요성과 수요 또한 증가하고 있다. 그러나 수작업으로 문서간의 표절 검사를 하는 것은 인적·시간적 비용이 너무 크다는 단점이 존재한다.

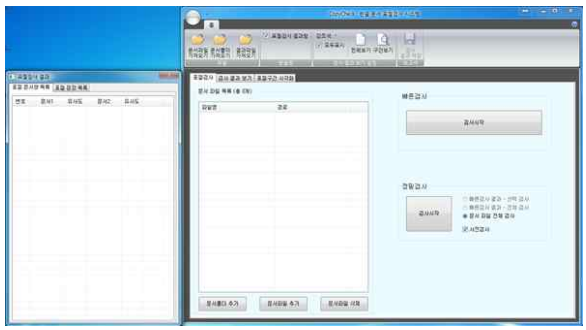
이러한 추세에 따라 한글문서 표절검사 방법에 대한 연구가 활발하게 진행되고 있으며 이미 공개되어 있는 한국어 표절 검사 소프트웨어들도 다수 존재한다. 그러나 대부분의 한국어 표절 검사 소프트웨어들은 수행시간이 지나치게 오래 걸리거나 표절의 가능성이 낮은 문장에 대해서도 발견해낸다는 문제가 있다.

본 논문은 표절 검사 대상 문서 군에서 표절이 의심되는 문서 군을 군집화 하고, 표절 의심 문서들 간의 문자열 일치구간을 빠르고 정확하게 찾아낼 수 있는 표절 검사 소프트웨어인 CopyCheck를 설계 및 개발하였다.[1] CopyCheck는 표절 검사 방법을 빠른 검사와 정밀 검사를 두어 보다 사용자가 편리하게 사용할 수 있도록 하였다. 표절검사를 진행한 후, 전체보기와 구간보기, 표절구간 시각화의 3가지 방법을 통해 사용자가 다양한 방법으로 표절 문서를 파악할 수 있도록 도와준다. 또한, 표절검사 결과를 저장할 수 있도록 하여 사용자가 언제든지 다시 볼 수 있도록 하였다.

2. CopyCheck

우리는 서론에서 소개한 이미 공개되어 있는 소프트웨어들의 문제점을 극복하고자 새로운 표절검사 소프트웨어인 CopyCheck를 설계 및 개발하였다. 본 절에서는 CopyCheck version 1.0의 구조와 기능들을 상세히 설명한다.

2.1 CopyCheck의 구조



[그림 1] CopyCheck 소프트웨어 실행 화면

CopyCheck의 초기화면은 [그림 1]과 같이 main화면과 main화면 좌측에 위치한 '표절검사 결과' 창으로 구성된다. main화면의 상단은 홈 탭으로 구성되어 있으며, 홈 탭의 하단은 표절검사, 검사 결과 보기, 표절구간 시각화로 총 3개의 탭으로 구성되어 있다. 초기 소프트웨어를 실행하면 표절검사 탭에서 시작하게 된다.

[그림 2] 홈 탭 화면

홈 탭의 메뉴는 [그림 2]와 같이 파일, 창설정, 검사

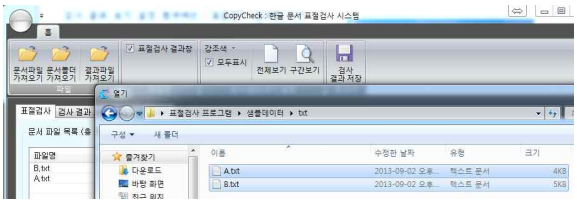
결과 보기 설정, 보고서 4개의 범주가 있다. 파일 범주에는 표절검사를 위한 문서파일 가져오기, 문서폴더 가져오기, 결과파일 가져오기를 선택하기 위한 버튼들이 있다. 창설정 범주에는 main화면의 좌측에 위치한 표절검사 결과창 활성화를 설정할 수 있는 체크박스가 있다. 검사 결과 보기 설정 범주에는 표절검사 결과를 보기 위한 강조색, 모두표시, 전체보기, 구간보기를 선택하기 위한 버튼들이 있다. 보고서 범주에는 표절검사 수행 후, 결과 파일을 저장하는 검사 결과 저장 버튼이 있다.

2.2 표절검사 문서 파일 선택

표절 검사할 문서 파일을 선택하는 방법은 파일 단위로 선택하는 방법과 폴더 단위로 선택하는 방법 2가지가 있다. 또한 파일을 선택하는 버튼이 상단과 하단에 2가지가 있으며, 상단에 위치한 버튼은 기존의 파일 목록을 없애고 새롭게 목록을 생성하는 기능이며, 하단에 위치한 버튼은 기존의 파일 목록에 추가되는 기능이다.

2.2.1 파일 단위 선택

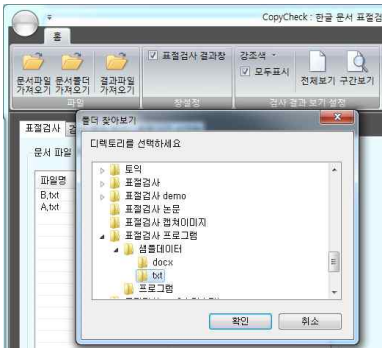
표절 검사할 파일을 파일 단위로 선택하는 방법은 상단의 문서파일 가져오기 또는 하단의 문서파일 추가 버튼을 클릭하는 것이다. 클릭 시 파일을 선택할 수 있도록 탐색기 창이 나타나며, 표절 검사할 문서 파일들을 선택하고 열기 버튼을 누르면 된다. [그림 3]은 샘플 데이터 폴더 아래의 txt폴더에서 A.txt와 B.txt를 선택한 예이다.



[그림 3] 파일 단위 선택 화면

2.2.2 폴더 단위 선택

표절 검사할 파일을 폴더 단위로 선택하는 방법은 상단의 문서폴더 가져오기 또는 하단의 문서폴더 추가 버튼을 클릭하는 것이다. 클릭 시 폴더를 선택할 수 있도록 탐색기 창이 나타나며, 표절 검사할 문서 폴더를 선택하고 확인 버튼을 누르면 된다. [그림 4]는 샘플 데이터 폴더 아래의 txt폴더를 선택한 예이다.



[그림 4] 폴더 단위 선택 화면

2.3 표절검사 모드

표절검사 모드는 빠른 검사와 정밀 검사가 있다. 빠른 검사는 검사 속도는 빠르지만 정밀 검사에 비해 표절의 심구간을 파악하는 정확도가 떨어질 수 있다. 반면 정밀 검사의 경우는 검사 속도가 느린 편이지만 표절의심구간을 파악하는 정확도가 빠른 검사에 비해 높다.

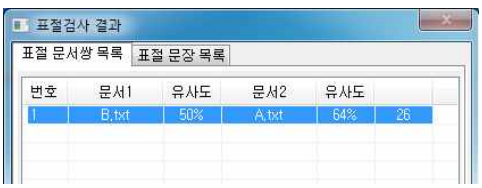
빠른 검사는 검사 후 '빠른 검사 결과-선택 검사'와 '빠른 검사 결과-전체 검사' 2가지 모드가 있다. 먼저 '빠른 검사 결과-선택 검사'는 빠른 검사를 통해 표절이 의심되는 문서 파일들을 추려내고, 사용자가 원하는 문서 파일들을 선택하여 정밀 검사하는 기능이다. 다음으로 '빠른 검사 결과-전체 검사'는 빠른 검사를 통해 추려낸 문서 파일 전체를 정밀 검사하는 기능이다. 문서 파일 전체 검사의 버튼을 체크한 후, 정밀 검사의 검사 시작 버튼을 누르면 사용자가 업로드한 문서 파일 전체에 대해 정밀검사를 진행한다.

[그림 5] 표절검사 모드

2.3 표절검사 결과보기

표절검사를 진행한 후, 표절 의심 문서 파일들이 존재하면 [그림 6]과 같이 표절검사 결과창의 표절 문서쌍 목록 탭에 한 행마다 한 쌍의 의심되는 파일들의 이름과 유사도, 의심되는 문장 수가 출력된다. 상세하게 보고 싶은 한 행을 더블 클릭하면 자동으로 표절검사 결과창은 [그림 7]과 같이 표절 문서쌍 목록 탭에서 표절 문장 목록 탭으로 이동하게 되고, main화면은 표절검사 탭에서 검사 결과 보기 탭으로 이동한다.

표절검사 결과는 사용자가 업로드한 문서 파일들 중에서 표절 의심 문서 파일들에 대해 1:1 비교 방식으로 나타내며, 결과를 보는 방법은 검사 결과 보기와 표절구간 시각화 2가지가 있다. 검사 결과 보기는 표절 의심 문서 파일들 중 각각의 의심되는 문장을 보여준다. 또한 표절 구간 시각화는 검사 결과 보기와는 달리 표절 의심 문서 파일들의 내용이 보이지 않고, 표절 의심이 되는 구간의 분포도를 보여준다. 자세한 내용은 각각의 범주에서 설명하도록 한다.



[그림 6] 표절검사 결과 - 표절 문서쌍 목록

[그림 7] 표절검사 결과 - 표절 문장 목록

2.3.1 검사 결과 보기

검사 결과 보기는 전체보기와 구간보기 2가지 방법이 있다. 이는 홈 탭 메뉴에 위치한 검사 결과 보기 설정 범주에서 선택할 수 있다. 전체보기와 구간보기는 [그림 7]과 같은 표절검사 결과창에서 각 행을 클릭하면 해당 행의 표절의심구간에 대해 비교하여 볼 수 있다. 전체보기는 [그림 8]과 같이 표절 의심 문서 파일의 내용 전체를 보여주며, 그 중 의심되는 문장을 강조색을 통해 보여준다. 반면 구간보기는 [그림 9]와 같이 표절 의심 문서 파일 중 의심되는 문장 별로 비교하여 보여준다.

[그림 8] 검사 결과 보기 - 전체보기

[그림 9] 검사 결과 보기 - 구간보기

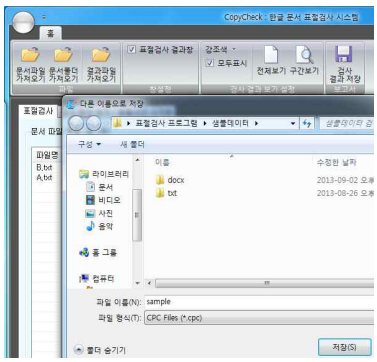
2.3.2 표절구간 시각화

표절구간 시각화는 앞서 설명한 검사 결과 보기와는 달리 실제 의심되는 문서 파일의 내용이 보이지 않는다. 다만, [그림 10]과 같이 표절 의심 문서 파일에 대해서 표절의심구간인 곳들을 표시하여 사용자가 표절의심구간의 분포를 한 눈에 확인할 수 있다는 이점이 있다.

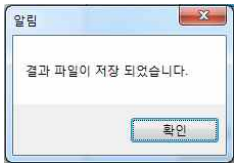
[그림 10] 표절구간 시각화

2.4 표절검사 결과 저장

표절구간 결과를 저장하고 싶다면 보고서 범주의 검사 결과 저장 버튼을 클릭한다. 이때 [그림 11]과 같이 탐색기 창이 뜨면 원하는 경로를 선택한 다음 원하는 파일 이름을 입력하고 저장 버튼을 누른다. 이와 같이 진행하면 [그림 12]와 같이 알림창이 뜬다.



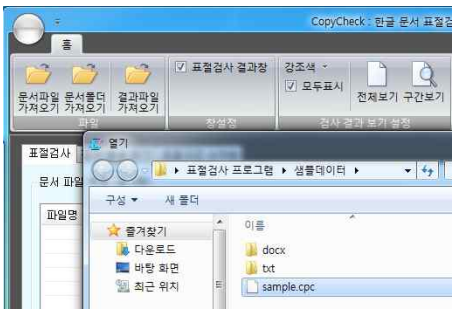
[그림 11] 표절검사 결과 저장



[그림 12] 결과 저장 알림창

2.5 표절검사 결과 불러오기

이전에 표절검사를 진행한 결과를 다시 보고 싶다면, [그림 13]과 같이 파일 범주의 결과파일 가져오기 버튼을 클릭한다. 이때 주의할 점은 이전에 검사를 진행하고 나서 검사 결과 파일(*.cpc)을 저장하였을 때 가능하다.



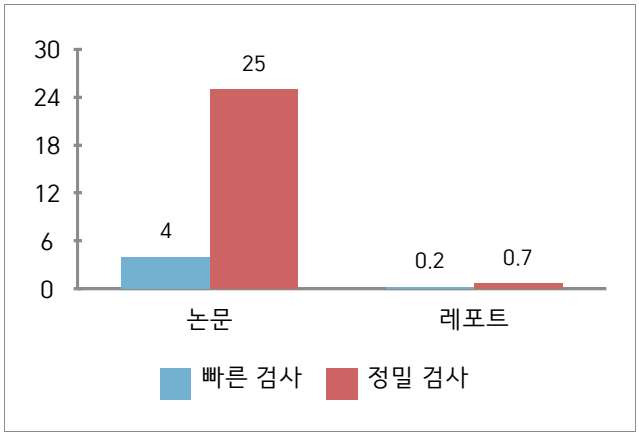
[그림 13] 표절검사 결과 파일 가져오기 화면

3. 실험 및 평가

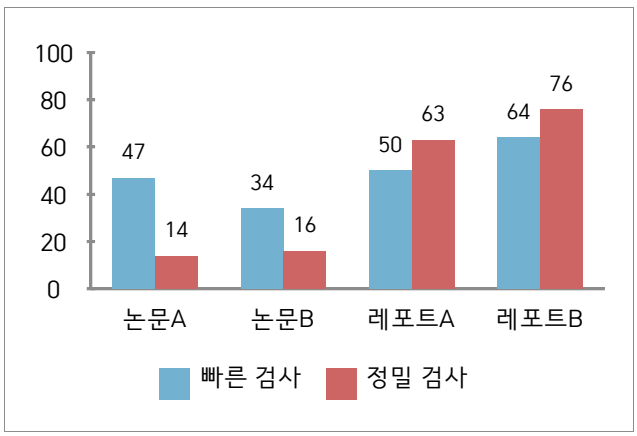
본 논문에서 설계 및 개발한 한글문서 표절검사 소프트웨어를 이용하여 실험을 진행한다. 실험은 문서를 1:1로 비교하여 진행하며 빠른 검사와 정밀 검사에 대해 각

각 표절검사 수행시간과 표절의심 문서간의 유사도를 측정한다. 실험에서 사용되는 샘플 문서의 종류는 총 2가지로 논문과 학생들의 레포트로 구성하였다.

[그림 14]는 2가지의 샘플 문서의 종류에 대해서 각각의 표절검사 모드인 빠른 검사와 정밀 검사의 수행시간을 측정하여 비교한 것이며 [그림 15]는 표절의심 문서간의 유사도를 비교한 것이다.



[그림 14] CopyCheck: 한글문서 표절검사 소프트웨어 수행시간 측정



[그림 15] CopyCheck: 한글문서 표절검사 소프트웨어 표절의심 문서간의 유사도 측정

4. 결론

본 논문에서는 한글문서 표절검사 소프트웨어인 CopyCheck를 설계 및 개발하였다. CopyCheck는 사용자의 편의성을 높이기 위해 다양한 표절검사 방법과 표절검사 결과 보기 방법을 제공한다. 또한, 표절검사 결과를 저장 또는 불러오기 기능을 두어 사용자가 언제든지 표절검사 결과를 다시 볼 수 있도록 한다.

실제 CopyCheck를 사용·도입하여 표절 의심문서를 판별해냄으로 요즘 사회문제로 대두되고 있는 콘텐츠의 지적재산권을 보호할 수 있다. 또한, 수작업으로 표절검사를 진행할 때에 비해 쉽고 빠르게, 정확하게 표절 의심문서를 판별할 수 있다는 장점이 있다. 궁극적으로는 문서간의 표절을 줄일 수 있을 것이라고 기대한다.

참고문헌

[1] 장은서, 권도형, 김낙원, 박소영, 강승식, “CopyCheck : 한국어 표절 검사 소프트웨어”, 제24회 한글 및 한국어 정보처리 학술대회, pp.117-118, 2012.

[2] 류창건, 김형준, 조환규, “한글 말뭉치를 이용한 한글 표절 탐색 모델 개발”, 정보과학회논문지: 컴퓨팅의 실제 및 레터 제14권 제2호, pp.231-235, 2008.

[3] 지혜성, 조준희, 임희석, “한국어 문장 표절 유형을 고려한 유사 문장 판별.”, 컴퓨터교육학회논문지 제13권 제6호, pp.79-89, 2010.

[4] 안병렬, 김문현, “문서를 위한 표절 탐지 소프트웨어에 관한 연구.”, 한국퍼지 및 지능소프트웨어학회 2006 년도 춘계학술대회 학술발표논문집 제16권 제1호, pp.413-415, 2006.

[5] 박선영, 조환규, “성분 정렬을 이용한 한글 유사 문서 탐색 방법.”, 한국정보과학회 2011 한국컴퓨터종합학술대회 논문집 제38권 제1호, 2011.

연관 어휘 추출을 통한 질의어 관련 이슈 탐지

김제상[0], 김동성, 조효근, 이현아
금오공과대학교
oiu124@naver.com, kaiserangel@live.co.kr, whitesky0109@naver.com, halee@kumoh.ac.kr

Query Related Issue Detection using Related Term Extraction

Je-Sang Kim[0], Dong-Sung Kim, Hyo-Geun Jo, Hyun-Ah Lee
Kumoh National Institute of Technology, Dept. of Computer Software Engineering

요 약

근래 트위터와 페이스북 등의 SNS(Social Network Service)에서 일반 대중의 관심사나 트렌드 등의 이슈를 탐지하는 많은 연구가 이루어지고 있다. 본 논문에서는 검색어에 대한 연관 어휘 추출을 통해 검색어에 연관된 이슈나 화제를 트위터에서 추출하기 위한 방법을 제안한다. 본 논문에서는 연관성이 높은 단어는 서로 가깝게 발생할 것으로 기대하고, 단어 간 거리가 가까울수록, 공기빈도가 높을수록 커지는 단어연관도 계산법을 제안한다. 연관도 값이 임계치를 넘는 어휘를 연관 어휘로 보고 네트워크의 형태로 관련 이슈를 제시한다.

주제어: 연관 어휘, 인접도 행렬, 질의어 관련 이슈, 이슈 자동 탐지, SNS.

1. 서론

SNS와 스마트 디바이스의 보급으로 인하여 실시간으로 생성되는 비정형 데이터가 급격히 늘어나고 있다. 이러한 빅데이터에 대해서는 실시간으로 생성되는 대량의 데이터를 빠르게 처리하기 위한 방법이 필요하다. SNS에 대해 근래 활발하게 시도되고 있는 분야로 실시간 이슈 탐지가 있다. [1]에서는 빅 인터렉션 데이터에서 이슈를 중심으로 현재를 모니터링하고 미래를 예측하기 위한 실제 개발 사례를 소개하고 있다.

본 논문에서는 SNS 중 트위터를 기반으로 사용자가 입력한 질의어에 관련된 이슈를 자동으로 탐지하는 방법을 제안한다. 키워드에 대한 관련 이슈를 탐지하기 위해 연관 어휘를 추출한다. 연관 어휘 추출에서는 단어 간 거리에 반비례하고 공기 빈도에 비례하는 단어 간 인접도의 합으로 연관도를 구한다.

2. 기존 연구

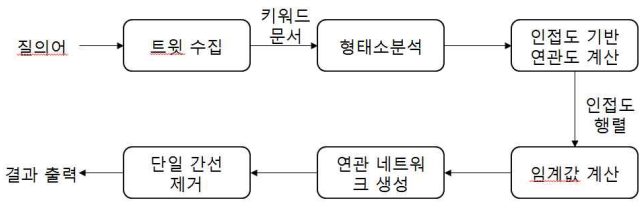
SNS에서의 이슈 또는 토픽 추출은 주로 어휘 빈도의 시간적 추이를 활용한다. 이 중 [2]와 [3]은 용어의 시간적 추이를 기준으로 권위성 등의 자질을 활용하여 떠오르는 토픽을 파악하기 위한 방법을 제안하였다. [4]에서는 특징적인 트렌드 추출을 위해 변동성, 지속성, 안정성, 누적량의 속성을 활용하여 트렌드 순위 결정 방법을 제안하여, 트렌드 탐지를 위한 방법을 제안하였다.

키워드에 대한 연관성을 통해 SNS의 비정형 문서들을 분석하여 정보를 제공해주는 서비스로는 다음소프트의 소셜 인사이트가 있다[5]. 이 서비스는 복합명사 검출과 단어 속성 등의 추출을 위해서 수작업을 필요로 하여 높은 비용이 필요하며, 최신 관련어나 복합어가 추출되지 않는다는 단점이 발견된다.

연관 단어 추출은 자동 시소러스(thesaurus) 구축이나 정보검색에서의 질의어 확장 등의 다양한 목적으로 연구되어 왔다. [6]은 뉴스에서 연관 인물명을 제시하기 위하여 문장 내 공기 어휘에 기반한 변형된 TF-IDF와 연관 규칙 마이닝을 이용한 방법을 제시하였다. [7]에서는 공기 어휘의 인접도와 빈도, IDF를 결합하여, 질의어 확장을 위한 연관어휘 추출 방식을 제안하였다. 이러한 방식은 단일 문서내 정보가 아닌 문서간 정보인 역문서빈도(IDF)를 사용하기 때문에, 대량의 실시간 문서가 발생하는 SNS환경에는 적절하지 않을 수 있다.

3. 연관 어휘를 이용한 이슈 탐지 시스템

본 논문에서는 질의어에 대한 관련 이슈 어휘를 실시간으로 추출하기 위한 방법을 제시한다. [그림 1]은 본 연구에서 제안하는 질의어에 대한 연관 어휘 추출의 단계를 보인다. 본 연구에서는 한국어 형태소 분석기를 사용하여 문서를 분석하고 형태소 분석 결과 중 명사만을 연관 어휘 후보로 사용한다.



[그림 1] 질의어에 대한 관계 명사 추출의 플로우차트

3.1 인접도에 기반한 연관도 측정

본 논문에서는 문장 길이가 짧은 SNS의 문장 특성에 맞게 [8]에서 사용한 공기 어휘간 거리 역수 즉, 인접도를 사용하여 어휘 연관도를 측정하고자 한다. 연관도 측정에서는 공기 어휘 간 행렬 모델을 제안하여, 문맥의

방향성과 어휘 간 연접 여부를 효율적으로 표현한다.

키워드 문서 내 문장 s에서 지정된 문맥의 크기 $windowSize$ 안에서 발생하는 i번째 명사 w_i와 j번째 명사 w_j간의 인접도 $proximity_s(w_i, w_j)$를 1/(j-i)로 구한다. 인접도로 두 단어 사이의 거리 $j-i$의 역수를 사용하여 인접한 단어는 1에 가까운 값을, 인접도가 떨어지는 값은 0에 가까운 값을 얻는다. 이때 $j-i$의 값이 1보다 크고 $windowSize$보다 작도록 지정하여 문맥의 우측 방향에서 문맥 크기 이내의 공기 어휘만을 고려한다. 또한 w_i와 w_j가 같은 경우를 배제하여 같은 단어가 중복적으로 발생하는 경우에 의한 노이즈를 제거한다. 검색된 문서 전체에서의 인접도 $proximity(w_i, w_j)$는 전체 키워드 문서에 포함된 문장에 대한 인접도의 합으로 구한다.

얻어진 인접도를 기반으로 단어 간 연관도를 구한다. 단어 w_i와 w_j 사이의 연관도 $relatedness(w_i, w_j)$는 $proximity(w_i, w_j) + proximity(w_j, w_i)$로 구하여, 문맥의 우측 방향과 좌측 방향에 대한 인접도를 합산한다.

[표 1] '싸이'에 대한 연관 행렬 추출의 예

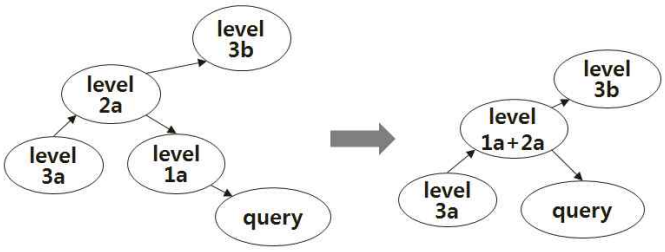
	임윤택	부담	싸이	전액	장례비용	대통령	취임식	당신	멋쟁이
임윤택	0	266.8	28	332.6	809.8	0	0	71.6	71.6
부담	0.2	0	28.4	0	0	0	0	358	143.6
싸이	441.4	90.2	0	31.2	13	0.8	8.8	71.6	0
전액	0.2	933	15.6	0	0	0	0	0	0
장례비용	4.4	307.2	28.2	638.6	0	0	0	0	0
대통령	0	0	7.4	0	0	0	384.4	0	0
취임식	0	0	18.2	0	0	0	0	0	0
당신	0	0	1.4	0	0	0	0	0	358
멋쟁이	0	0	1	0	0	0	0	0	0
단독	174.8	86.6	0	87.4	87.4	0	0	0	0

〔표 1〕은 2013년 2월 15일 기준으로 추출한 '싸이'에 대한 키워드 문서에서의 $windowSize=4$에서의 인접도 행렬을 보인다. $relatedness$(싸이, 임윤택)은 $proximity$(싸이, 임윤택) + $proximity$(임윤택, 싸이) = 441.4 + 28로 구할 수 있다. 인접도 행렬에서는 명사 간의 연관성의 경로를 쉽게 파악할 수 있다.

3.2 연관 어휘 판별 및 네트워크 생성

제안하는 시스템에서는 키워드 문서에서 인접도를 이용하여 얻어진 어휘간 연관도를 계산하고, 질의어와 연관도가 높은 단어를 '연관 어휘'로 결정하여 사용자에게 제시한다.

질의어 $query$에 대해서 연관도 $relatedness(query, w_i)$가 1차 연관 어휘 임계값 이상인 단어 w_i를 구하고, 구해진 w_i에 대해서 연관도 $relatedness(w_i, w_j)$가 2차 연관 어휘 임계값 이상인 w_k를 찾는다. 얻어진 1차 연관 어휘 w_i, 2차 연관 어휘 w_j, 3차 연관 어휘 w_k에 대해서 〔그림 2〕의 왼쪽과 같은 그래프를 구한다. 그림에서 level1은 1차, level2는 2차, level3는 3차 연관 어휘를 표시한다. 네트워크는 인접도 행렬의 좌우 문맥의 방향성에 따라 방향 네트워크의 형태로 생성한다. 〔그림 2〕에서 level1a는 level2a 이외의 연결 간선이 없는 단일 간선을 구성한다. 〔표 1〕의 '당신'과 '멋쟁이', '대통령'과 '취임식' 같은 단일 간선에 대해서, 방향 그래프의 간선 방향을 고려하여 명사를 조합하면 '당신 멋쟁이', '대통령 취임식'과 같은 복합어를 구성할 수 있다. 시스템에서는 단일 간선을 통합하고 2차 관계까지 만을 남겨 〔그림 2〕의 오른쪽과 같은 결과 네트워크를 구성한다.



[그림 2] 단일 간선과 3차 관계를 제거한 결과 네트워크

4. 실험 및 평가

4.1 질의어 관련 이슈 추출 시스템

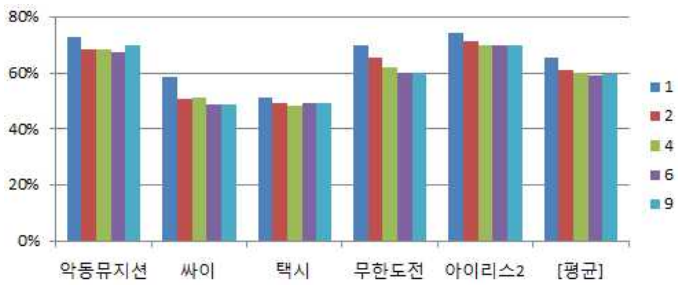
시스템에서는 사용자 가독성을 고려하여 1차와 2차 연관 어휘로 최대 10개의 결과를 제공한다. 최신성이 중요한 이슈의 특성에 맞추어 트위터와 동일하게 시스템도 7일간 자료만을 저장한다. 형태소 분석기로 [9]를 사용하고 트위터 검색에서는 [10]을 사용하였다.

제안한 방식의 성능을 평가하기 위해 2013년 2월 14일부터 2월 20일까지의 기간 중 5개 키워드(악동뮤지션, 싸이, 택시, 무한도전, 아이리스2)를 중심으로 평가를 시행하였다.

4.2 문맥 크기에 따른 성능 평가

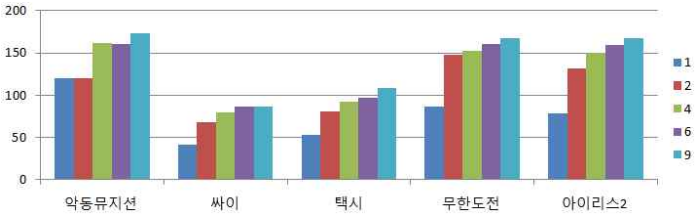
문맥의 크기에 따른 성능을 비교하기 위한 실험을 수행하였다. 10개의 키워드를 대상으로 실험군을 형성하기 위해 연관 어휘 분석 모듈의 1차와 2차 연관 어휘의 가중치를 1% 즉 0.01으로 고정하고, $windowSize$를 1, 2, 4, 6, 9로 변화시켜, 인접 체언의 범위에 따라 연관 어휘 추출에 어떤 영향을 미치는지 살펴보았다.

[그림 3]은 문맥 크기에 따른 시스템의 정확도를 보인다. 이슈성에 근거하여 엄격하게 수동 평가한 결과에서, 상위 10위까지에서 대부분 정확한 연관 어휘가 추출된 것으로 볼 수 있다. 문맥의 크기를 키우더라도 추가적인 연관 어휘가 추출되지 않아 성능 차이가 크지 않았다. 또한 평균적인 정확률이 60%이상으로 실용성 있는 이슈 추출이 가능함을 볼 수 있다.



[그림 3] windowSize에 따른 연관 어휘 정확도

보다 정확한 분석을 위하여 연관도 상위 50개를 추출한 경우를 분석하였다. 결과는 [그림 4]와 같다. 추출되는 연관 어휘 개수는 최대 7일 * 50 = 350개가 나올 수 있으나, 지속성이 큰 이슈들로 최대 230여개가 추출되는 것을 볼 수 있다. 결과에서는 windowSize가 증가함에 따라 추출되는 연관 어휘 개수가 증가하는 모습이 보다 뚜렷하다.



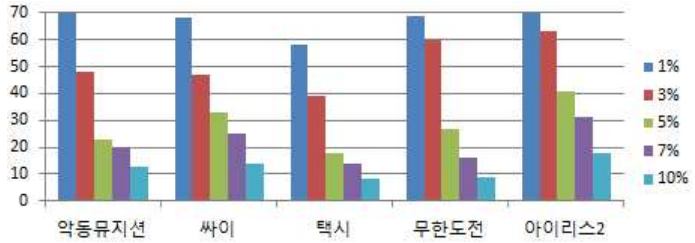
[그림 4] 상위 50개 대상 windowSize에 따른 연관 어휘 개수

상위 50개에 대한 정확도 분석에서는 수동 평가로 인해 키워드 '아이리스2'에 대한 결과만을 분석하였다. 결과에서는 문맥 크기가 1인 경우 80%의 정확도를, 문맥크기가 2~9의 경우 61%~65%의 정확도를 보여 문맥크기를 키워 연관 어휘 개수가 많아지는 만큼, 부정확한 연관 어휘도 많이 추출되는 것으로 나타났다. 큰 문맥에 의해서는 어휘에 노이즈가 포함될 여지가 커지는 동시에 인접도 행렬에 저장되는 값이 많아져서 시스템 속도가 느려질 수 있으며, 작은 문맥에 의해서는 명확한 연관 어휘만 추출되지만 폭넓은 연관 어휘를 추출할 수 없는 문제가 있어, 응용 분야에 맞는 적절한 문맥 크기의 선택이 필요한 것으로 분석되었다.

4.3 연관도 임계치에 따른 성능 평가

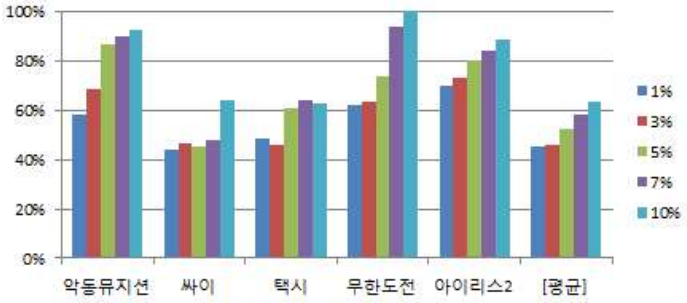
연관 어휘 추출을 위한 임계치에 따른 실험을 수행하였다. 실험 환경에서 제시한 10개의 키워드를 대상으로 *windowSize*의 값을 4로 고정시키고 1차 연관 어휘 추출을 위한 가중치를 1, 3, 5, 7, 10%로 변화함에 따른 연관 어휘 발생 빈도를 분석하였다.

[그림 5]는 가중치 변화에 따른 연관 어휘 추출 개수를 보여준다. 가중치의 임계치 값을 높게 할수록 연관 어휘 개수가 점차적으로 줄어든다는 것을 알 수 있다.



[그림 5] 가중치에 따른 상위 10개 연관 어휘 개수

임계값에 따른 정확도에 대한 평가는 [그림 6]과 같다. 결과는 키워드 5개에 대하여 가중치 변화에 따른 이슈성이 있는 사건의 비율 즉, 정확도를 나타낸다. 결과는 평균 57%~82%의 연관 어휘 추출의 정확도를 보여, 인접도 행렬을 이용한 빠른 처리에도 불구하고 실용성 있는 결과를 얻을 수 있었다. 또한 가중치와 임계치가 증가함에 따라 연관 어휘 개수가 줄어들지만 연관성이 높은 사건을 추출하는 정확도는 증가하는 것으로 나타났다. 이는 임계치가 높아 추출되는 연관 어휘 개수가 줄어들어 발생하는 효과로 볼 수 있다. [그림 5]에서 가중치가 1%에서 3%로 증가되는 경우 연관 어휘 개수는 20% 정도 감소하는 것에 비해, [그림 6]에서는 정확도의 차이는 크지 않은 것으로 나타났다. 또한, 가중치가 1%에서 10%로 증가되는 경우 추출되는 연관 어휘가 1/10의 수준으로 감소하는 것에 비해, 정확도는 20% 정도 증가하는 것으로 파악되었다. 문맥 크기와 마찬가지로 응용 분야에 맞는 가중치의 설정이 필요할 것으로 보인다.



[그림 6] 가중치에 따른 연관 어휘 정확도

5. 결론

본 논문에서는 질의어에 대한 트윗 문서에서의 연관 어휘 추출 방식을 제안하였다. 연관 어휘 추출에서는 질의어와 가까운 거리에서 자주 발생하는지의 여부가 중요하다는 점에 착안하여, 인접도의 합으로 단어 간 연관도를 얻었다. 형태소 분석된 문서를 순차적으로 한번만 탐색하면서 인접도 행렬을 구할 수 있어 빠른 시간 안에 결과를 얻을 수 있었다. 또한, 연관 관계에 따른 네트워크 형성 후 간선 조정을 통해 복합어를 추출하여 비교적 정확하면서 손쉬운 방법으로 동적으로 생성되는 복합어 처리 결과를 얻을 수 있었다. 실험에서는 임계치와 문맥의 크기에 따른 영향성을 분석하였다. 전반적으로 높은 성을 보여 비교적 간단한 방법을 사용하는데도 불구하고 실용성 있는 결과를 얻을 수 있었다.

참고문헌

[1] 류법모, 김현진, 김현기, 박상규, “심층 언어분석 기반 소셜미디어 이슈 탐지 및 모니터링 기술”, 한국정보과학회지, 제30권, 제6호, pp. 47-58, 2012.

[2] Mario Cataldi, Luigi Di Caro, Claudio Schifanella, "Emerging Topic Detection on Twitter based on Temporal and Social Terms Evaluation", Proceedings of the 10th International Workshop on Multimedia Data Mining at KDD, 2010.

[3] Michael Mathioudakis, Nick Koudas, "TwitterMonitor: trend detection over the twitter stream", Proceedings of the ACM SIGMOD International Conference on Management of data, pp. 1155-1158, 2010.

[4] 오흥선, 최윤정, 신욱현, 정윤재, 맹석현, "자동 트

렌드 탐지를 위한 속성의 정의 및 트렌드 순위 결정 방법", 정보과학회논문지 : 소프트웨어 및 응용, 제36권, 제3호, pp. 236-243, 2009.
[5] 다음소프트, "http://insight.some.co.kr/searchKeywordMap.html"
[6] 김한준, 장재영, "연관규칙 마이닝을 활용한 뉴스기사 키워드의 연관성 탐사", 한국인터넷방송통신학회 논문지, 제11권, 제6호, pp. 63-71, 2011.
[7] Tetsuya Oishi, Shunsuke Kuramoto, Tsunenori Mine, Ryuzo Hasegawa, Hiroshi Fujita, Miyuki Moshimura, "A Method for Query Expansio Using the Related Word Extraction Algorithm", IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology, 2008.
[8] 정석팔, 임성현, 전진형, 김병만, 이현아, "요약문을 이용한 웹 검색 결과 군집화", 정보과학회논문지:데이타베이스, 제39권, 제5호, pp.321-331, 2012.
[9] Lucene Core 4.0 and SolrTM 4.0 Available, “http://lucene.apache.org”, 12 October 2012
[10] 트윗 검색 시스템, "https://dev.twitter.com/“

CRFs를 이용한 의존구조 구문 레이블링

정석원○, 최맹식, 김학수
강원대학교, IT대학, 컴퓨터정보통신공학전공
{nlpsw, nlpmschoi, nlpdrkim}@kangwon.ac.kr

Labeling Dependency Structures using CRFs

Seokwon Jeong○, Maengsik Choi, Harksoo Kim
Program of Computer and Communication Engineering,
College of Information Technology, Kangwon National University

요 약

본 논문에서는 의존구조 분석 결과로부터 구문 레이블을 생성하는 방법을 제안한다. 제안 시스템은 의존구조 분석 결과의 의존소-지배소 쌍에 대해 자질을 생성하고, 문장 단위로 CRFs를 이용하여 구문 레이블을 부착한다. 실험을 통해 90.8%의 정확도를 보였고, 구문 레이블이 없는 의존구조 시스템의 후처리로 사용 가능하다.

주제어: 구문 레이블, 의존 구조 레이블

1. 서론

구문 분석은 문장의 구조를 분석하는 것으로, 크게 구구조 분석과 의존구조 분석으로 나누어진다. 한국어의 경우, 영어와는 달리 비교적 자유로운 어순과 문장 구성 성분의 빈번한 생략 등으로 인해 의존구조 분석이 더 적합하다[1,2]. 의존구조 분석은 문장을 성분의 조합으로 간주하는 구구조 분석과는 달리, 문장을 구성하는 어절들의 의존관계를 파악하는 것으로, 각 어절의 지배소를 찾는 과정이다.

오진영 외[3]는 구문 레이블을 CRFs(Conditional Random Fields)를 이용하여 부착하고 의존구조 생성에 사용하였다. 그러나 일부 의존구조 분석 시스템은 구문 레이블을 사용하지 않으므로 구문 레이블이 생성되지 않는다[4].

본 논문에서는 의존구조 분석 결과로부터 구문 레이블을 생성하는 방법을 제안한다. 본 논문의 구성은 다음과 같다. 2장은 본 논문에서 제안하는 시스템을 설명하고 3장은 실험 및 분석, 4장에서 결론과 향후 연구 방향을 제시하며 끝을 맺는다.

2. 제안 시스템

본 논문에서 제안하는 시스템은 [그림 1]과 같은 의존구조 분석 결과를 입력으로 하여 구문 레이블을 부착하는 것이다.

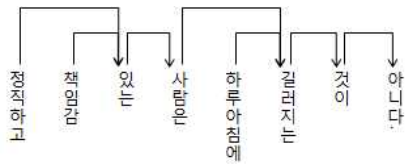


그림 1. 의존구조 분석 결과

[그림 1]의 문장으로부터 구문 레이블 부착 대상이 되는 의존소-지배소 쌍에 대하여 [표 1]의 자질을 추출한다.

표 1. CRFs 자질 예

자질	구문레이블
1:하 2:XSA 3:고 4:EC 5:있 6:VV 7:는 8:ETM 9:EC-ETM	기타
1:책임감 2:NNG 3:- 4:- 5:있 6:VV 7:는 8:ETM 9:NNG-ETM	주어
1:있 2:VV 3:는 4:ETM 5:사람 6:NNG 7:은 8:JX 9:ETM-JX	체언 수식어
1:사람 2:NNG 3:은 4:JX 5:지 6:VX 7:는 8:ETM 9:JX-ETM	주어
1:하루아침 2:NNG 3:에 4:JKB 5:지 6:VX 7:는 8:ETM 9:JKB-ETM	용언 수식어
1:지 2:VX 3:는 4:ETM 5:것 6:NNB 7:이 8:JKC 9:ETM-JKC	체언 수식어
1:것 2:NNB 3:이 4:JKC 5:아니 6:VCN 7:다 8:EF 9:JKC-SF	보어
1:아니 2:VCN 3:다 4:EF 5:아니 6:VCN 7:다 8:EF 9:SF-SF	기타

[표 1]에서 각각의 자질은 [표 2]와 같다. 의존구조는 어절 단위로 이루어지므로 어절을 내용어와 기능어로 분리하고 자질을 추출한다. 내용어는 21세기 세종계획 형

* 이 논문은 2013년도 정부(교육과학기술부)의 재원으로 한국연구재단의 지원을 받아 수행된 기초연구사업임(2013R1A1A4A01005074) 또한 본 연구는 지식경제부 및 한국산업기술평가관리원의 산업융합원천기술개발사업(정보통신)의 일환으로 수행하였음. [10041678, 다중영역 정보서비스를 위한 대화형 개인 비서 소프트웨어 원천 기술 개발]

태소 태그에서 용언, 체언, 수식언, 독립언에 해당하며, 기능어는 관계언, 의존형태, 기호에 해당한다. 그리고 의존형태 중에서 접미사는 내용어에 포함된다.

표 2. 자질 집합

자질 번호	자질
1	의존소 내용어 어휘
2	의존소 내용어 품사
3	의존소 기능어 어휘
4	의존소 기능어 품사
5	지배소 내용어 어휘
6	지배소 내용어 품사
7	지배소 기능어 어휘
8	지배소 기능어 품사
9	의존소, 지배소의 마지막 품사

[표 1]과 같은 자질을 이용하여 CRFs를 모델을 생성한다. CRFs는 조건부 확률을 최대로 하는 무방향성 그래프 모델이다[5]. 입력열 $X = x_1 x_2 \cdots x_n$, 상태열 $T = t_1 t_2 \cdots t_n$가 주어지고, 가중치 $\Lambda = \{\lambda, \ldots\}$가 주어졌을 때, CRFs에서는 조건 확률로 식 (1)과 같이 정의된다.

$$P(T|X) = \frac{1}{Z(X)} exp(\sum_{i=1}^{n} \sum_{k} \lambda_k f_k(t_{i-1}, t_i, x, i)) \quad (1)$$

여기서 $Z(X)$는 확률 값으로 만들어 주는 정규화 값이고, $f_k(t_{i-1}, t_i, x, i)$는 자질 함수이다. 또한 λ_k는 각 자질에 대한 가중치를 나타낸다. k는 k번째 자질이며, 자질 함수는 현재 시간 i에 대한 관측열 x, 상태 변이 $t_{i-1} \rightarrow t_i$에 대해서 전이의 향상을 측정할 수 있다. 매개변수들은 주어진 입력 열과 이에 대응하는 상태열에 대한 조건부 확률이 최대화하는 최대 우도(maximum likehood)에 의해서 추정된다[3]. 본 연구에서는 [6]의 CRFs를 사용하였다.

3. 실험 및 분석

3.1 실험환경

실험에서는 21세기 세종계획 구문 말뭉치를 사용하였다. 구구조 말뭉치를 의존구조 말뭉치로 변경하였고, 구문 레이블 부착 실험을 위해서 구문 분석 태그 중 기능 태그만 사용하였다. 기능태그가 없는 어절은 ETC로 분류하였다. 전체 71,091문장(844,766어절) 중 구문 레이블을 학습하기 위해서 66,454문장(778,938어절)을 사용하였고 평가를 위해서 4,637문장(65,828어절)을 사용하였다.

실험 결과의 측정을 위하여 정확률을 사용하였다. 평가 척도는 식 (2)와 같다.

$$\text{정확률} = \frac{\text{시스템이 올바르게 추출한 정답 수}}{\text{시스템이 추출한 정답 수}} \quad (2)$$

3.2 실험 및 분석

제안한 시스템의 성능 평가를 위해 구문 레이블별로 정확률을 측정하였다. 측정한 정확률은 [표 3]과 같다.

표 3. 성능

구문 레이블	정확률	빈도
주어	94.85	7,625
목적어	95.30	5,829
보어	79.61	1,373
체언 수식어	99.66	12,353
용언 수식어	93.32	7,704
접속어	90.69	1,278
독립어	75.00	32
기타	97.97	29,634
전체	90.80	65,828

[표 3]에서 전체 정확률은 90.80%이지만, 구문 레이블의 빈도에 따라 성능 차이가 크게 나타남을 알 수 있다. 특히 독립어의 빈도는 32, 정확률은 75%로 평균에 비해 매우 낮은 성능을 보였다.

4. 결론

본 논문에서는 의존구조 분석 결과로부터 구문 레이블을 부착하였다. 실험을 통해 한국어 문장에서 구문 레이블의 빈도에 따른 성능의 차이를 확인하였다.

향후 연구로서는 빈도가 낮은 구문 레이블의 성능을 향상시킬 수 있는 자질에 대한 추가 연구를 통해 전체 성능을 향상 할 수 있을 것이라 기대한다.

참고문헌

[1] J. Nivre, “An efficient algorithm for projective dependency parsing”, In Proc. of IWPT, pp.149-160, 2003.

[2] 김성용, 이공주, 최기선, “복합 레이블을 적용한 한국어 구문 규칙”, 한국정보과학회논문지: 소프트웨어및 응용, 제 31권, 제 2호, pp.235-244, 2004년 2월.

[3] 오진영, 차정원, “다단계 구단위화를 이용한 고속 한국어 의존구조 분석”, 한국시뮬레이션학회 논문지, Vol19, No.1, pp.103-111, 2010년 3월.

[4] 이용훈, 이종혁,. “SVM을 이용한 결정적 한국어 구문분석”, 한국어정보학 제10권 제2호, pp.7-14, 2008년 12월.

[5] J. Lafferty, A. McCallum, F. Pereira, “Conditional random fields: Probabilistic models for segmenting and labeling sequence data”, Proceedings. 18th International Conference on Machine Learning, pp.282-289, 2001.

[6] McCallum, Andrew Kachites., “MALLET: A Machine Learning for Language Toolkit”, http://mallet.cs.umass.edu. 2002.

음절 n-gram 기반의 미등록 어휘 추정기 구현

신준수, 홍초희
샤인웨어소프트
jsshin@shineware.co.kr, chhong@shineware.co.kr

Out of Vocabulary Word Extractor based on a Syllable n-gram

Junsoo Shin, Chohee Hong
SHINEWARE SOFT

요 약

다양한 콘텐츠가 생성됨에 따라 신조어 및 미등록어도 다양한 형태로 나타나고 있다. 이러한 신조어 및 미등록어는 텍스트 처리 단계에서 오분석 되어 성능 저하의 원인이 된다. 본 논문은 이러한 문제를 해결하기 위해서 대량의 문서로부터 신조어 및 미등록 어휘를 추정하는 방법에 대해서 제안한다. 제안 방법은 대량의 문서로부터 음절 n-gram을 추출한 뒤, 각 n-gram에서 n을 한음절 축소 및 확장 시켜, (n+1)gram, (n-1)gram을 추가적으로 추출한다. 추출된 음절 n-gram을 기준으로 (n+1)gram, (n-1)gram과의 빈도 차이를 계산하여 빈도차가 급격하게 발생하는 구간을 신조어 및 미등록 어휘로 추정한다. 실험 결과 신조어 뿐만 아니라 트위터, 미투데이 등과 같은 도메인에 종속적인 미등록 어휘도 추출되는 것을 확인할 수 있었다.

주제어: 미등록어 추정, 음절 n-gram, 키워드 추출

1. 서론

최근 다양한 분야에서 새로운 콘텐츠(Content)가 생성됨에 따라 신조어 및 미등록어도 다양한 형태로 나타나고 있다. 이러한 신조어 및 미등록어는 텍스트 전처리 단계에서 오분석되어 상위 시스템의 전반적인 성능 저하로 이어진다. 본 논문에서는 이러한 문제를 해결하기 위해 대량의 말뭉치로부터 신조어 및 미등록어를 추출하기 위한 방법을 제안한다. 본 논문의 2장에서 신조어 및 미등록어와 관련된 연구에 대해서 소개하고 3장에서 제안 방법에 대해서 설명한다. 4장에서는 제안 방법에 대한 성능 실험과 결과를 평가하고 마지막으로 5장에서 결론을 맺는다.

2. 관련연구

신조어 및 미등록어 처리를 위한 연구는 전문 용어 추출과 형태소 분석기의 미등록어 추정 연구로 분류 할 수 있다. 전문 용어 추출 연구는 제한된 영역에 대하여 CRFs, SVM 등과 같은 기계 학습 방법 등을 이용하여 용어를 추출하는 방법이다[1][2]. 기존 미등록어 추정 연구는 형태소 분석기 등과 같은 사전 정보를 활용한 시스템에서 사전에 등재되지 않은 어휘가 출현할 때 이를 추정하는 방법이다[3]. 또한 successor variety를 이용하여 대량의 코퍼스로부터 어근을 추출할 수 있는 방법이 있다. 그러나 기계 학습을 이용한 방법은 영역에 따라 서로 다른 학습 데이터가 필요하다는 단점이 있으며, 미등록어 추정은 사용자가 직접 사전 정보를 관리해야한다는 단점이 있다. 또한 successor variety는 신조어와 같이 어근이 명확하지 않은 분야에 적합하지 않다는 단점이 있다.

본 논문에서는 기존 연구의 단점을 보완하기 위해 대량의 말뭉치로부터 다양한 형태의 어휘 후보를 자동으로 추출하는 방법을 제안한다.

3. 미등록 어휘 추정 시스템 구현

문서 내에서 자주 표현되는 어휘는 앞, 뒤에 다양한 활용이 가능하다. 예를 들어 “아이폰”과 같은 어휘는 “아이폰은”, “아이폰을”, “아이폰에서”등과 같은 활용형 어휘가 나타난다. 이 때 전체 말뭉치에서 빈도수를 살펴보면 “아이폰”이 “아이폰*”보다 훨씬 더 높은 빈도를 갖는다. 따라서 이 빈도 차이를 이용하여 “아이폰”을 하나의 신조어 및 미등록어로 추정 할 수 있다. 본 논문에서는 이러한 현상을 통하여 미등록 어휘를 추정하는 시스템을 구현한다.

그림 1은 본 논문에서 제안하는 미등록 어휘 추정 시스템의 구조도이다. 다량의 문서로부터 음절 n-gram과 wildcard를 적용한 음절 n-gram을 추출하고 각 빈도를 계산한다. 계산된 빈도를 기반으로 음절 n-gram 빈도와 wildcard 음절 n-gram 빈도를 비교 후 임계치 이상의 빈도차를 보이는 어휘를 최종 미등록 어휘로 판별한다.

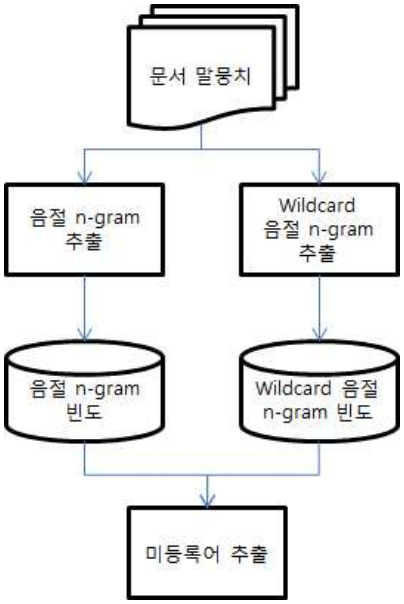


그림 1. 미등록 어휘 추정 시스템 구조도

3.1. 음절 n-gram 추출

문서 내 출현하는 어절을 기준으로 음절 n-gram을 추출하여 빈도수를 계산한다. 추출된 음절 n-gram의 예는 아래 표 1과 같다.

표 1. 음절 n-gram 및 빈도수

음절 n-gram	빈도수
아이	3098
아이폰	2017
아이폰은	28
아이폰을	29
아이폰에	20
아이폰에서	17
아이폰용	19
트윗	2072
트윗은	175
트윗을	158
트윗이	154

3.2. Wildcard 음절 n-gram 추출

Wildcard 음절 n-gram은 검색 엔진에서 사용되는 wildcard 질의처리 방식을 이용하여 추출한다. 예를 들어 “아이폰은”, “아이폰을”, “아이폰에”는 “아이폰*”로 표현이 가능하다. 이때 “아이폰*”의 빈도수를 측정한다. 추출된 wildcard 음절 n-gram의 예는 아래 표 2와 같다.

표 2. Wildcard 음절 n-gram 및 빈도수

n-gram	빈도수
아*	3098
아이*	2017
아이폰*	96
아이폰에*	17
트*	2072
트윗*	487

3.3. 미등록 어휘 추출

추출된 음절 n-gram 중 wildcard 음절 n-gram과 비교하여 빈도수의 차이가 급격하게 나타나는 n-gram을 최종 미등록 어휘로 추출한다. 이 때 wildcard 음절 n-gram에서 출현하지 않는 어휘는 추출하지 않는다. 예를 들어 표 1의 음절 n-gram “아이폰”의 빈도수는 2017이며, 표 2의 wildcard 음절 n-gram “아이폰*”의 빈도수는 96로 급격한 차이를 보이므로 미등록 어휘로 추출된다. 그러나 음절 n-gram “아이폰에”와 wildcard 음절 n-gram “아이폰에*”의 빈도는 각각 20, 17으로 빈도수 차이가 상대적으로 낮으므로 미등록 어휘로 추출되지 않는다. 표 1과 표 2를 통해 추출된 미등록 어휘는 아래 표 3과 같다. 표 3에서는 빈도수의 차이가 음절 n-gram 빈도수의 70% 이상이 되는 어휘만을 최종 후보로 추출하였다.

표 3. 추출된 미등록 어휘

결과	음절 n-gram 빈도	wildcard 음절 n-gram 빈도	빈도 차이
아이폰	2017	111	1906
트윗	2072	487	1585

4. 실험 및 결과

본 논문에서 제안한 미등록 어휘 추출 방법의 성능 평가를 위해 트위터 10만개, 미투데이 30만개의 글을 수집하였다. 표 4는 제안 시스템으로 추출된 어휘 중 빈도수 상위 20개를 나타낸다.

표 4. 제안 방법의 추출 결과 중 빈도수 상위 20 개의 어휘

순위	트위터 어휘	빈도	미투데이 어휘	빈도
1	;;	3001	ㅠㅠ	21692
2	아이폰	2150	ㅎㅎ	15104
3	트윗	1924	ㅜㅜ	5724
4	트위터	1642	미투데이	3886
5	한성주	1291	친신	3592
6	아..	988	ㅡㅡ	3386
7	ㄷㄷ	795	댓글	3128
8	화이팅	749	아이유	2786
9	대박	718	미친님들	2205
10	멘션	715	2011년	2046
11	:)	713	미친들	2002
12	ㅇㅇ	697	심심해	1974
13	아님	587	ㄱㄱ	1870
14	아~	581	아...	1844
15	선팔	567	격반	1571
16	음.	466	드림하이	1560
17	한국쓰리엠	425	인책	1555
18	우와	421	>_<	1536
19	팔로우	385	아~	1535
20	네이버	368	[인책]	1492

본 논문에서 제안한 방법으로 트위터에서 추출된 상위 20개의 결과 중 “트윗”, “트위터”, “멘션”, “선팔” 등과 같이 트위터에서 주로 사용되는 어휘가 추출된 것을 확인할 수 있다. 또한 미투데이에서 추출된 결과 중 “친신”(친구신청), “미친님들”, “미친들”(미투데이 친구들), “격반”(격하게 반긴다), “인책”(인원체크) 등의 줄임말이 출현하는 것을 확인할 수 있다. 상위 20개의 결과 외에도 트위터에서는 “카톡”, “아이패드”, “블랙베리”, “페이스북”, “안드로이드” 등의 어휘 들이 추출 되었으며, 미투데이에서는 “소녀시대”, “비스트”, “화이트데이”, “샤이니” 등의 어휘들이 추출되는 것을 확인할 수 있다.

그러나 “;;”, “ㄱㄱ”, “우와”,“암튼”, “있네요” 등과 같이 의미가 없는 어휘들도 상당 수 추출되는 것을 확인 할 수 있다.

5. 결론 및 향후 연구

본 논문에서는 음절 n-gram을 기반으로 한 미등록 어휘의 추정 방법에 대해서 제안하고 이를 평가하기 위하여 신조어 및 미등록어가 많이 출현하는 데이터를 사용하여 실험 하였다. 실험 결과 특정 사이트에서 빈번하게 사용되는 어휘들이 추출되는 것을 확인할 수 있었으며 연예인명, IT 용어 등과 같은 신조어들도 추출되는 것을 확인할 수 있었다. 그러나 의미가 없는 어휘들이 추출되는 것을 확인하였는데 이를 해결하기 위해 시간대 별로 카테고리를 분류하여 “ㄱㄱ”,“우와” 등과 같이 시간에 무의미한 어휘를 추출하여 적용할 수 있는 방법에 대해서 연구할 예정이다. 또한 주제 별로 카테고리를 분류하여 주제에 무의미한 일반적인 어휘를 추출하여 적용할 수 있는 방법에 대해서도 연구를 진행할 예정이다.

참고문헌

[1] 송수민, 신준수, 김학수, “음절 바이그램과 CRFs를 이용한 의학 전문 용어 추출”, 한국정보처리학회 춘계 학술발표대회, pp. 505-507, 2010.

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

[3] 고영중, “한국어 명사 출현 특성과 후절어를 이용한 명사 추출기” 정보과학회논문지: 소프트웨어 및 응용 제 37권 12호, pp. 919-927, 2010

한국어 부분언어에 대한 문법 정의 및 GLR 파싱[1)]

김지현, 정병채[0], 이재성
충북대학교 정보컴퓨터교육과, 비즈니스데이터융합학과, 디지털정보융합학과

A grammar definition and the GLR parsing for Korean sub-language

Ji Hyun Kim, Byung Chae Jung[0], Jae Sung Lee
Dept. of Information & Computer Education, Dept. of Business & Data Convergence, Dept. of Digital Informatics & Convergence, Chungbuk National University

요 약

최근 한국어를 배우는 외국인의 증가로 '외국어로서의 한국어 학습(KFL)'에 대한 관심이 늘고 있다. 본 논문에서는 외국인을 위한 한국어 교재에서 사용된 회화 문장으로부터 문장 패턴을 분석하고 이를 기반으로 한국어 부분 언어 문법을 정의한다. 대개 부분 언어 문법은 간단하고 배우기 쉬우므로 외국어로서의 한국어 학습자들이 쉽게 한국어로 의사소통을 할 수 있을 것이다. 특히, 본 논문에서는 이 부분 문법이 컴퓨터로 해석될 수 있도록 문법을 정의하였고, 이를 자동 어휘분석기 생성기(flex)와 자동 파서 생성기(bison)을 이용해 기본적인 검증을 하였다.

주제어: 한국어 부분 언어 문법, GLR 파싱, 외국어로서의 한국어 교육

1. 서론

최근 한류 열풍에 의해 외국 이민자의 유입이 증가하고 있다. 따라서 한국어를 배우고 싶어하고, 사용하고자 하는 외국인들이 증가하고 있다[2]. 배우고자 하는 외국어가 모국어와 많이 다른 경우 이를 배우는 것이 어렵고, 일정 수준 이상의 의사소통을 위해서는 많은 시간을 필요로 한다. 하지만 이해가 쉽고 배우기 간단한 부분언어를 개발한다면, 그리고 이것을 컴퓨터로 해석할 수 있도록 한다면 외국어를 쉽게 학습할 수 있을 것이다.

부분언어 문법은 제한된 영역에서 보다 명확하고, 쉽게 의사전달을 하기 위해 개발되었으며, 대표적으로 AECMA[3]가 있다. 이는 항공관제에서 사용하는 널리 알려진 통제언어이다. 이러한 통제 언어에서는, 사용되는 단어 수, 모호성을 가진 어휘의 사용, 수동태의 사용 등을 제한하고, 20단어 이내의 짧은 문장의 사용을 권한다. 이렇게 간략화된 영어를 국제표준화 하기 위한 움직임이 나타나고 있다[4].

CFG는 파싱에 사용되는 잘 알려진 문법 중 하나이며 기본적인 언어구조를 정의하는데 편리하다. 하지만 자연언어에서 CFG를 사용할 경우, 대개 모호성이 많아, 파스트리를 여러 가지로 생성해 낸다. 가능한 한 모호성을 줄이고 적은 수의 트리를 만들어 내기 위해 문맥 속성을 이용하여 제약을 가한 문법으로 PSG, HPSG, LFG 등이 있다. KPSG는 PSG을 한국어에 적용한 것이다[5]. 그럼에도 불구하고 부분언어처럼 언어의 구조가 간단할 경우, CFG는 기본 문형 정의에 사용될 수 있다.

영어는 어휘가 문장에서 사용된 위치에 따라 그 문법적 성분이 결정되는 언어로, CFG와 같이 구성요소의 순서에 따라 적용되는 규칙이 결정되는 문법이 적용되기 쉽다. 하지만 한국어는 문장요소의 위치가 자유롭고, 조사로 문장 성분을 나타낸다. 또한 주어나 목적어 등의 생략이 빈번하고, 모호성이 높아 이를 CFG로 표현하는 것은 매우 불편하고, 효율이 떨어진다. 이에 자유로운 어순을 반영할 수 있는 ECFG를 제안하기도 했다[6, 7]. 이러한 방법은 주로 문형을 문장 성분(주어, 동사, 목적어)을 중심으로 분류한 것이다. 반면에 서술어 중심 분류는 서술어에 따라 문장의 유형으로 분류하고 사용하는 방식이다[8]. 하지만 실제적으로 이것을 파싱에 체계적으로 사용하기는 어려운 점이 있다.

본 논문에서는 한국어를 이용한 의사소통에서 가장 필수적인 부분언어를 정의하고, 이에 대한 문형을 CFG로 정의하며 이를 GLR 파싱 방법으로 파싱할 수 있도록 한다. CFG의 구조를 정의하고, 속성을 추가하여 제약하는 기존 연구들과는 달리, 본 논문에서는 한국어 부분언어를 제안하여 어순을 제약하고, 이를 CFG 구조로 표현했다는 점에서 차별점을 갖는다. 한국어 부분언어는 외국인을 위한 한국어 교재에서 쓰인 문장을 이용하여 추출하였고, 문장의 구조는 품사 수준에서 정의하였다.

다음으로 2장 관련연구에서는 CFG를 이용하여 개발한 한국어 파서의 문형들을 살펴본다. 3장에서는 기본적인 의사소통이 가능한 수준의 한국어 부분언어를 정의하고, 이를 반영한 CFG를 제안한다. 4장에서는 제안된 CFG를 파싱할 수 있는 GLR 파싱 방법을 제시한다. 이어 5장은 실험 결과, 6장은 결론으로 마무리한다.

2. 관련 연구

CFG는 구성요소의 순서에 따라 문법적 성분이 결정되기 때문에 영어와 같이 위치에 따라 문법적 성분이 결정되는 언어에 강점이 있다. 하지만 한국어는 부분자유어순으로 문장요소들의 위치가 유동적이므로 이를 CFG로 모두 표현하려면 경우의 수가 매우 많아 비효율적이다.

이러한 문제점을 해결하기 위해 자유로운 어순을 반영

1) 이 논문은 [1]의 일부를 확장 보완한 것임.

하여 파싱하는 ECFG를 제안하기도 했다[6, 7]. 나동렬[6]에서는 문장(S)에서 반복적으로 사용되는 체언구(PP)와 부사구(ADVP)를 쉽게 표현하기 위해 그림 1과 같은 CFG를 제안했다. 또한 그림 2와 같이 반복 표시 *를 이용하여 간단하게 표현하였다.

```
S -> PP S
S -> ADVP S
S -> VP
```

그림 1. 간단한 한국어 CFG

```
S -> PP* VP
S -> (PP | ADVP)* VP
```

그림 2. 반복 표현을 이용한 한국어 CFG[6]

양성일[7]은 이를 확장한 ECFG를 제안하며 인용절, 대등절, 종속절 등을 처리할 수 있도록 했다. 문장의 종류를 파악할 수 있는 장치를 하고 이를 문법으로 표현하여 이들 관계를 제약하도록 했다. 예를 들어, 그림 3은 대등절 및 종속절을 처리하는 문법으로 $S_{[ce]}$는 대등절 연결어미로 끝나는 대등절, $S_{[dce]}$는 종속적 연결어미로 끝나는 종속절, ! 는 부정을 의미한다. 즉, 대등절 연결어미로 끝나는 문장이 반복된 후, 대등절 어미가 아닌 문장이 오면, 문장이 종료하는 첫 번째 문장과, 종속절 연결어미로 끝나는 문장이 온 후, 일반 문장이 와서 끝나는 두 번째 문장을 정의한 것이다.

$$S \rightarrow S_{[ce]}^{*}\ S_{![ce]} \mid S_{[dce]}\ S$$

그림 3. ECFG 예[7]

문장의 구조적 유형을 공식화한 틀을 문형이라 하며 주로 서술어에 따라 구분된다[8]. 이용석[11]은 이러한 문형을 CFG로 구현하여 처리하였다. 이러한 문형의 일부 예는 그림 4와 같다.

```
N(이/ 는/ 은/ 가) + V
N(이) + N(에/ 에게) + V
N(이) + N(로/ 으로) + V
N1(은) + N2(이) + V   // 근심되다, 생각나다, 욕심나다
N(이) + A             // 아름답다, 붉다 ...
N(이) + N(에) + A     // 유리하다, 능하다, 해롭다 ...
N(이) + N(와) + A     // 같다, 다르다, 동일하다...
```

그림 4. 문형 예[11]

3. 한국어 부분언어 정의

한국어는 회화체에서의 생략이 매우 빈번하다. 문장 내에서 주어나 목적어 등을 생략하거나 어절내에서 격조사를 생략하는 등 여러 가지 경우가 있다. 이러한 생략에도 의사소통이 가능한 것은 구문분석으로는 모호성이 많으나 의미나 화용 분석에서는 모호성을 해결할 수 있기 때문이다. 따라서 구문분석에서는 생략에 의한 모호성에도 불구하고, 구문분석의 결과를 다음 단계인 의미분석이나 화용분석으로 보내 줄 수 있어야한다. 또, 외국인들에게는 격조사의 생략이 훨씬 편리한 언어 용법일 수 있다. 가능하면 구성적 요소를 반영하여 이에 익숙한 외국어 학습자들에게 쉽게 적응할 수 있도록 하는 것도 한 방법일 수 있다.

이 논문에서는 이러한 특성을 반영하여 외국인이 이해하기 쉽고, 또한 CFG로 표현이 쉽게 되는 부분 언어 문법을 개발한다. 이를 이루기 위한 기준은 다음과 같다.

1) 어순은 주어 목적어 술어의 순으로 제약한다.
2) 부사와 같이 임의의 위치에서 수식하는 어절은 가능하면 위치를 정해 주어 혼동이 없도록 한다.
3) 외국인들이 조사 사용에 어려움을 겪는 것을 고려하여 조사가 생략된 한국어 문형을 개발하되, 대표적 외국어 어순에 맞도록 한다.
4) 구문구조가 불명확한 경우에 한 해, 조사 사용을 필수화하여 구문구조가 드러날 수 있도록 한다.
5) CFG 표현에서 한국어 조사나 어미 형태를 고려하여 문법 표현을 한다.
6) 문법 내에 어휘나 태그를 사용하여 문법범주를 세분화하여 표시한다. 어휘는 직접 문법에 사용할 수도 있지만, 가능하면 태그로 대신한다.

조사가 생략된 문형을 개발할 경우, 조사를 생략한 채 어순을 변화시키면 잘못된 구조의 문장이 나올 수 있다. 즉, 한국어는 어순이 자유로와 SOV, OSV 등의 구조가 가능하여 조사를 생략해도 이해가 가능하지만, 외국인들에게는 (1)의 원문이 아래의 (3)처럼 밥이 나를 먹었다는 뜻으로 전혀 다르게 해석될 수 있다.

- 나는 밥을 먹었다. I ate steamed rice (1)
- 나 밥 먹었다. I ate steamed rice (2)
- 밥 나 먹었다. Steamed rice ate me (3)

조사를 사용할 경우, 격조사 정보가 상위 문법 정보에 포함되어 있어야 한다. 즉, 기존의 명사에 주격조사가 붙어 '주어' 태그가 되거나, 동사 어간에 연결어미가 붙어 '연결서술어'와 같은 형태로 아래와 같이 표현하여 문법 범주를 세분화할 필요가 있다.

<주어> -> <명사> + 주격조사
<연결서술어> -> <동사어간> + 연결어미

한국어 부분언어 문형을 개발하기 위해 국립국어원에

서 발간한 ‘여성결혼이민자와 함께하는 한국어1’ 교재[12]에 수록된 총 1,841문장을 사용하였다. 이 중 존칭형 ‘습니다.’와 같은 문장은 기본형 ‘ㄴ다’, 와 해요체 ‘요’로 바꾸었다. 그 결과 총 989문장이 나왔으며 이를 간소한 문형 개발을 위한 원본 문장으로 사용하였다. 이 문장들을 형태소 분석기 및 태거 프로그램[13]을 사용하여 품사 태그열을 추출하고 이 중 반복되는 문장 패턴(문장의 품사 태그열)을 제거한 186개의 문장 패턴만을 기초자료로 사용하였다. 품사 태그열을 이용하여 문법규칙(grammatical rule)을 정의했지만, 가능하면 태그열이 나온 어절 단위(띄어쓰기 단위)를 고려하여 기술하였다. 그리고 문장에 어순이 주어, 목적어, 부사어에 기술되는 위치가 다르더라도 각각의 체언 뒤에 붙은 조사를 구분하여 주어, 목적어, 부사어에 해당되는 어절 단위로 인식하였다.

4. GLR 파싱 및 실험

CFG의 파싱은 일반적으로 LR 파싱 방법으로 수행하고 대부분 결정적인 프로그래밍 언어의 특성에 따라 LR 파싱 방법은 효과적이다. 문법이 모호하여 두 개 이상의 파스트리가 발생하는 경우, 우선 순위를 두고 우선 순위에 따라 하나를 선택하였다. 하지만 자연언어는 많은 모호성을 가지고 있고, 모호성의 해결은 어휘, 구문, 의미 등의 분석과정을 거치며 해소된다. 본 논문에서는 문법 정의만을 목적으로 하여, 어휘 모호성이 해결된 형태소 태그열을 입력 받아 모호성이 포함된 구문 분석 결과를 내놓는다. 여기서 생성된 모호성이 포함된 구문분석 결과는 상위 단계에서 해결되어야 할 것이다.

모호성이 포함된 파스트리는 LR 파싱으로는 한계가 있기 때문에 파싱 과정을 일반화시킨 GLR 파싱 방법을 사용한다. LR 파싱은 푸시다운 오토마타를 수행하면서 모호성이 발생하면 그중 선호하는 하나의 오토마타 경로만을 선택하여 수행하고, 만약 그 경로가 잘못되면 오류로 종료한다. 반면에 GLR 파싱은 모호성이 발생하면 가능한 모든 오토마타 경로를 수행하다 최종적으로 병합되는 경로를 찾아 그 결과를 출력한다. 따라서 GLR파서는 모호성이 포함된 문장의 구문분석이 어느 정도 가능하다.

본 논문에서는 bison을 이용하여 GLR 파싱이 가능한 부분 문법을 정의하였다[10]. 그림 5는 GLR 파싱이 실행되는 방법을 설명하기 위한 것이며, 격조사를 생략했을 경우, 명사가 주어와 목적어로 파싱될 수 있도록 만든 문법의 예이다.(세종 말뭉치[14]의 품사 태그를 사용함)

%merge 문은 공통의 경로가 나왔을 경우, 이를 하나의 경로로 병합하는 구문분석 명령이다. bison의 RHS 문법 표기에서 모호성이 있는 비단말 문법기호가 제대로 병합되기 위해서는 그 문법기호 다음에 반드시 모호성이 없는 문법기호가 나와야 병합이 이루어진다. 이를 처리하기 위해, 문법 기술을 다양하게 변형하여 사용하였다. 또, cn은 create_node 모듈의 약칭으로 매개변수를 이름으로 갖는 노드를 만들어 그 포인터를 리턴한다. ac는 add_child 모듈의 약칭으로 첫 번째 매개변수에 두 번째 매개변수의 포인터를 자식 노드로 붙이되 세 번째 매개변수의 위치에 자식노드로 붙인다. 예를 들어, ac($$, $1, 1)은 $1의 노드를 $$의 1번째 자식 노드로 붙인다.

```
PGM   : s END        { Tree_print($1);  printf("OK\n"); return 1;}
      ;
s     : subj vp %merge <stmtMerge> {$$=cn("s"); ac($$, $1, 1); ac($$, $2, 2); }
      | obj vp %merge <stmtMerge> {$$=cn("s"); ac($$, $1, 1); ac($$, $2, 2); }
      ;
subj  : noun JKS     { $$=cn("subj"); ac($$, $1, 1); ac($$, $2, 2); }
      | noun         { $$=cn("subj"); ac($$, $1, 1); }
      ;
obj   : noun JKO     { $$=cn("obj"); ac($$, $1, 1); ac($$, $2, 2); }
      | noun         { $$=cn("obj"); ac($$, $1, 1); }
      ;
noun  : NNG          { $$=cn("noun"); ac($$, $1, 1); }
      | NNP          { $$=cn("noun"); ac($$, $1, 1); }
      ;
vp    : VV EF              { $$=cn("vp"); ac($$, $1, 1); ac($$, $2, 2); }
      | VA EF              { $$=cn("vp"); ac($$, $1, 1); ac($$, $2, 2); }

      ;
```

그림 5. GLR 파싱을 위한 간단한 예

이 문법은 bison에 의해 c파일로 바뀌고, 이를 컴파일하여 부분문법 파서를 만들 수 있다. 이 과정에서 품사 태그를 인식하여 토큰으로 넘겨주는 어휘분석기가 필요한데, 이는 flex를 이용하여 구현하였다[9]. 아래는 이 문법에 따라 실험한 예이다. 즉, 태그열을 입력하여 그에 상응하는 파서 트리를 출력한 것이다. 파스트리의 표현은 일반트리를 이진트리로 변환하여 표현하였다. 즉, 첫 번째 자식노드만 부모의 자식노드로 표현하고, 두 번째부터의 자식노드들은 첫 번째 자식노드의 형제노드(sibling)로 표현하였다. 그림 6은 문장 아래 subj과 vp가 있고, subj 밑에 noun과 JKS가 자식 노드로 있는 그림이다. 이 경우, 모호성이 없으므로 파스트리가 하나이다.

```
D>subglr
NNP JKS VV EF END

s
|
subj--------vp
|           |
noun--JKS   VV----EF
|
NNP
```

그림 6. 모호성이 없는 GLR 파싱 예

그림 7은 격조사의 생략으로 모호성이 있는 경우로 ‘?’ 노드 밑에 가능성이 있는 파스트리가 표현된다. 그림 7의 경우는 obj vp를 갖는 s와 subj vp를 갖는 s의 두 가지가 모두 그려진 결과이다.

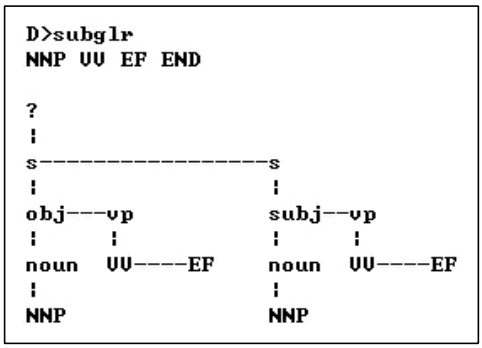


그림 7. 모호성이 있는 GLR 파싱 예

실험은 외국어로서의 한국어 교재[12]에서 일부를 발췌하여 수행하였다. 형태소 품사 태그는 어휘 중의성과 형태소 중의성이 해소된 상태로 입력된 것으로 가정하여, 파서의 성능만을 독립적으로 평가하였다. 따라서, '나는' 이라는 입력이 들어오면 이를 태깅한 결과 '나/NNP + 는/JKC' 가 나온다고 가정하고, 이들 중 NNP+JKC 태그열을 입력으로 파서에 넘긴다. 총 1,841문장에서 동사에서 존칭 생략 등을 통해 문장을 간략화하여 989문장을 추출하였고, 이 문장의 패턴을 정리하여 186개의 문장 패턴(태그열 패턴)을 얻었다. 이를 기반으로 부분 CFG 문법을 개발하고, 이를 이용하여 간략화된 989문장을 테스트해 본 결과 326개의 문장(33%)이 파싱에 성공하였다.

5. 결론

부분문법 언어는 주로 제한된 영역에서 명확한 의사전달을 목적으로 한다. 본 논문에서는 외국인의 쉬운 한국어 학습을 위하여 한국어 구문을 단순화시킨 부분문법을 개발하였다. 외국인을 위한 한국어 교육 교재에 나온 문장들을 이용하여 개발하였고, 기본 문형을 추출한 후 GLR 파싱이 가능한 형태로 변형하였다.

본 논문에서는 총 1,841문장을 동사에서의 존칭 생략 등을 이용해 간략화하고 문장 패턴을 정리하여 186개의 문장 패턴(태그열 패턴)을 얻었다. 이를 기반으로 부분 CFG 문법을 개발하고, GLR 파싱의 가능성을 검증하였다. 이 연구는 간소 패턴을 사용한 의사소통이 궁극적인 목적이다. 따라서, 다시 원본 문장과 같은 의미의 전달이 가능하면서 GLR 파싱이 가능한 문장으로 패턴을 변경할 경우, GLR 파싱 성공률은 증가하였고, 이를 계속할 경우, 어느 정도 안정된 형태의 부분 문법을 완성할 수 있을 것이다. 본 논문에서는 기본 문형 패턴의 개발과 GLR 파싱의 가능성을 확인하였으며, 이를 이용하여 문형개발을 계속할 필요가 있다.

참고문헌

[1] 김지현, "외국어로서의 한국어 교재를 위한 한국어 문형," 충북대학교 교육대학원, 교육학석사학위논문, 2013.

[2] 김승권 외, "2009년 전국 다문화 가족 실태 조사 연구," 보건복지부, 법무부, 여성부, 한국보건사회연구원, 2010.

[3] AECMA Simplified English : http://www.techscribe.co.uk/ta/aecma-simplified-english.pdf.

[4] Richard Hodgkinson, A standard for simplified natural language : http://www.tcanz.org.nz/site/tcanz/files/Standards/InternationalStandardsSimplifiedEnglish.pdf.

[5] 김영택, 자연언어처리, 생능출판사, 2001.

[6] 나동렬, "한국어 파싱에 대한 고찰," 정보과학회지 제 12권, 제 8호, pp.33-46, 1994.

[7] 양성일, "확장 문맥 자유 문법과 패턴-액션 규칙을 이용한 한국어 구문 분석에 관한 연구," 연세대학교 대학원 전산과학과, 석사학위 논문, 1999.

[8] 강은국, 조선어 문형 연구, 박이정 출판사, 1996.

[9] flex 2.2.5 : ftp://reality.sgiweb.org/freeware/fw-5.3/fw_GNUflex/GNUflex.html.

[10] GNU bison 2.7 : http://www.gnu.org/software/bison/manual/bison.html

[11] 이용석, "구문분석의 실용화를 위한 구문분석기 개발에 관한 연구," 한국전자통신연구원, 1999.

[12] 국립국어원, 여성결혼이민자와 함께하는 한국어 1 : http://www.korean.go.kr/09_new/edu/kteacher_book01.jsp.2010

[13] 김보겸, 이다니엘, 이재성, "3단계 확률기반 형태소 분석기를 이용한 한국어 품사 태거 구축 방법," 한국컴퓨터교육학회 학술발표대회논문집, 제 16권, 제 1호, pp. 129-134, 2012.

[14] 국립국어원, 21세기 세종계획 최종결과물(2011년 12월 수정판), 2011.

등급 재현율: 이중언어 사전 구축에 대한 평가 방법

서형원○, 권홍석, 김재훈
한국해양대학교 IT공학부
wonn24@gmail.com, hong8c@naver.com, jhoon@kmou.ac.kr

Rated Recall: Evaluation Method for Constructing Bilingual Lexicons

Hyeong-Won Seo○, Hong-Seok Kwon, Jae-Hoon Kim
Korea Maritime University, Computer Engineering Institute

요 약

이중언어 사전 구축 방법을 평가하는 방법에는 정확률, 재현율, MRR(Mean Reciprocal Rank) 등이 있다. 이들 방법들은 평가 집합에 있는 대역어를 정확하게 찾는 것에 초점을 맞추고 있다. 그러나 어떤 대역어가 얼마나 많이 사용되는지는 전혀 고려하지 않는다. 즉 자주 사용되는 대역어를 빨리 찾을 수 있는 방법이 좋은 방법이라고 말할 수 있다. 이와 같은 문제를 해결하기 위해서 본 논문에서는 이중언어 사전 구축의 새로운 평가 방법인 등급 재현율을 제안한다. 등급 재현율(rated recall)은 대역어가 학습 말뭉치에 나타난 정도를 반영하는 재현율이며, 자주 사용되는 대역어를 얼마나 정확하게 찾는지를 파악할 수 있는 좋은 측도이다. 본 논문에서는 문맥벡터와 중간언어를 이용한 이중언어 사전 구축 시스템의 성능을 평가하고 기존의 방법과 비교 분석하였다.

주제어: 이중언어 사전, 문맥벡터, 중간언어, 등급재현율

1. 서론

이중언어 사전(bilingual lexicon)은 자연언어처리(Natural Language Processing), 기계 번역[1], 다중언어(Multilingual) 정보검색[2] 등의 분야에서 주요한 자원으로 활용되고 있다[3]. 영어와 불어과 같이 널리 사용되는 언어에 대해서는 이중언어 사전의 구축이 그다지 어렵지 않다. 그러나 모든 언어 쌍마다 이중언어 사전을 구축하는 것은 많은 시간과 노력이 소요된다[4-7]. 이런 연구들은 주로 병렬 말뭉치(parallel corpora)나 비교 말뭉치(comparable corpora)를 이용하고 있다. 병렬 말뭉치는 통계기반 기계번역에서 널리 사용되는 단어 정렬(word alignment)을 이용하고 병렬 말뭉치는 문맥벡터 기반 방법을 이용한다. 이 방법들은 초기 사전(seed dictionary)이나 말뭉치의 존재 유무가 매우 중요하다. 그러나 어떤 언어 쌍에 대해서는 초기 사전뿐 아니라 병렬 및 비교 말뭉치조차도 쉽게 구할 수 없다. 예를 들면 한국어(KR)와 스페인어(ES)/불어(FR)로 공개된 이중언어 사전도 없고, 더 나아가 공개된 병렬 혹은 비교 말뭉치도 없다. 이런 환경에 맞서 적은 노력과 시간을 투자하면서도 이중언어 사전을 구축할 수 있는 간단하면서도 효과적인 방법이 제안되었다[8]. 이 방법은 사전을 구축할 때 생길 수 있는 도메인 문제를 완벽하게 해결할 수는 없지만 병렬 말뭉치와 중간언어(pivot language)를 사용함으로써 초기 사전이 필요하지 않다는 점과 구축이 어려운 언어 쌍에서도 충분히 이중언어 사전을 구축하기에 용이하다는 장점이 있다.

한편, 이중언어 사전 구축 방법을 평가하는 방법에는 정확률(accuracy), 재현율(recall), MRR(Mean Reciprocal Rank) 등이 있다. 이들 방법들은 평가 집합에 있는 대역어를 정확하게 찾는 것에 초점을 맞추고 있다. 그러나 어떤 대역어는 얼마나 자주 사용되는지는 고려하지 않는다. 자주 사용되는 대역어를 빨리 찾을 수 있는 방법이 좋은 방법이다. 예를 들면 한국어 '학교'는 영어 'school', 'college', 'institution'로 번역될 수 있다. 그러나 많은 경우에는 'school'로 번역된다. 이처럼 이중언어 사전 구축 시스템에서도 번역되는 대역어를 빨리 찾을 수 있는 시스템이 좋은 시스템이다. 이와 같은 문제를 해결하기 위해서 본 논문에서는 이중언어 사전 구축의 새로운 평가 방법인 등급 재현율을 제안한다. 등급 재현율(rated recall)은 대역어가 학습 말뭉치에 나타난 정도를 반영하는 재현율이며, 자주 사용되는 대역어를 얼마나 정확하게 찾는지를 파악할 수 있는 좋은 측도이다.

본 논문의 구성은 다음과 같다. 2장에서 문맥벡터에 기반을 둔 이전 연구에 대하여 간략히 기술하고, 3장에서는 새롭게 제안하는 평가 방법인 등급재현율에 대하여 기술한다. 4장에서는 실험에 대한 내용을 기술하고 마지막으로 5장에서 결론을 짓는다.

2. 관련 연구

2.1 이중언어 사전 자동구축

이중언어 사전을 효과적으로 구축하기 위해 다양한 기존 연구들이 진행되어 왔다[9]. 이런 연구들에는 중간언어를 이용하는 방법[4][5][9], 병렬 말뭉치를 이용한 방법[10][11], 그리고 비교 말뭉치를 이용한 방법[6][7][12] 등이 있다. 병렬 말뭉치를 이용하여 이중언어 사전을 만들었을 때 충분히 좋은 성능을 보였다는 연구 결과가 있다[13]. 하지만 영어를 제외한 언어에 대해서는 병렬 말뭉치가 공개된 것이 드물고, 만약 구축해서 효과적으로 쓰기 위해서는 상당히 많은 양의 말뭉치가

필요하다는 한계점이 있다.

이런 문제들을 해결하기 위해서 중간언어를 활용하는 연구들이 있었다[3]. 중간언어를 이용하면 언어 자원이 많지 않은 언어 사이에서도 보다 쉽게 병렬 말뭉치를 얻을 수 있다는 장점이 있다. 이에 반해, 비교 말뭉치는 특정 도메인에 대하여 언어가 서로 다르지만 문맥이 비슷한 문서들(일반적으로 집필 날짜가 겹치는 뉴스 문서)을 모아서 구축하였기 때문에 영어처럼 언어 자원이 풍부한 언어가 아닌 쌍에 대해서도 병렬 말뭉치보다 비교적 쉽게 구축할 수 있다는 장점이 있다. 또한 병렬 말뭉치보다 도메인 문제를 어느 정도 해결할 수 있다는 장점도 있다. 하지만 기존에 비교 말뭉치를 사용하여 이중언어 사전을 구축한 연구[14]를 보면 이 방법은 초기 사전이 필요하다는 것을 알 수 있다. 초기 사전은 원시 언어(SL: Source Language)나 대상 언어(TL: Target Language) 문맥벡터를 다른 언어로 번역할 때 사용하며, 그 단어의 양이 많을수록 좋지만 처음에 적은 양이여도 무방하다는 특징을 가지고 있다.

2.2 기본 문맥벡터 방법

본 절에서는 대표적인 이중언어 사전 구축 방법인 '문맥벡터 기반 방법'[14]을 살펴본다. Fung은 이중언어 사전을 구축하기 위해서 문맥벡터를 만들었고 이것의 대략적인 과정을 기술하면 다음과 같다. 먼저 원시 언어와 대상 언어의 비교 말뭉치에서 모든 단어를 대상으로 문맥벡터를 만든다. 이 때, 두 단어의 연관도를 측정하기 위해서 Chi-square Test[18]와 같은 연관성 측도(association measure)를 이용한다. 그 다음, 초기 사전을 이용하여 한 쪽 언어의 벡터를 다른 쪽 언어에 맞게 번역한다. 그러면 벡터 공간의 차원이 같아지기 때문에 원시 언어와 대상 언어의 벡터들을 서로 비교할 수 있게 된다. 원시 언어의 한 단어에 대한 벡터와 대상 언어의 모든 단어에 대한 벡터들을 그들 간의 유사도를 통해 서로 비교한 후, 그 유사도에 따라 정렬하고 상위 몇 개의 후보를 추출한다.

이 방법을 활용한 몇 가지 다른 변형된 방법들을 기술하면 다음과 같다.

<문맥 범위 조절>
- 3문장[15]
- 3단어 25개[16]

<초기 사전의 양 조절>
- 16,000개의 단어[17]
- 대략 2만개[12][14,15]

<벡터들 간의 유사도 계산 방법 조절>
- city-block measure[17]
- cosine [12][14-16]
- dice 혹은 Jaccard indexes[12][15]

2.3 중간언어 기반 문맥벡터 방법

2.2 절에서 기술한 방법에 대하여 정리하면 다음과 같다. 이전의 문맥벡터 기반 연구는 비교 말뭉치와 초기 사전을 사용한다. 초기 사전은 전체 성능에 영향을 주기 때문에 매우 중요한 요소이며, 이 사전이 문서에 포함된 단어들을 얼마나 포함하고 있는지도 중요한 문제가 될 수 있다. 또한 언어가 바뀔 때마다 사전도 구축해야 한다는 문제점이 있다.

이에 반해, 중간언어 기반의 문맥벡터 방법은 우리가 대상으로 하는 언어에 대해 비교 말뭉치가 아닌 병렬 말뭉치를 사용한다. 이 방법을 이용하면 원시 언어 문맥벡터를 대상 언어에 맞게 번역할 필요가 없기 때문에 초기 사전을 일일이 구축하지 않아도 된다는 장점이 있다.

이런 중간언어를 이용한 문맥벡터 방법을 좀 더 자세하게 묘사하면 그림 1과 같이 나타낼 수 있다. (1) 먼저

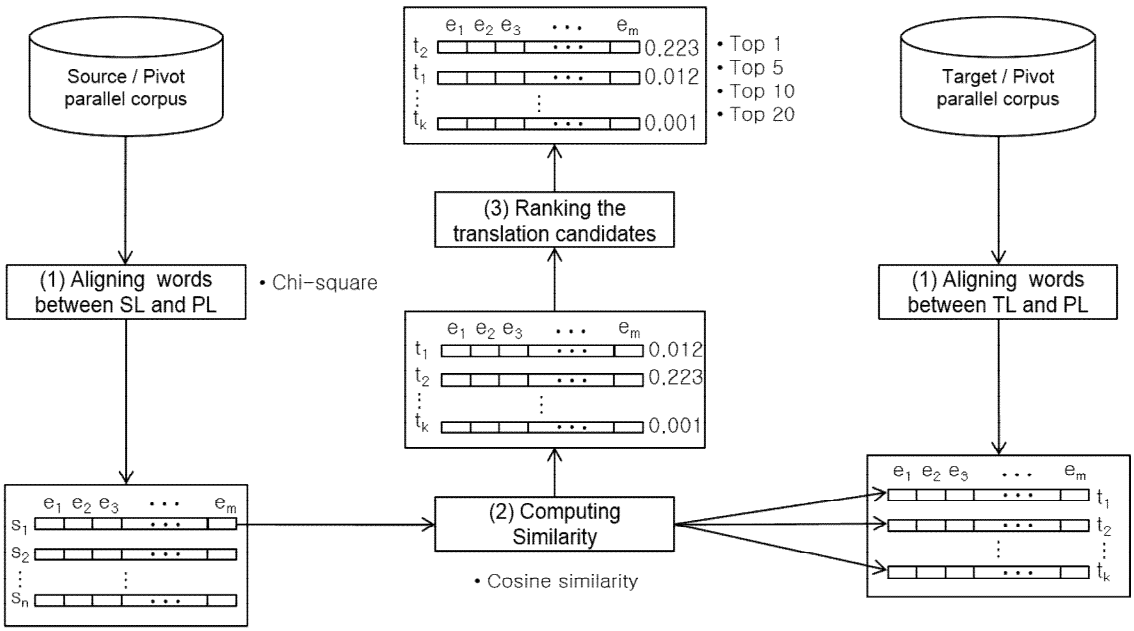


그림 1. 중간언어를 활용한 문맥벡터 방법 흐름도

각각의 병렬 말뭉치들에 포함된 모든 단어들에 대하여 연관성을 측정한다. 여기서 병렬 말뭉치는 SL와 중간언어(PL: pivot language) 그리고 TL와 PL로 구성된 2개의 말뭉치를 의미한다. 본 논문에서는 기호나 불용어를 제외한 나머지 단어를 대상으로 하며 명사, 동사, 형용사, 부사를 제외한 품사의 단어는 제외시킨다. 그 후, 남아있는 단어들 사이에 Chi-square test를 이용하여 두 단어(SL/TL과 PL의 단어)가 서로 얼마나 연관성이 있는지를 측정한다. 여기서 사용된 단어 빈도수는 DF(Document frequency)를 사용하고, 문장을 문서로 간주하여 단어를 포함하고 있는 문장의 수를 센다. (2) 이렇게 만들어진 문맥벡터들(SL-PL, PL-TL) 사이에 Cosine measure를 이용하여 벡터들 간의 거리 유사도를 계산한다. (3) 그 후 유사도가 높은 순으로 정렬하여 각 SL 단어마다 상위 k개의 TL 번역 후보들을 추출한다.

이 방법을 이용하면 초기 사전과 같은 외부 자원 없이 병렬 말뭉치만으로도 SL과 TL 사이에 번역 후보들을 쉽게 추출할 수 있다.

3. 등급 재현율

일반적으로 이중언어 사전 구축 방법을 평가하는 방법에는 정확률, 재현율, MRR 등이 사용된다. 이들 방법들은 평가 집합에 있는 대역어를 정확하게 찾는 것에 초점을 맞추고 있다. 그러나 어떤 대역어는 얼마나 자주 사용되는지는 고려하지 않는다. 하지만 자주 사용되는 대역어를 빨리 찾는 방법이 좋은 방법이다. 예를 들면 한국어 '학교'는 영어 'school', 'college', 'institution'로 번역될 수 있다. 그러나 많은 경우에는 'school'로 번역된다. 이처럼 이중언어 사전 구축 시스템에서도 번역되는 대역어를 빨리 찾는 시스템이 좋은 시스템이다. 이와 같은 문제를 해결하기 위해서, 본 논문에서는 등급 재현율을 제안한다. 등급 재현율은 대역어가 학습 말뭉치에 나타난 정도를 반영하는 재현율이며, 자주 사용되는 대역어를 얼마나 정확하게 찾는지를 파악할 수 있는 좋은 측도이다.

먼저 재현율에 대해서 살펴보자. 재현율은 정답 단어를 시스템이 얼마나 찾았는가를 나타내며, 상위 n번째 후보 대역어의 재현율 R_n은 식 (1)과 같이 정의된다.

$$R_n = \frac{1}{N}\sum_{i=1}^{N}\frac{1}{|C_i|}\sum_{j=1}^{n} c_{ij}, \quad c_{ij} = \begin{cases} 1, \text{ if } t_{ij} \in C_i \\ 0, \text{ otherwise} \end{cases} \qquad (1)$$

여기서, N은 원시 단어의 총 수이며, C_i는 i번째 원시 단어의 정답 단어(대역어 집합)이고, $|C_i|$는 C_i의 개수이다. c_{ij}는 크로네커 델타 함수(Kronecker delta function)로서 i번째 원시 단어 s_i에 대한 j번째 시스템 결과 t_{ij}가 정답에 포함되면 1이 되고, 그렇지 않으면 0이 된다. 즉, 정답 단어를 시스템이 얼마나 정확하게 찾는가를 나타낸다.

등급재현율은 RR_n은 식(1)를 약간 수정하여 식 (2)와 같이 정의한다.

$$RR_n = \frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{n} c_{ij}\, r(t_{ij}), \quad c_{ij} = \begin{cases} 1, \text{ if } t_{ij} \in C_i \\ 0, \text{ otherwise} \end{cases} \qquad (2)$$

여기서, $r(t_{ij})$는 i번째 원시단어 s_i에 대한 후보 대역어 t_{ij}가 학습 말뭉치에 출현한 비율을 의미하며, $\sum_{t \in C_i} r(t)$은 1이다. 예를 들어, 스페인어 단어 decisión에 대한 정답 사전이 표 1과 같고, 시스템이 낸 결과가 표 2와 같다면 각 순위에 따른 등급 재현율의 값은 표 3과 같다.

정답 단어	빈도수	$r(t)$
결정	6,007	0.752
결심	173	0.022
결의	369	0.046
결단	130	0.016
결단력	10	0.001
재정	880	0.110
판정	414	0.052
합계	**7,983**	**1.000**

표 1. 스페인어 decisión의 정답 단어

순위	번역 후보	$r(t)$
1	결정	0.752
2	통합력	
3	결단	0.016
4	여부	
5	최종판단	
6	의사결정	
7	판정	0.052
8	판단	
9	결정권한	
10	관망자세	
11	확정	
12	독자위성	
13	유격대	
14	앙케트	
15	결심	0.022
16	사업확장계획	
17	개인신용대출	
18	판결	
19	재정능력	
20	사항	

표 2. decisión에 대한 시스템 결과

순위	등급 재현율(RR_n)	재현율(R_n)
1	0.752	1/7 = 0.143
3	0.752 + 0.016 = 0.768	2/7 = 0.286
10	0.768 + 0.052 = 0.820	3/7 = 0.429
20	0.820 + 0.022 = 0.842	4/7 = 0.571

표 3. 각 랭크에서의 등급재현율과 재현율

표 1에서 알 수 있듯이 실제 문서에서 얼마나 자주 출현되느냐에 따라 각 단어의 비율이 결정되고, 이 비율이 실제 등급 재현율을 계산할 때 더해지게 된다. 표 2는 원시 언어인 스페인어 decisión에 대한 시스템 도출 결과를 나열한 것이다. 그리고 정답 사전으로부터 각 단어의 비율을 나타낸다. 비율이 표시되어 있지 않은 단어는 정답 사전에 없는 것으로써 대부분 합성명사인 것을 알 수 있다. 표 3은 실제 등급재현율을 계산하는 과정을 표로 나타낸 것이고 세 번째 줄에서 빈 칸은 등급 재현율이 0.0이다. 각 순위에서의 등급재현율은 상위부터 해당 순위까지 포함된 단어 중 정답 사전에 포함됨과 동시에 실제로 문서 안에 얼마나 많이 출현하였는지를 고려한 재현율이 계산되게 된다. 여기서 의미하는 비율은 하나의 원시 언어를 기준으로 하는 군집으로써의 비율이므로 일반적인 재현율 계산 때의 N_i로 나누는 작업은 생략하게 된다.

4. 실험 및 결과

4.1 실험 환경

4.1.1 데이터

본 논문에서는 실험을 위해 한국어(KR)-영어(EN), 스페인어(ES)-EN, 불어(FR)-EN의 병렬 말뭉치를 사용하였다. KR-EN은 433,151개의 문장으로 구성되었고 Seo et al.[19]의 연구에서 사용된 말뭉치를 보완하여 만든 뉴스 기사다. 이 말뭉치의 문장 당 평균 단어(한글의 경우에는 형태소)의 수는 각각 42.46(KR), 36.02(EN)이다. 반면에 ES-EN과 FR-EN 병렬 말뭉치는 Eurorarl(European parliament proceedings) 병렬 말뭉치이며 각각 약 160만, 200만 문장을 포함하고 있다. 이들의 문장 당 평균 단어 수는 각각 29.40(ES-EN에서 추출된 ES), 28.65(ES-EN에서 추출된 EN), 31.17(FR-EN에서 추출된 FR), 28.68(FR-EN에서 추출된 EN)이다.

평가를 위한 정답 사전은 다음과 같이 구축하였다. 사전을 구성하는 단어를 정하기 위해 각 병렬 말뭉치로부터 빈도수가 높은 순서대로 정렬한 후 가장 빈도수가 높은 100개의 명사(High)와 빈도수가 낮은 100개의 명사(Low)를 웹 사전을 참조하여 사람이 직접 구축하였다. 최종적으로 구성된 정답 단어 당 평균 번역 단어의 수는 각각 11.41(FR-KR), 10.3(ES-KR), 5.79(KR-FR), 7.36(KR-ES)개이다. 불어와 스페인어의 한글 번역 단어의 개수가 그 반대인 경우보다 상대적으로 많은 것을 확인할 수 있다.

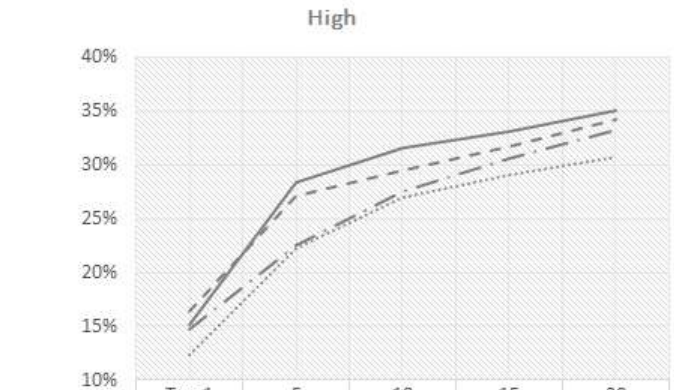
4.1.2 전처리

KR의 경우 한나눔 태거(KAIST Tagger)[1][20]를 이용하여 품사를 부착하는 전처리만 수행하였다. EN, ES, FR의 경우에는 Tree Tagger[2][21]를 이용하여 토큰 분리, 원형 분리(lemmatization)를 한 후 품사를 부착하였다. 품사가 모두 부착된 상태에서 각각의 언어에 맞는 불용어(KR은 제외)와 특정 품사를 제외하였다. KR 말뭉치에서는 총 69개의 품사 중 보통명사, 고유명사, 용언 그리고 수식언을 제외한 나머지 51개의 품사에 해당하는 단어들은 모두 배제하였다. 나머지 언어에 대해서도 같은 작업을 하여 EN은 61개 품사 중 19개, ES는 72개 중 33개, 마지막으로 FR은 36개 중 18개를 배제하였다.

전처리 후에 남은 단어의 타입 수는 각각 KR-EN 67,210(KR의 경우에는 형태소)/41,719(EN), ES-EN 12,926(ES)/28,764(EN), FR-EN 47,220(FR)/51,245(EN)이다. 이 중에서 같은 성격의 말뭉치임에도 불구하고 두 영어 문서(ES-EN과 FR-EN)의 타입 수(각각 28,764와 51,245)가 차이가 나는 이유는 실제 FR-EN의 영어 문서에 다수의 불어 문장이 포함되어 있기 때문이다.

4.2 실험 결과

본 논문에서 제안한 방법으로 실험한 결과는 다음과 같다. 그림 2와 3은 연관성 측도와 유사도 측정 방법을 각각 Chi-square test와 Cosine measure로 정해놓고, 각 병렬 말뭉치로부터 만든 문맥벡터 간에 유사도 계산 결과를 등급 재현율로 나타낸 것이다. 그림 2는 빈도수가 높은 단어(High), 그림 3은 빈도수가 낮은 단어(Low)에 대한 결과이다.

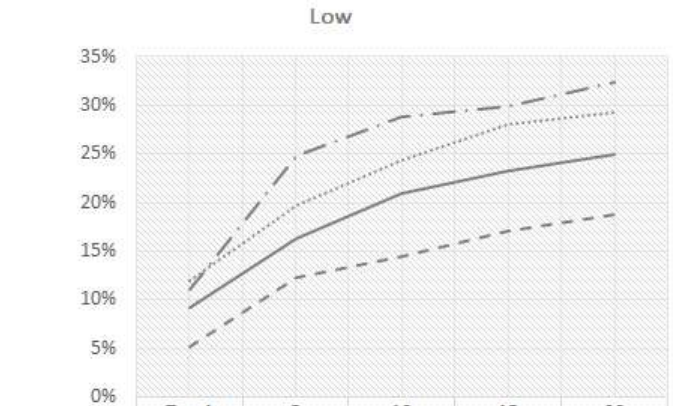


	Top 1	5	10	15	20
KR→FR	0.1228	0.2222	0.2700	0.2908	0.3074
FR→KR	0.1638	0.2710	0.2948	0.3171	0.3415
KR→ES	0.1473	0.2255	0.2754	0.3053	0.3318
ES→KR	0.1518	0.2840	0.3157	0.3317	0.3512

그림 2. 빈도수 높은 단어(High)에 대한 등급재현율

그림 2와 3에서 볼 수 있듯이 KR 번역 후보를 찾는 경우, 즉 대상 언어가 KR인 경우는 High에서 높은 결과를 보였고, 그 반대인 경우(원시 언어가 KR)에는 Low에서 높은 결과를 보였다는 것을 알 수 있다. 4.1.1절에서 기

1) http://kldp.net/projects/hannanum
2) http://www.ims.uni-stuttgart.de/projekte/corplex/TreeTagger/



	Top 1	5	10	15	20
KR→FR	0.1192	0.1963	0.2437	0.2799	0.2926
FR→KR	0.0510	0.1228	0.1443	0.1707	0.1880
KR→ES	0.1102	0.2473	0.2882	0.2995	0.3241
ES→KR	0.0925	0.1635	0.2093	0.2330	0.2490

그림 3. 빈도수 낮은 단어(Low)에 대한 등급재현율

술된 정답 사전은 대상 언어가 KR일 경우가 아닌 경우에 비해 상대적으로 많은 정답 단어를 포함하고 있다. 이것으로 볼 때, 여러 뜻을 가지는 단어일수록 문서에 많이 포함된다는 사실을 알 수 있다. 또한 상위 1위부터 5위까지 그래프가 급격하게 기울었다가 이후로 갈수록 점점 기울기가 낮아지는 것을 볼 수 있다. 이런 특징은 Low보다 High에서 좀 더 두드러지지만 대부분의 중요한 단어들(빈도수가 높거나 정확한 번역 후보)은 중하위보다 주로 상위에 포진되어 있다는 것을 의미한다.

그림 4와 5는 모두 일반적인 재현율과 등급재현율로 시스템을 평가한 것으로써 각각 원시 언어 혹은 대상 언어가 KR↔FR, KR↔ES인 것에 대한 결과를 통합하여 평균낸 결과이다. 그림에 나타난 결과들을 보면 심한 경우(KR-ES, 상위 20위)에 15% 정도의 차이가 난다는 것을 알 수 있다. 이는 일반 재현율로 봤을 때 정답 단어의 20% 정도에 가까운 단어들을 시스템이 도출해냈지만, 사실 이 단어들은 사전 안에서 35% 정도를 차지하고 있다는 사실을 의미한다. 또한 상위 1위에서 재현율과 등급재현율의 성능 차이를 살펴보면, KR-ES와 KR-FR 모두의 경우에 High는 대략 10%, Low는 대략 5% 정도의 차이를 확인할 수 있다. 즉, 사전에 있는 정답 단어 중 5% 혹은 10% 정도만을 시스템이 도출해내었다고 생각할 수 있지만 이 단어들이 실제 문서에서 차지하는 중요도는 그 이상이라는 점을 의미한다. 오히려 문서에 별로 나오지 않은 단어를 시스템이 도출해낸 것보다는 문서에서 빈번하게 등장하는 단어를 시스템이 도출해내는 것이 훨씬 중요하기 때문이다. 따라서 이런 경향으로 봤을 때, 본 논문에서 제안하는 평가 방법이 큰 의미가 있다.

5. 결론

본 논문은 기존에 문맥벡터를 이용하여 이중언어 사전을 자동으로 구축하는 방법이 얼마나 효과적인지 새롭게 평가하였다. 일반적인 재현율은 단순히 얼마나 많은 단

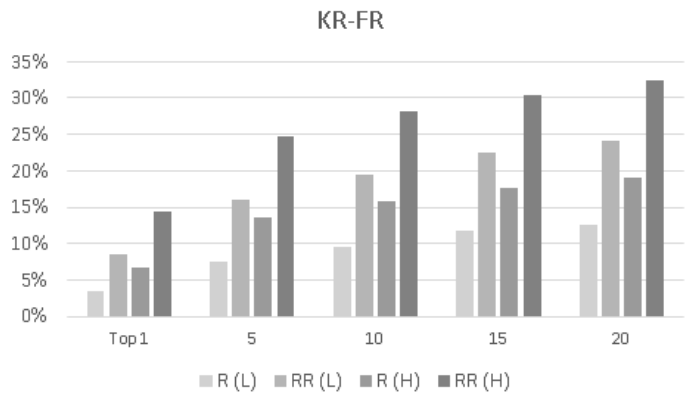


그림 4. 일반 재현율과 등급재현율의 비교1 (KR-FR). L은 Low, H는 High, R은 Recall, RR은 Rated Recall을 의미한다.

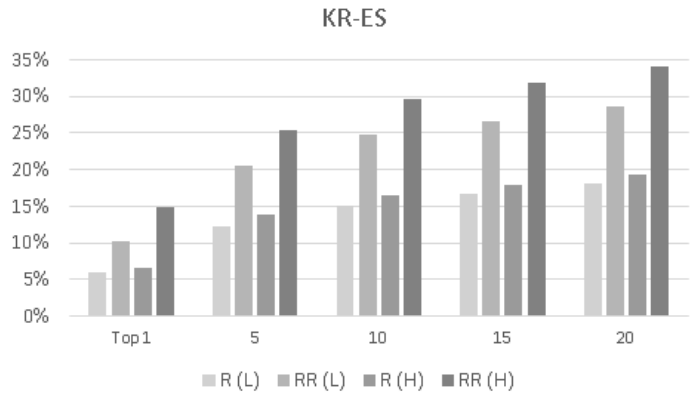


그림 5. 일반 재현율과 등급재현율의 비교2 (KR-ES).

어를 실제 시스템이 결과로 도출해내는지에 대한 평가 방법이지만 모든 단어들에 대하여 똑같이 비율을 지정하여 실제 얼마나 중요한 단어들을 찾아냈는지 판단하기 어렵다. 하지만 본 논문에서 제안한 등급 재현율을 이용하여 평가해보면 시스템이 도출해낸 번역 후보 단어들이 실제 문서에서 얼마나 큰 영향을 끼치는지를 알 수 있기 때문에 시스템이 얼마나 효과적인지 파악할 수 있다는 장점이 있다.

향후 연구로는 스페인어와 불어 이외의 언어에 대하여도 이중언어 사전을 구축해보는 것과 다중단어(multi-word expression)에 대한 연구도 해볼 수 있을 것이다.

감사의 글

본 연구는 미래창조과학부 및 한국산업기술평가관리원의 산업융합원천기술개발사업(정보통신)의 일환으로 수행하였음. [10041807, 지식학습 기반의 다국어 확장이 용이한 관광/국제행사 통역률 90%급 자동 통번역 소프트웨어 원천 기술 개발]

참고문헌

[18] P. Brown, J. Cocke, Stephen A. Della Pietra, V. Pietra, F. Jelinek, J. Lafferty, R. Mercer and P. Roossin 1990 "A statistical approach to machine translation" Coling'90 16(2) pp. 79-85.

[19] J. Nie, M. Simard, P. Isabelle, and R. Durand 1999 "Cross-language information retrieval based on parallel texts and automatic mining of parallel texts from theWeb" In Proc. of the ACM SIGIR pp. 74-81.

[20] T. Tsunakawa, N. Okazaki, and J. Tsujii 2008 "Building Bilingual Lexicons Using Lexical Translation Probabilities via Pivot Languages" In proc. of LREC.

[21] K. Tanaka and K. Umemura 1994 "Construction of a Bilingual Dictionary Intermediated by a Third Language" In Proc. of the Coling'94 pp. 297-303.

[22] L. Nerima and E. Wehrli 2008 "Generating Bilingual Dictionaries by Transitivity" In Proc. of the LREC'08 pp. 2584-2587.

[23] P. Fung 1995 "Compiling Bilingual Lexicon Entries From a Non-Parallel English-Chinese Corpus" In Proc. of the VLC'95 pp. 173-183.

[24] K. Yu and J. Tsujii 2009 "Bilingual dictionary extraction from Wikipedia" In Proc. of the MT Summit XII pp. 379-386.

[25] 서형원, 권홍석, 김재훈 2013 "이중언어 병렬말뭉치와 중간언어를 활용한 이중언어 사전 자동 구축" 한국정보처리학회 춘계학술발표대회 논문집. 제20권. 제1호. pp. 307-310.

[26] F. Bond, R. Binti Sulong, T. Yamazaki, and K. Ogura 2001 "Design and Construction of a machine-tractable Japanese-Malay Dictionary" In Proc. of the MT Summit VIII pp. 53-58.

[27] D. Wu and X. Xia 1994 "Learning an English-Chinese lexicon from a parallel corpus" In Proc. of the AMTA'94 pp. 206-213.

[28] P. Fung and K. Church 1994 "K-vec: A New Approach for Aligning Parallel Texts" In Proc. of the Coling'94 2 pp. 1096-1102.

[29] Y. Chiao and P. Zweigenbaum 2002 "Looking for Candidate Translational Equivalents in Specialized, Comparable Corpora" In Proc. of the Coling'02 pp. 1208-1212.

[30] A. Lardilleux, J. Gosme and Y. Lepage 2010 "Bilingual Lexicon Induction: Effortless Evaluation of Word Alignment Tools and Production of Resources for Improbable Language Pairs" In Proc. of the LREC.

[31] P. Fung 1998 "A Statistical View on Bilingual Lexicon Extraction: From Parallel Corpora to Non-Parallel Corpora" In Proc. of the Parallel Text Processing, pp. 1-17.

[32] B. Daille and E. Morin 2005 "French-English Terminology Extraction from Comparable Corpora" Natural Language Processing - IJCNLP 3651.

[33] E. Prochasson and E. Morin 2009 "Anchor points for bilingual extraction from small specialized comparable corpora" TAL 50(1) pp. 283-304.

[34] R. Rapp 1999 "Automatic Identification of Word Translations from Unrelated English and German Corpora" In Proc, of the ACL'99 pp. 519-526.

[35] T. Dunning 1993 "Accurate methods for the statistics of surprise and coincidence" Coling'93 19(1) pp. 61-74.

[36] H.-W. Seo, H.-C. Kim, H.-Y. Cho, J.-H. Kim and S.-I. Yang 2006 "Automatically Constructing English-Korean Parallel Corpus from Web Documents" Korea Information Processing Society 13(02) pp. 0161-0164.

[37] 박상원, 최동현, 김은경, 최기선 2010 "플러그인 컴포넌트 기반의 한국어 형태소 분석기" 한글 및 한국어 정보처리 학술대회 (HCLT) Poster. pp. 197-201.

[38] H. Schmid 1995 "Improvements in Part-of-Speech Tagging with an Application to German" In Proc. of the ACL SIGDAT-Workshop. pp. 47-50.

한국어 품사 및 동형이의어 태깅을 위한 마르코프 모델과 은닉 마르코프 모델의 비교

신준철○, 옥철영
울산대학교 컴퓨터정보통신공학
ducksjc@nate.com, okcy@ulsan.ac.kr

Comparison between Markov Model and Hidden Markov Model for Korean Part-of-Speech and Homograph Tagging

Joon-Choul Shin○, Cheol-Young Ock
Dept. of Computer Engineering & Information Technology, Ulsan University

요 약

한국어 어절은 많은 동형이의어를 가지고 있기 때문에 주변 어절(또는 문맥)을 보지 않으면 중의성을 해결하기 어렵다. 이런 중의성을 해결하기 위해서 주변 어절 정보를 입력받아 통계적으로 의미를 선택하는 기계학습 알고리즘들이 많이 연구되었으며, 그 중에서 특히 은닉 마르코프 모델을 활용한 연구가 높은 성과를 거두었다. 일반적으로 마르코프 모델만을 기반으로 알고리즘을 구성할 경우 은닉 마르코프 모델 보다는 단순하기 때문에 빠르게 작동하지만 정확률이 낮다. 본 논문은 마르코프 모델을 기반으로 하면서, 부분적으로 은닉 마르코프 모델을 혼합한 알고리즘을 제안한다. 실험 결과 속도는 마르코프 모델과 유사하며, 정확률은 은닉 마르코프 모델에 근접한 것으로 나타났다.

주제어: 마르코프 모델, 은닉 마르코프 모델, 형태소, 형태소분석기, 의미 중의성, 동형이의어

1. 서론

하나의 어절이 가지는 의미는 종종 여러 가지일 수 있으며, 대체로 주변 문맥을 통해 한 가지 뜻으로 해석된다. 이것을 의미중의성이라고 하며, 컴퓨터가 이런 중의성을 자동으로 해결할 수 있다면 자연어처리 시스템이 획기적으로 발전할 것이다.

중의성을 가진 어절에 품사와 의미번호를 표시하여 의미를 명확하게 하는 작업을 태깅이라고 한다. 태깅을 위한 품사 태그는 대표적으로 세종태그셋이 있다. 의미번호는 표준국어대사전의 것이 널리 사용되고 있다. 예를 들어 '나는' 은 다음과 같이 태깅될 수 있다.

●나__03/NP+는/JX
●나__01/VV+는/ETM
●날__01/VV+는/ETM

예와 같이 중의성을 가진 어절은 다양한 품사와 의미로 해석될 수 있으며, 이는 여러 가지의 분석후보들로 표현된다. 일반적인 자동태깅 절차는 하나의 어절을 분석해서 분석후보들을 만든 다음에, 주변 문맥을 통해 적절한 후보 하나를 선택하는 것이다.

하나의 어절에서 적절한 분석후보들을 만드는 것은 오래전부터 많이 연구되어왔다. 초기에는 규칙기반 알고리즘들이 연구되었으며, 세종말뭉치가 정립된 이후에는 기계학습을 활용한 방법들이 연구되고 있다.

적절한 후보를 선택하는 방법은 세종말뭉치로 기계학습을 하고, 인접한 어절이나 공기어를 활용하여 통계적으로 처리하는 방법들이 연구되어왔다. 그러나 대부분 품사만 결정하거나, 정확률이 낮거나, 이 모든 것을 해결하더라도 속도가 느린 단점을 가지고 있다.

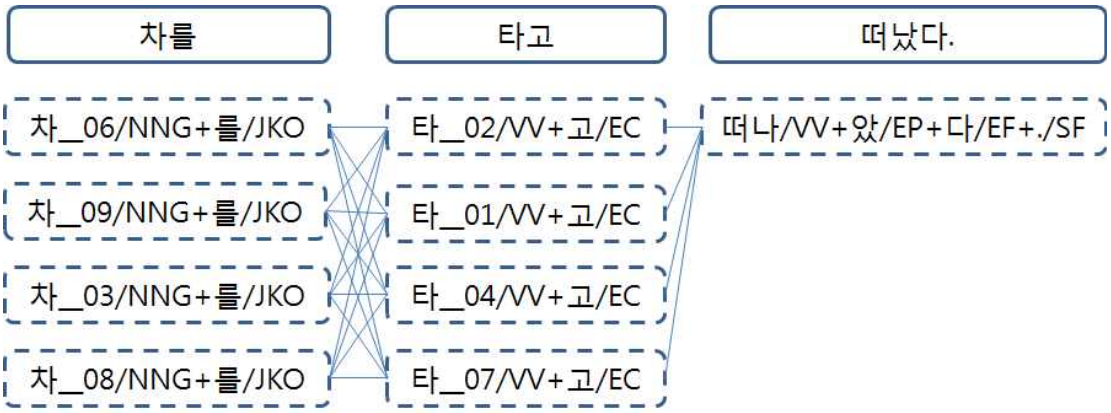
본 논문은 적절한 후보를 선택하는 새로운 알고리즘을 제안한다. 이 알고리즘은 세종말뭉치를 최대한 활용하며, 정확률과 재현율을 최대한 보존하면서 속도를 더 빠르게 하는 것에 목표를 둔다.

2. 관련 연구

자동태깅의 첫 단계는 하나의 어절에서 분석후보들을 생성하는 것이다. 초창기에 어절에서 형태소 원형을 하나씩 복원하는 규칙기반 방법이 주로 연구되었다. 대용량 말뭉치가 정립되기 전에는 기계학습 기반의 분석방법을 적용할 수 없었기 때문이다. 대표적인 연구로 [1]은 다양한 한국어의 불규칙변화를 전산화하여 규칙기반의 분석 방법으로 적용하기 위해 다층 형태론을 제안하였다.

대용량 말뭉치가 정립되고 부터는 기계학습 방법을 활용한 연구가 진행되고 있다. [2]는 대용량 말뭉치에서 기분석 사전과 "부분 어절 기분석 사전"을 만들고, 이를 사용하여 어절을 분석하고 형태소 분석 후보들을 생성하는 알고리즘을 제안한다. 세종말뭉치로 실험한 결과 99.05%의 형태소 분석 재현율(후보 중에 정답이 있을 확률)을 보였으며, 형태소분석을 위한 조합형코드로의 변환 과정, 형태소 분리 및 복원 등의 복잡한 과정이 필요 없으며 띄어쓰기 오류에도 안정적인 특징이 있다.

하나의 어절을 분석하여 나타나는 분석 후보들 중에서 가장 적절한 하나의 후보를 자동으로 선택하기 위한 연



[그림 1 Bigram HMM의 예]

구들이 진행되어왔다. 세종말뭉치가 정립되지 않았을 때에는 내부적으로 자체 제작한 소량의 말뭉치를 사용하였다. 이런 경우에는 정확률이 낮았다[3].

소량의 말뭉치에서 은닉 마르코프 모델(Hidden Markov model : HMM)을 기계학습한 연구가 있다[4, 5, 6]. 말뭉치의 양이 적기 때문에 상대적으로 적은 품사 태그를 사용하여 정확률을 높였다. 비록 소량의 말뭉치였지만 초기에 HMM의 유용성을 실험해본 의미 있는 연구들이였다.

[7]은 KAIST에서 구축한 '대한민국 국어정보 베이스'의 품사 부착된 말뭉치를 사용하여 HMM을 학습하였다. 이 방법에서는 수작업으로 구축되는 말뭉치의 특성상 나타날 수밖에 없는 오류에 대해서 저신뢰도 구간 검사를 통해 학습 말뭉치의 신뢰도를 높여 태깅 성능의 향상을 모색하였다.

[8]은 비교적 최근의 HMM 연구로, 약 천만어절의 세종말뭉치를 사용하였다. 우선 어절별로 분석 후보를 생성하기 위해서 CKMA[2]를 사용하였으며, 하나의 후보를 선택하기 위해서 여러 가지의 HMM을 단계별로 적용하였다. 우선 정확률이 높은 모델을 적용해보고, 재현에 실패한 경우(학습된 적이 없는 경우다.) 정확률이 낮더라도 재현율이 높은 모델을 적용하였다. 동형이의어까지 태깅하면서 정확률이 매우 높은 결과를 보였다.

3. 알고리즘

분석 후보에서 의미와 품사가 적합한 하나의 후보를 선택하기 위해서 연구된 방법 중 가장 알려진 것은 HMM이다. HMM은 은닉층(분석 후보, 형태소 원형) 정보간의 전이확률로 최적의 전이열을 찾는 모델이다. HMM 보다 단순한 모델인 마르코프 모델(Markov model : MM)은 오직 관찰되는 정보만을 통해 현재 또는 미래의 은닉 정보를 추측한다.

3.1 은닉 마르코프 모델

HMM은 은닉층을 사용하는 모델이며, 태깅에서는 인접한 어절들의 분석 후보들이 대표적인 은닉층이다. 사람이 수동으로 태깅하지 않은 상태에서는 인접한 어절들이 중의성을 가지면 오직 여러 가지의 분석 후보들만 알 수 있다. 이런 특성 때문에 HMM은 인접한 어절들의 분석 후보가 다양할수록 연산량이 많아진다.

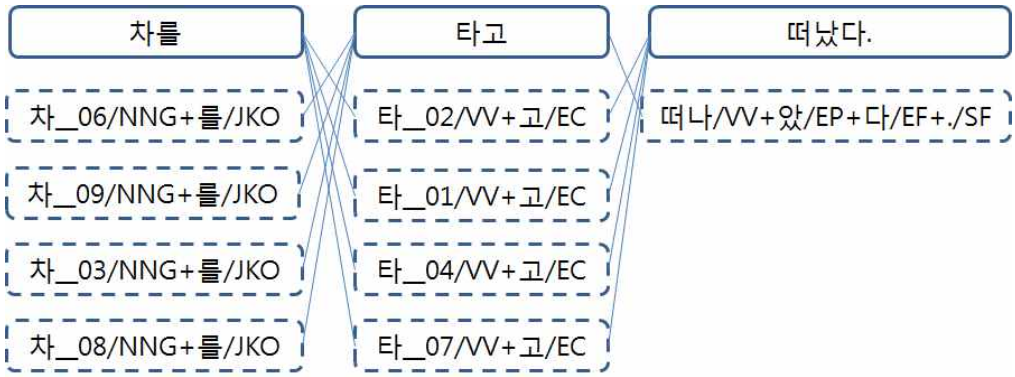
직접적으로 인접한 어절만을 고려하는 HMM을 바이그램(Bigram) HMM이라고 하며, [그림 1]은 Bigram HMM으로 분석 중인 예이다. 3개의 어절로 구성된 문장은 각 어절별로 분석 후보들이 있으며, HMM은 인접 어절의 후보와 자기 어절의 후보 간의 전이확률을 계산하고, 전이확률이 가장 높은 최적열을 Viterbi 알고리즘으로 찾는다.

Bigram HMM의 전이확률은 좌측 어절의 특정 후보와 우측 어절의 특정 후보가 동시에 출현할 확률을 표현한다. [그림 1]안에서 예를 들자면, 첫 번째 전이확률은 좌측 어절이 '차를__06/NNG+를/JKO'이고 우측 어절이 '타__02/VV+고/EC'일 확률을 의미한다. 그러나 실제로 후보 내용 전체를 모델에서 사용하기에는 학습데이터의 용량과 재현율에서 비효율적인 문제가 발생하기 때문에 [8]에서는 인접 어절의 가장 가까운 형태소만을 모델에서 사용하는 모델을 제안한다. 예를 들어 '차를'의 후보들을 처리하는 과정에서 첫 번째로 계산되는 전이확률은 '차__06/NNG+를/JKO'와 '타__02/VV*'가 동시에 출현할 확률이다. '*'는 어절의 나머지 부분이 무엇이 되던 동일한 것으로 간주한다는 의미다.

태깅 시점에서는 좌측 어절에는 후보가 4가지이기 때문에, 정확히 어떤 후보가 정답인지 알 수 없다. 따라서 HMM에서는 모든 가능한 후보 간의 전이확률을 계산해야 한다. [그림 1]에서 차를'과 '타고'의 후보가 각각 4개씩이므로 두 어절 사이에서 구해야하는 전이확률은 총 16가지이고, '타고'와 '떠났다.' 사이의 것을 합치면 총 20개이다.

[그림 1]의 모습은 일부 후보를 생략한 것이며, 실제로는 더 많은 후보가 존재(기분석 사전) 하거나, 생성될 수 있기 때문에 더 많은 전이확률 정보가 필요하다. 각 전이확률을 학습된 정보에서 찾는 과정과, 모든 전이확률들로부터 최적열을 구하는 과정에서 HMM은 MM보다 컴퓨터 자원을 많이 사용해야한다.

3.2 마르코프 모델



[그림 2 Bigram MM의 예]

MM은 오직 관찰된 정보만을 이용하여 알고자 하는 은닉된 정보를 추측한다. 태깅에서는 인접한 어절들과 현재 어절의 표층형이 관찰된 정보고, 현재 어절의 분석 내용이 추측하고자 하는 은닉 정보다.

MM 중에서 직접적으로 인접한 어절만을 고려하면 Bigram MM이라고 하며, [그림 2]는 Bigram MM으로 분석 중인 예이다. 각 어절의 후보는 인접 어절의 표층형과의 전이 확률을 고려한다. '차를'의 후보 4개는 인접 어절인 '타고'와의 전이확률만을 계산하고, '타고'의 후보 4개는 '차를'과의 전이확률을, 그리고 '떠났다.'와의 전이확률 4개를 계산한다. 따라서 그 두 어절 사이에 필요한 전이확률 계산은 총 8개이고, '타고'와 '떠났다' 사이의 것 4개를 합치면 총 12개로 [그림 1]의 것보다 적다.

[8]에서 제안하는 모델과 마찬가지로, 인접한 어절의 표층형 전체를 전이모델에 포함하는 것은 학습용량과 재현율 측면에서 비효율적이기 때문에 일부만을 포함하는 것이 좋다. 1개의 어절만 포함하기엔 인접한 어절을 너무 모호하게 표현하고, 3어절 이상을 포함하면 어절 내용을 이미 거의 다 포함하기 때문에 본 논문에서는 인접한 어절들에 서로 가장 가까운 2음절씩을 학습하는 것을 제안한다.

예를 들어 '타고'와 '떠났다.' 사이에서 첫 번째로 구하는 전이확률은 '*타고'와 '떠났*'을 통해서 구한다. 말뭉치에서 위의 각각 2음절을 모두 포함하는 인접한 쌍 어절을 찾고, 해당 어절에 태깅된 형태소가 분석 후보에 포함되는지 확인하는 방식이다. 이 모델을 Bi-Syllable of End and First(BIS-EF)이라고 한다.

3.3 MM과 음운 변동

태깅에서 MM모델이 가지는 문제점은 표층형을 중심으로 전이모델이 구성되기 때문에 동사의 음운변동에 취약하다는 것이다. 예를 들어서 "타고 떠났다."와 "타고 떠난다."는 표층형이 다르기 때문에 구분된다. 둘 다 우측 어절은 '떠나/VV'를 사용하지만 2번째 음절이 '났'과 '난'으로 다르기 때문에 한쪽 문장을 학습하면 다른 한쪽 문장이 입력으로 들어올 때 학습된 내용이 적용되지 못한다. 이런 특성은 HMM을 사용한 방법[8]보다 재현율이 낮아질 요소다.

때문에 본 논문에서 제안하는 BIS-EF모델에서는 일반적인 재현에 실패할 경우에 표층형 대신 형태소 원형을 사용하는 단계를 전이모델에 추가한다. 이 경우 전이확률의 가짓수는 Bigram HMM과 마찬가지로 두 어절의 분석 후보 수의 곱만큼 많아지게 되지만, 오직 두 어절 사이에서 전이확률만 계산하여 결정하기 때문에 최적열을 연산하는 Viterbi과정은 없다.

3.4 MM과 부사절

Bigram을 기반으로 하는 모델은 인접어절을 기준으로 의미를 결정하기 때문에 부사는 악영향을 줄 수 있다. 예를 들어서 "차를 빨리 마셨다."를 처리한다면, '차를'의 의미를 결정지어주는 어절은 '마셨다.' 이지만 부사 '빨리'에 의해 인접하지 못하게 된다. 이런 경우에 BIS-EF는 부사와 함께 그 다음 어절도 마치 인접한 것과 마찬가지로 취급하여 전이확률을 계산한다. HMM에 비하여 이런 튜닝은 MM에서 더 쉽게 구현이 가능하다.

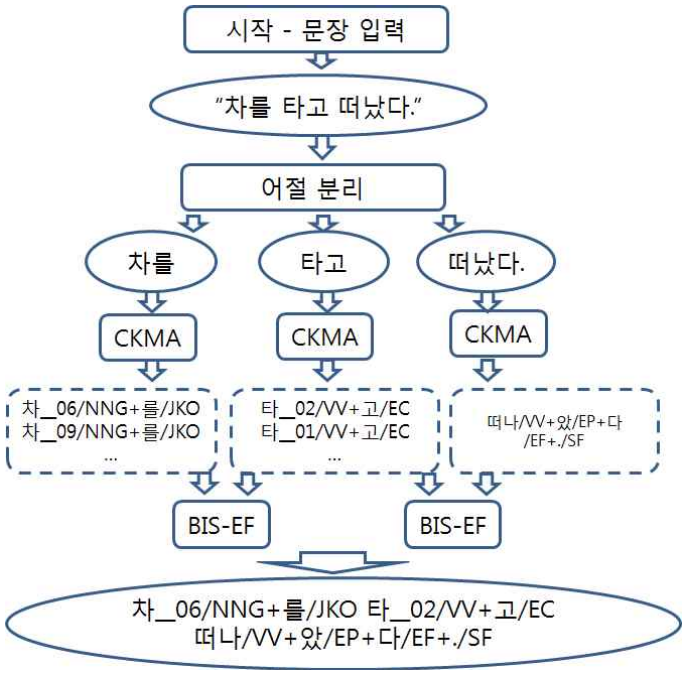
4. 전체 시스템 구성

전체 시스템은 [그림 3]과 같이, 먼저 문장을 입력받으면서 시작한다. 문장은 공백을 기준으로 어절들로 분리되고, 각 어절은 CKMA[2]모듈에 입력되면 분석 후보들이 출력된다. 인접한 어절 쌍 단위로 BIS-EF모듈에 입력되면 각 후보별로 확률점수가 계산되고, 어절별로 가장 높은 점수를 가진 후보가 선택되어 최종 결과로 출력된다.

5. 실험 결과

본 논문의 실험을 위해 사용된 CKMA는 [2]이후 계속된 개선 작업을 통해 일부분이 변경된 것으로 정확률이 조금 더 높다. 때문에 HMM과의 엄격한 비교를 위해 [8]의 실험을 수정된 CKMA와 동일한 말뭉치로 다시 시행하였다.

말뭉치는 일부 오류가 수정된 세종말뭉치를 사용하였



[그림 3 전체 시스템]

으며, 약 1천만 어절을 포함한다. 문장 단위로 잘라서 10문장 중에 1문장씩을 제외하여 테스트셋을 구성하였고, 나머지 9문장씩을 학습에 사용하였다.

[표 1 HMM과 BIS-EF 결과 비교]

	정확률(%)	시간(초)
HMM	96.5393	21.328
BIS-EF	96.3361	9.999

[표 1]은 HMM과 BIS-EF를 비교 실험한 결과다. 둘 다 완전히 동일한 말뭉치와 테스트셋을 사용했으며, 동일한 버전의 CKMA를 사용하여 분석 후보를 생성하였다. 시간 측정은 CKMA에 소모된 시간을 제외한 것으로 순수하게 HMM 또는 BIS-EF에서 소모된 시간만 측정한 것이다.

[표 1]의 실험 결과에 따르면 본 논문이 제안하는 BIS-EF는 HMM에 비하여 약 0.2% 낮은 정확률을 보이지만 속도가 약 2.1배 빠른 것으로 나타난다.

5. 결론

품사 및 동형이의어가 태깅된 형태소 분석 후보에서 적절한 후보 1개를 선택하는 방법으로 HMM은 이미 많이 연구된 모델이다. HMM으로도 높은 정확률을 보이고 있으나, 본 논문에서는 속도를 더욱 개선하기 위해 더 단순한 MM모델에 기반을 둔 BIS-EF를 제안한다.

BIS-EF는 기존 HMM에 비하여 정확률이 0.2% 낮지만, 속도는 2배 이상 빠른 결과를 보여주었다. 따라서 정확률 보다 속도를 필요로 하는 곳에서 더 적합할 것이다.

또한 BIS-EF는 한 가지 전이모델만을 사용하고 있어 개선의 여지가 많이 남아있다. 좀 더 다양한 전이모델을 추가해서 단계적으로 또는 동시에 적용한다면 정확률을 더욱 향상시킬 수 있을 것으로 예상된다.

참고문헌

[1] 강승식, "다층 형태론과 한국어 형태소 분석 모델", 제6회 한글 및 한국어 정보처리 학술발표 논문집, pp.140-145, 1994.

[2] 신준철, 옥철영, “기분석 부분 어절 사전을 활용한 한국어 형태소 분석기”, 정보과학회논문지 : 소프트웨어 및 응용, 제39권, 제5호, pp.415-424, 2012.5.

[3] 김영훈, "한국어에 적합한 효율적인 품사 태깅", 한국콘텐츠학회논문지, 제2권, 제2호, pp.98-102, 2002.6.

[4] 신중호, 한영석, 박영찬, 최기선, "어절구조를 반영한 은닉 마르코프 모델을 이용한 한국어 품사태깅", 한글 및 한국어정보처리 학술대회, p389-394, 1994.10.

[5] 김재훈, 임철수, 서정연, "은닉 마르코프 모델을 이용한 효율적인 한국어 품사의 태깅", 정보과학회논문지, 제22권, 제1호, pp.136-146, 1995.1.

[6] 강유환, 서영훈, "어절간 주품사 정보와 제약 규칙을 이용한 한국어 품사 태깅 시스템", 제11회 한글 및 한국어 정보처리 학술대회, pp.433-437, 1999.10.

[7] 설용수, 김동주, 김규상, 김한우, "말뭉치 오류를 고려한 HMM 한국어 품사 태깅 시스템", 한국컴퓨터정보학회지, 제15권, 제1호, pp.117-124, 2007.

[8] 신준철, 옥철영, "한국어 품사 및 동형이의어 태깅을 위한 단계별 전이모델", 정보과학회논문지 : 소프트웨어 및 응용, 제39권, 제11호, pp.889-901, 2012.

Y-HisOnto: Q&A 시스템에서의 활용을 위한 역사 온톨로지 모형

이인근○, 정재은, 황도삼
경북대학교 의료정보학과○, 영남대학교 컴퓨터공학과
inkeunlee@gmail.com○, j2jung@gmail.com, dshwang@yu.ac.kr

Y-HisOnto: A History Ontology Model for Q&A System

In Keun Lee○, Jason J. Jung, Dosam Hwang
Department of Medical Informatics, Kyungpook National University School of Medicine○
Department of Computer Engineering, Yeungnam University

요 약

본 논문에서는 시간 개념이 포함된 역사적 지식을 표현할 수 있는 사건 온톨로지(event ontology) 기반의 역사 온톨로지 모형인 Y-HisOnto 를 제안한다. 제안한 역사 온톨로지 모형은 기존의 온톨로지에서 사용되는 이진 관계(binary-relationship)로 표현된 단편적 지식들을 조합하여 다진 관계(n-ary relationship)를 이용하여 역사적 사건 관련 지식을 표현한다. 제안한 온톨로지 모형에 기반하여 사건 중심의 지식을 온톨로지로 구축하고, 사건 관련 질의에 대해 온톨로지 논리 검색 실험을 수행함으로써 제안한 온톨로지 모형이 Q&A 시스템에서 효과적으로 활용될 수 있음을 확인한다.

주제어: 사건 온톨로지(event ontology), 역사 온톨로지, 질의응답 시스템(Q&A System)

1. 서론

컴퓨터를 통한 지식의 구축 및 활용 요구에 따라 질의응답(Question and Answer: Q&A) 시스템에 관한 연구가 수행되어 왔고[1-3], 컴퓨터를 통한 지식의 표현을 위해 최근에는 온톨로지를 이용하는 방법에 대한 연구가 활발히 수행되고 있다[4-6]. T.Gruber의 온톨로지 정의에 의하면, 온톨로지는 광범위한 지식 보다는 관심 영역 내의 합의된 지식을 체계적으로 구축하는 것이 온톨로지의 활용 측면에서 더 의미가 있다고 해석할 수 있다. 이에 따라 다양한 분야에서 분야 전문 지식을 온톨로지로 구축하고 효과적으로 활용하기 위한 연구가 수행되어 왔다. 특히, 유럽에서는 VICODI project[7]를 통해 유럽의 역사적 지식을 온톨로지로 표현하려는 연구가 수행되었다. 타 분야의 지식과 달리, 역사적 사실의 전개는 주로 사건(event)에 기반하여 기술되고 있으며, 사건은 시간에 종속되어 표현된다. 따라서 이러한 복잡한 형태의 역사적 지식을 표현하고 활용하는 방법의 연구가 필요하다.

역사에서 하나의 사건은 주체, 시간, 대상, 장소, 행위 등과 같이 개념들이 복잡한 의미 관계를 형성하고 있으며, 사건이 일어나는 원인이나 특정 사건에 의해 발생되는 결과의 표현을 위해서는 역사적 지식이 사건 중심으로 표현되어야 한다. 실제 RDF(S), F-logic, OWL (web ontology language)등과 같은 다양한 온톨로지 언어[4]들이 지식의 표현을 위해 사용되고 있고, 이들 언어들은 이진 관계(binary-relationship)로 단편적 지식을 표현한다. 특히 OWL[8]에서는 지식의 표현을 위해 개념(class), 개체(individual), 속성(property), S-P-O (subject-predicate-object) 형태의 개념간의 관계(relationship), 개념의 범위를 지정하기 위한 제약사항(restriction), 그리고 string, integer 등과 같은 데이터(data) 등의 온톨로지 요소를 정의하고 있다. 그러나 OWL에서와 같이 단순한 이진관계로는 복잡한 사건 지식을 표현하기 어렵다.

따라서 본 논문에서는 시간 개념이 포함된 역사적 지식을 사건 기반으로 구축하고 활용할 수 있는 온톨로지 모형인 Y-HisOnto 를 제안한다. 제안한 온톨로지 모형은 이진 관계를 조합하여 다진 관계(n-ary relationship)를 표현하는 사건 온톨로지(event ontology)[5,6]를 이용한다. 그리고 제안한 온톨로지 모형에 기반하여 사건 중심의 지식을 구축하고, 제안한 온톨로지의 Q&A 시스템에서의 효용성을 확인하기 위해 온톨로지 논리 검색 실험을 수행한다.

2. 역사 온톨로지 모형의 설계

이진관계를 이용하여 이벤트 중심의 역사 지식을 표현하기 위해서 그림 1과 같은 형태로 온톨로지를 구성한다. 즉, 특정 “사건” 개념을 중심으로 사건의 부가적인 정보를 표현하기 위해, 사건을 유발한 “인물”, 사건이 발생한 “장소”, 사건의 발생 및 종료 시간, 그리고 사건의 발생 목적과 해당 사건으로 인해 야기되는 “사건”등과 같은 정보가 필요하다. 그림 1에서 보는 바와 같이, 이러한 정보를 온톨로지로 표현하기 위해 “인물”, “장소”, “사건”을 개념으로 표현하고, 화살표 근처의 “주체”, “장소”, “인과”, “목적”은 모두 개념 간의 관계 형성을 위한 속성으로 나타낸다. 특히, “시작시간”과 “종료시간”의 속성은 사건의 발생 및 종료 시점을 나타내는 것으로, 그 값은 정수(integer)로 표현하도록 하였다. 그러나 이 값은 날짜

(datetime) 등과 같은 다양한 자료형으로 정의할 수 있다.

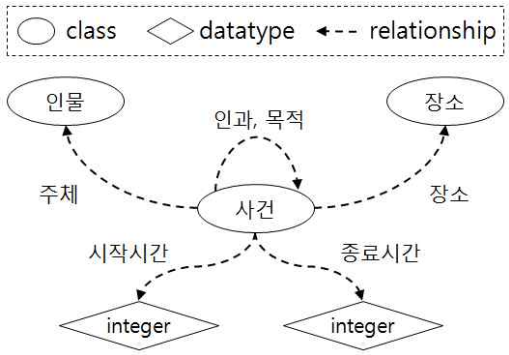


그림 1. 사건 지식 표현을 위한 온톨로지 스키마 예

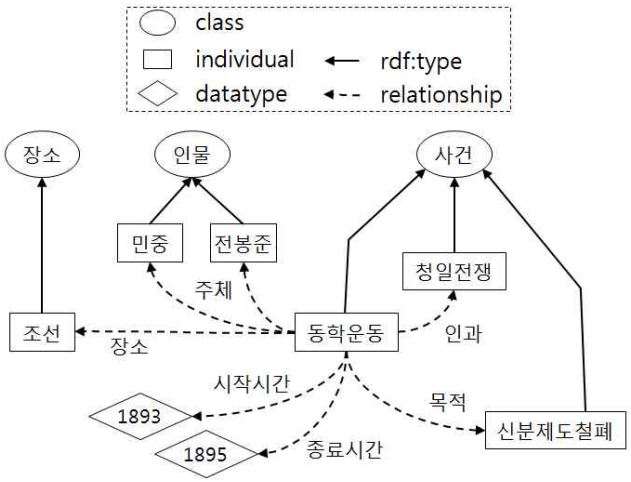


그림 2. 역사적 사건 지식의 온톨로지 표현 예

그림 2는 "전봉준을 중심으로 신분제도철폐를 위해 조선에서 일어난 민중 봉기인 동학운동은 청일전쟁의 원인이 되었다"의 역사적 사건 지식을 그림 1의 온톨로지 스키마에 기반하여 온톨로지로 표현한 예를 보인다. 즉, [표현 1]에서와 같이 이진관계들을 조합으로 하나의 사건 지식은 상세하게 나타낼 수 있다. [표현 1]에서는 일차 논리(First-order logic)를 이용하여 P(S, O) 형태로 사건 지식을 나타낸 것이다.

[표현 1]

주체(동학운동, 민중)	주체(동학운동, 전봉준)
장소(동학운동, 조선)	인과(동학운동, 청일전쟁)
시작시간(동학운동, 1893)	종료시간(동학운동, 1895)
목적(동학운동, 신분제도철폐)	

3. 역사 온톨로지 모형의 활용

제안한 온톨로지 모형에 기반하여 특정 역사적 사건 지식을 Protégé[9]를 이용하여 온톨로지로 구축하고, 구축한 온톨로지를 Q&A 시스템에서의 활용에 대한 효용성을 확인하기 위해 SPARQL[10]을 이용하여 몇 가지 질의에 대한 온톨로지 논리 검색을 수행한다. 그림 3은 온톨로지에 정의한 개체(individual)를 보이며, 그림 4는 개체간의 관계를 통해 사건 지식을 표현한 예를 보인다. 또한, 그림 5는 Protégé를 이용하여 역사 온톨로지를 구축하는 것을 보인다. 본 논문에서 온톨로지 요소에 대한 Namespace는 "http://ontology.yu.ac.kr/event"이며, 이에 대한 Prefix를 "ynu"로 지정하였다. 그러나 그림 3과 4에서는 편의상 이를 생략하였다.

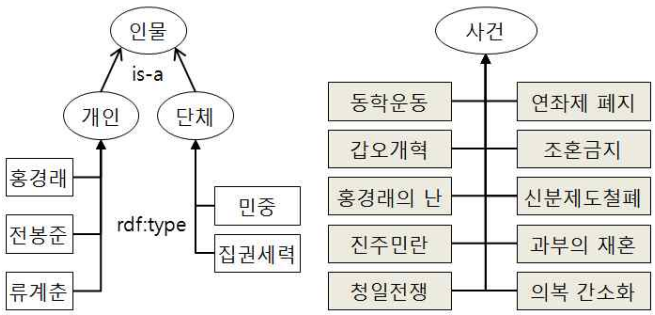


그림 3. 역사 온톨로지에서의 개체 정의

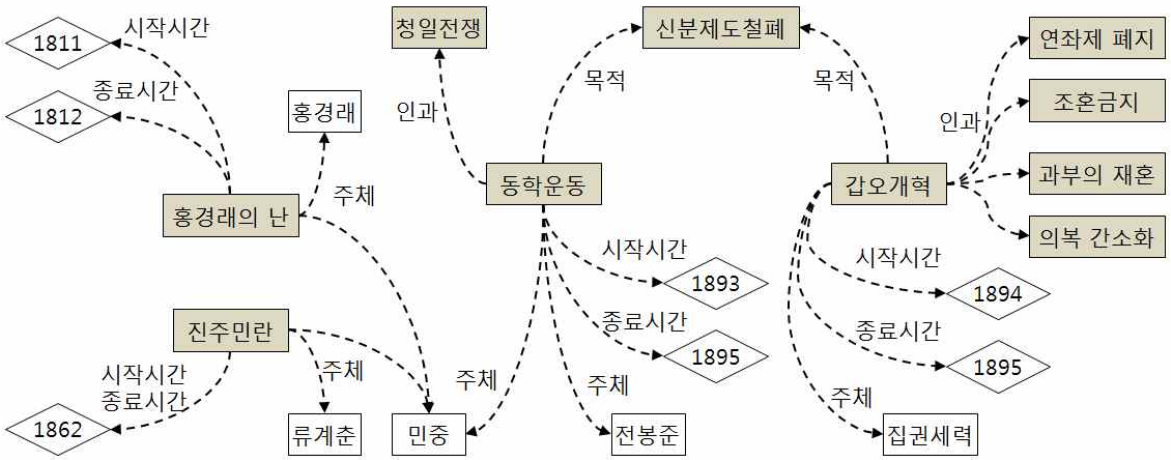


그림 4. 역사 온톨로지 구축 예

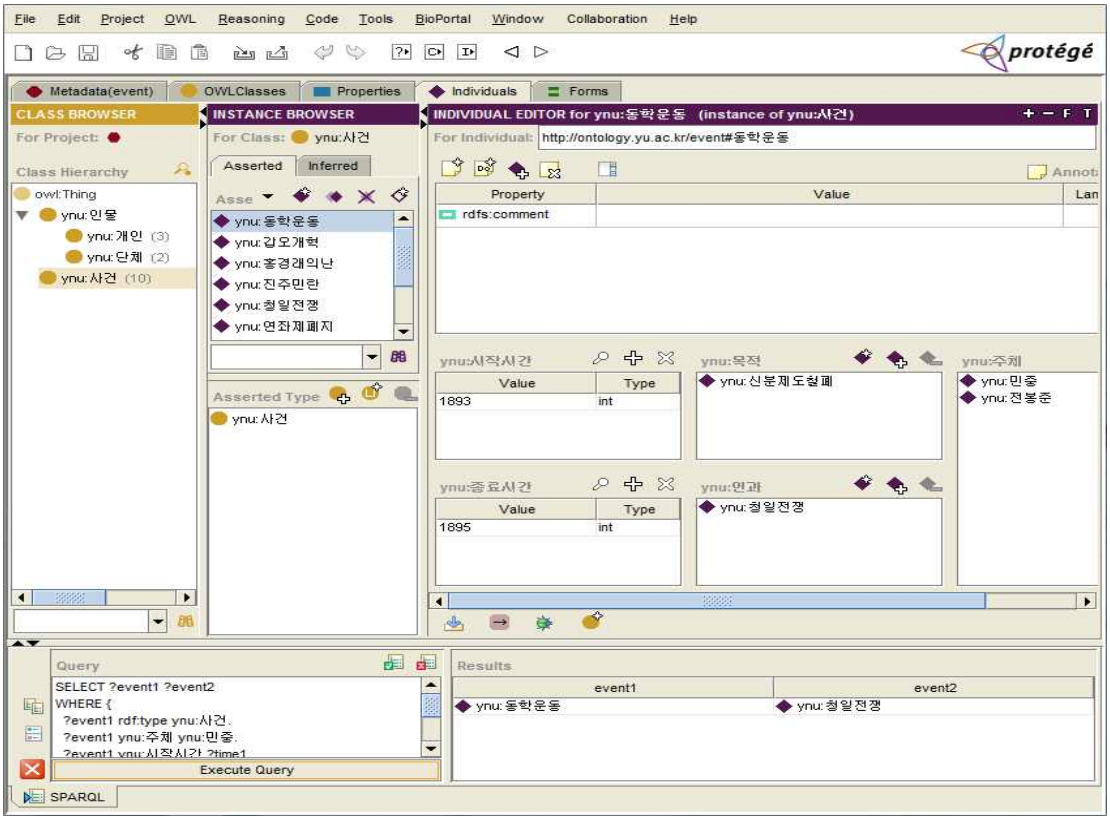


그림 5. Protégé를 이용한 역사 온톨로지 구축

특정 Q&A 시스템에서 구축한 역사 온톨로지를 이용하여 사용자의 3 가지 질의에 대해 답을 제시하는 상황을 가정하여 실험을 수행하였다. 온톨로지의 논리 검색은 Protégé에서 제공하는 검색엔진 및 검색 인터페이스를 이용하였다. 논리 검색을 위한 사용자의 3 가지 질의는 다음과 같다.

[질의1] "민중" 이 일으킨 사건은 무엇인가?
[질의2] "갑오개혁" 으로 인해 야기된 사건은 무엇인가?
[질의3] 고종의 재위기간(1863~1897) 중 일어난 "민중" 사건(1)과 그로 인해 야기된 사건(2)은 무엇인가?

[질의1]로부터 주체가 "민중" 인 사건의 검색을 위해 [SPARQL 질의 1]을 생성하여 논리검색을 수행하였고, 그림 6과 같이 검색 결과로 "홍경래의난", "동학운동", "진주민란" 을 얻었다.

[SPARQL 질의 1]

```
SELECT ?event
WHERE {
  ?event rdf:type ynu:사건.
  ?event ynu:주체 ynu:민중.
}
```

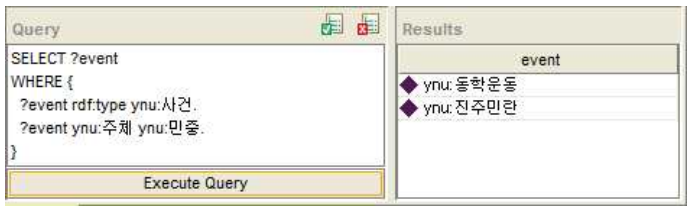


그림 6. [질의1]에 대한 온톨로지 논리 검색

[질의2]로부터 사건 "갑오개혁" 이 원인이 되는 사건의 검색을 위해 [SPARQL 질의 2]을 생성하여 논리검색을 수행하였고, 그림 7과 같이 검색 결과로 "과부의 재혼", "조혼금지", "연좌재폐지", "의복간소화" 를 얻었다.

[SPARQL 질의 2]

```
SELECT ?event
WHERE {
  ynu:갑오개혁 ynu:인과 ?event.
}
```

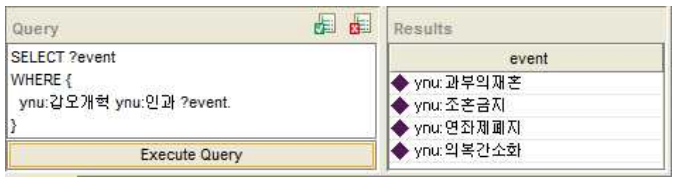


그림 7. [질의2]에 대한 온톨로지 논리 검색

[질의3]으로부터 1863년~1897년 사이에 발생한 사건으로써 주체가 “민중”인 사건(1)의 검색, 그리고 사건(1)과 인과관계가 있는 사건(2)의 검색을 위해 [SPARQL 질의 3]을 생성하여 논리검색을 수행하였다. 검색 결과를 통해 그림 8과 같이 사건(1)은 “동학운동”을, 사건(2)는 “청일전쟁”임을 알 수 있었다.

[SPARQL 질의 3]

```
SELECT ?event1 ?event2
WHERE {
  ?event1 rdf:type ynu:사건.
  ?event1 ynu:주체 ynu:민중.
  ?event1 ynu:시작시간 ?time1.
  ?event1 ynu:종료시간 ?time2.
  FILTER ( ?time1 >= 1863 && ?time2 <= 1897  )
  ?event1 ynu:인과 ?event2.
}
```

그림 8. [질의3]에 대한 온톨로지 논리 검색

본 실험을 통해 역사적 사건 지식을 온톨로지로 구축하고, Q&A 시스템에서 복잡한 사용자의 질의에 대해 논리적 검색을 통해 답을 제공할 수 있음을 확인하였다.

4. 결론

본 논문에서는 복잡하게 표현되는 역사적 사건을 온톨로지 모형인 Y-HisOnto를 설계하여 제시하고 구축함으로써 Q&A 시스템에서의 활용 가능성을 확인하였다.

구축한 온톨로지의 활용을 위해서는 온톨로지의 검색을 통해 지식을 추출해야 한다. 또한 검색 대상과 온톨로지 모형의 설계에 따라 검색 방법이 달라질 것이다. 즉, 본 실험에서 사용한 SPARQL 질의는 실험을 위해 구축한 역사 온톨로지를 기반으로 작성한 것이다. 따라서 만일 온톨로지 스키마가 새롭게 정의되고, 그에 기반하여 새롭게 온톨로지가 구축된다면 좀 더 다양한 조건의 검색질의가 생성되어야 한다.

역사적 지식을 정형화하여 표현함에 있어 다음과 같은 문제들을 고려해야 한다. 즉, (1)동일한 사건 및 대상의 상태는 시간에 따라 변화할 수 있고, (2) 역사적 기록은 주관적이어서 온톨로지에서의 지식 표현의 일관성의 유지가 어려우며, (3)역사적으로 복잡한 사건 간의 인과관계를 단편적으로 정의하기가 어렵다. 따라서 본 논문에서 제시한 온톨로지 스키마는 모든 역사적 지식을 구축하기에는 무리가 있다.

본 논문에서는 시간 개념을 포함된 사건 지식 및 그들 사이의 인관관계를 형성하기 위한 방법을 제안하고, 그의 활용 가능성을 확인하였다는 것에 의미가 있다고 본다. 또한, 역사 지식의 정형화 및 온톨로지로의 표현 방법에 대한 구체적인 연구는 지속적으로 수행되어야 하며, 이에 대한 연구는 향후 지속할 계획이다.

Acknowledgement

본 연구는 미래창조과학부 및 한국산업기술평가관리원의 산업융합원천기술개발사업의 일환으로 수행하였음 [10044457, 자율지능형 지식/기기 협업 프레임워크 기술개발]

참고문헌

[1] S. Kim, D. Baek, S. Kim, and H. Rim “Question Answering Considering Semantic Categories and Co-occurrence Density,” In Proceeding of 9th Text Retrieval Conference, pp. 317-325, 2000.

[2] K.C. Litkowski, “CL Research Experiments in TREC-10 Question Answering,” In Proceeding of 10th Text Retrieval Conference, pp. 122-131, 2001.

[3] S. Vassiliadis, G. Triantafyllos, and W. Kobrosly, “A Fuzzy Reasoning Database Question Answering System,” IEEE Transactions on knowledge and data engineering, Vol. 6, No. 6, pp. 868-882, 1994.

[4] A.G. Perez, M.F. Lopez, and O. Corcho, Ontological Engineering, Springer, 2005.

[5] H. Lin and J. Liang, “Event-based ontology design for retrieving digital archives on human religious self-help consulting,” In Proceeding of IEEE International Conference on EEE'05, pp. 522-527, 2005.

[6] Y. Raimond and S. Abdallah, “The Event Ontology,” http://motools.sourceforge.net/ event/event.html, 2004.

[7] VICODI project, http://www.vicodi.org, 2013.

[8] OWL Web Ontology Language Guide, http://www.w3.org/TR/owl-guide, 2009.

[9] Protégé, http://protege.stanford.edu, 2013.

[10] SPARQL Query Language for RDF, http://www.w3.org/TR/rdf-sparql-query, 2013.

블로그 포스트의 자동 분류 시스템

조희선○, 김수아, 이현아
금오공과대학교 컴퓨터소프트웨어공학과
shinhwa3528@naver.com, sa4956@nate.com, halee@kumoh.ac.kr

Automatic Classification of Blog Posts

Hee-Sun Jho○, Su-Ah Kim, Hyun-Ah Lee
Kumoh National Institute of Technology, Department of Computer software Engineering

요 약

편리한 블로그 사용과 블로그에서의 정보 탐색을 위해서는 내용에 기반한 분류가 필요하다. 대부분의 블로그 사이트에서는 내용 기반 분류를 제공하고 있으나, 블로거들은 자신이 작성한 블로그에 대한 수동 분류를 입력하지 않는 경우가 많다. 본 논문에서는 분류가 제공되는 블로그 사이트에서 각 분류별 문서를 수집하고, 어휘빈도와 문서빈도, 분류별 빈도를 활용하여 문서 내 어휘의 자질 가중치를 부여하고, 다양한 학습기를 이용하여 분류 모델을 생성한 뒤 블로그의 특성에 적합한 자질 추출 알고리즘과 분류 알고리즘을 찾아낸다. 실험에서는 본 논문에서 고안한 CTF-IECDF와 나이브 베이즈 멀티노미얼로 조합한 분류 모델이 75.40%의 분류 정확률을 보였다.

주제어: 블로그, 자동 분류, 자질 추출

1. 서론

정보 개방과 사용자 참여를 가능하게 하는 웹 2.0의 시대에 블로그는 웹 2.0의 특성을 살린 1인 미디어로 부상하고 있다. 최근에는 많은 이용자들이 블로그를 개인적 기록이나 사회 참여의 도구로 사용할 뿐만 아니라, 취미나 관심 분야의 정보 획득 및 공유의 목적으로 많이 사용하고 있다[1]. 블로그 사용자가 늘어남에 따라서 블로그 사이트에서는 주제와 목적에 맞게 블로그를 분류하는 서비스를 제공하고 있다. 블로그 글이 주제에 맞게 분류가 되어있다면, 블로그를 통해 정보를 찾으려는 사용자는 많은 시간과 노력을 감소할 수 있다. 하지만, 다양한 분야(예를 들어 영화, IT, 주식)의 블로그 포스트는 하나의 분류로 대응시키기 적합하지 않을 수 있고, 블로그 포스트를 작성할 때 마다 포스트의 분류를 결정하게 하는 네이버 블로그의 경우 대부분의 블로거들이 기본 분류를 선택하는 양상을 보이고 있다.

본 논문에서는 네이버 블로그에서 주제별 분류가 등록된 포스트들을 수집하고, 이를 학습 데이터로 사용하여 자동으로 포스트의 주제별 분류를 추천하기 위한 시스템을 제안한다. 시스템에서는 분류의 특성을 추출하기 위하여 네가지의 방식을 제시한다. 분류에서는 다양한 학습방법을 적용해 본다. 각각의 결과에 대한 분석과 평가를 통해 블로그 글의 특성에 맞는 특성 추출 알고리즘과 분류 알고리즘으로 블로그 자동 분류 시스템을 구현한다.

2. 관련 연구

문서에 대한 자동 분류 연구에서는 문서에 포함된 단어의 특성을 주로 이용한다. 국내 연구 중에서 [2]는 웹 문서에 대한 텍스트 자동 분류를 위한 특성 추출 기법을 제안한다. 학습 문서 벡터는 웹 디렉터리 내의 문서로부터 추출된 단어 및 관련 문서를 기반으로 구성하였으며, 학습 문서 구성 후 SVM 학습기를 통해 모델을 구성하여 문서 분류를 수행하였다. 이 연구에서는 클래스간의 연관성이 높은 경우 낮은 성능을 보이는 문제점이 있었고 추가적인 학습 문서의 정제가 필요하였다.

[3]에서는 한글 웹 문서에 사용된 한글 형태소 및 키워드의 빈도에 기초하여 문서의 특성을 추출하는 방법을 제시하고, 이를 기초로 비구조적인 문서의 주제를 자동으로 분류하는 방법을 제시하였다. 의사 결정 트리, 신경망 모델 및 SVM 방법을 사용하였으며, 주제간 구분이 명확할수록 정확도는 증가함을 보였다.

3. 블로그 포스트의 자동 분류 시스템

3.1 블로그 글 수집

국내의 대표적인 블로그 사이트인 네이버, 다음, 티스토리에서 수집된 블로그 문서를 기준으로 적합한 분류를 조사한 [4]에서 네이버 블로그는 장르 분류 일치도에서도 높은 결과를 보였다. 본 연구에서는 네이버의 장르, 즉 분류가 부착된 블로그를 이용하여 학습에 사용한다.

네이버 블로그에는 총 30개의 분류가 존재한다. 하지만, 각 분류들은 글을 수집하여 분석하고 학습 데이터로 쓸 만큼 충분한 양의 글이 올라오지 않는 경우가 있었다. 또한, 분류의 주제에 맞지 않는 광고성 글이 존재하는 경우도 존재했다. 이러한 문제를 보완하기 위해, 네이버의 30개의 분류 중 일부를 제거하거나 병합하여 아래 [그림 1]과 같은 16개 분류를 얻고, 이를 이용하여 자동 분류를 수행한다.

문학-책	육아-결혼	여행	요리-레시피/맛집
영화	건강-의학	스포츠	미술-디자인/공연-전시
음악	패션-미용	자동차	사회-정치
인테리어	IT-컴퓨터	게임	애완-반려동물

[그림 1] 블로그 분류를 위한 16개 분류

3.2 단어별 주제 분별 점수 계산

문서 분류를 위한 문서 특성은 제목과 본문에서 사용되는 명사에서 추출한다. 단어들을 정확히 추출하기 위해서 조사나 어미의 구분이 필요하다. 한국어 형태소 분석기를 이용하여 문서 내의 단어를 추출한 뒤, 각 단어의 빈도를 분석한다.

각 문서의 단어 점수는 문서에서 발생한 단어의 빈도 TF에 단어의 주제 분별 점수를 곱하여 얻는다. 이때 문서의 크기가 커지면 문서에서 발생한 단어의 절대 빈도도 커지므로 정규화가 필요하다. 문서 D에서 발생한 단어 w_i의 빈도수에 문서 D의 총 단어수로 나누어 정규화한 $TF_D(w_i)$를 식(1)로 구한다.

$$TF_D(w_i) = \frac{\text{문서 } D\text{에서 단어 } w_i\text{의 빈도}}{\text{문서 } D\text{의 총 단어 빈도}} \quad (1)$$

정규화한 단어 빈도 $TF_D(w_i)$와 아래에 설명할 IDF, CTF-IECDF, CDF-IDF, CDF-IECDF의 각 4가지 방식으로 구한 주제 분별 점수를 곱하여 문서 내 단어의 주제 분별력 점수를 생성한다. 아래에서는 각 주제별 점수 계산 방식을 설명한다.

3.2.1 IDF

IDF는 문서 빈도를 이용하여 단어의 희소성이나 정보성을 표현하는 통계적 방법으로, 첫 번째 방식에서는 IDF로 단어 w_i의 주제 분별 점수를 구한다. $IDF(w_i)$는 전체 문서수에서 단어 w_i가 발생한 문서의 빈도를 나눈 값에 log를 취한 값으로서 식으로 표현하면 다음 식 (2)과 같다.

$$IDF(w_i) = \log \frac{\text{전체문서의수}}{\text{단어 } w_i\text{가존재하는문서의수}} \quad (2)$$

3.2.2 CTF-IECDF

CTF-IECDF는 분류 c에서의 단어 누적 빈도와 c를 제외한 분류에서의 IDF값을 통해, 분류 c에서의 단어 중요도를 계산한다. CTF는 문서내 빈도인 TF를 분류 내 빈도로 확장시킨 값이며, IECDF는 특정 분류를 제외한 IDF를 의미한다. 이 방식에서는 단어 w_i의 누적빈도가 가장 큰 분류를 단어 w_i의 대표 분류로 보고, 해당당 분류 C_{max}를 수식 (3)으로 얻는다. C_{max}에서의 단어 w_i의 누적빈도 $CTF(w_i)$를 수식 (4)로 얻는다.

$$C_{\max} = \arg\max_{c}(c\text{에서의 단어}w_i\text{의 누적 빈도}) \quad (3)$$

$$CTF(w_i) = C_{\max}\text{에서 단어}w_i\text{의 누적빈도} \quad (4)$$

수식 (5)는 IECDF를 구한다. 수식에서 C_{others}는 전체 분류에서 C_{max}를 제외한 모든 분류를 의미한다. 분류 C_{others} 분류들에서의 단어 w_i의 IDF 값인 IECDF와 (4)의 CTF값을 곱한 후 루트를 취하여 단어 w_i의 주제 분별 점수를 구한다. CTF-IECDF의 값은 각각의 점수 차이가 크기 때문에 루트를 취하여 편차를 줄인다.

$$IECDF(w_i) = \log \frac{C_{others}\text{의 문서수}}{C_{others}\text{에서 } w_i\text{가 존재하는 문서수}} \quad (5)$$

$$CTF-IDCDF(w_i) = \sqrt{CTF(w_i) \times IECDF(w_i)} \quad (6)$$

3.2.3 CDF-IDF

CDF-IDF에서 CDF는 단어의 C_{max} 내 문서빈도를 의미한다. 수식(3)의 C_{max}를 이용하여, C_{max}에서의 문서빈도인 CDF_i를 수식 (7)으로 구한다.

$$CDF(w_i) = \frac{C_{\max}\text{에서 단어 } w_i\text{가 발생한 문서수}}{\text{카테고리 } C_{\max}\text{의 문서수}} \quad (7)$$

$CDF-IDF$에서는 식 (2)의 IDF와 식 (7)의 CDF를 곱하여 주제 분별 점수를 구한다. CDF를 통해 해당 단어가 대표 분류에서 폭넓게 사용될수록, IDF를 통해 해당 단어가 희소성이 높을수록 높은 점수를 얻는다.

$$CDF-IDF(w_i) = CDF(w_i) \times IDF(w_i) \quad (8)$$

3.2.4 CDF-IECDF

CDF-IECDF는 앞의 방식에서 사용한 CDF값과 IECDF 값을 곱하여 주제 분별력을 계산한다.

CDF를 통해 해당 단어가 대표 분류에서 폭넓게 사용될수록 높은 점수를 얻을 수 있고, 특정 분류를 제외한 IDF인 IECDF를 사용하여 나머지 분류에서의 희소성이 높을수록 높은 점수를 얻는다.

$$CDF-IECDF(w_i) = CDF(w_i) \times IECDF(w_i) \quad (9)$$

3.3 분류 모델 생성

문서별로 주제 분별 점수가 구해지면 이를 이용하여 분류 모델을 생성한다. 먼저, 학습용 문서 집합을 이용하여 분류 모델을 생성하고 평가한다. 분류 모델 생성을 위하여 기존의 소프트웨어 WEKA 3.6.10에 구현된 컴플리먼트 나이브 베이즈, 나이브 베이즈 멀티노미얼, SVM 알고리즘을 사용하였다.

각 분류 알고리즘을 이용해 입력을 각 문서 별 단어에 대한 주제 분별 단어점수로 하고 출력을 주제 필드로 설정하여 각각의 분류 모델을 생성한다. 분류 모델의 검증은 생성과 마찬가지로 검증용 문서 집합을 이용하여 생성된 각 분류 모델의 정확도를 검증한다.

4. 실험 및 평가

본 논문에서 제안하는 네 가지의 특성 추출 방식과 3가지의 분류 학습기를 통한 분류 정확률 실험 및 평가한

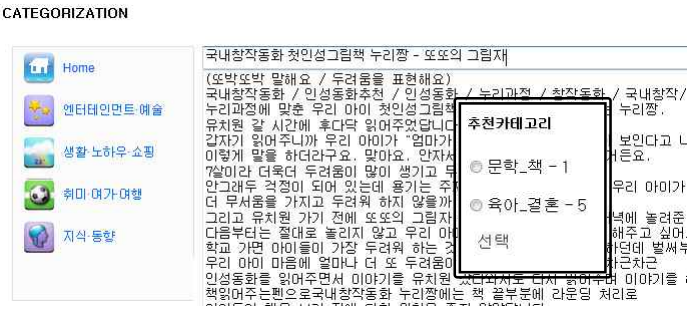
다. 학습 데이터와 테스트 데이터는 따로 수집하였다. 실험에서는 각 분류별 500개 총 8000개의 학습 데이터를 이용하였으며, 실험 데이터는 분류별 200개, 총 3200개를 사용한다. [표 1]은 결과를 보인다.

[표 1] 분류 학습기에 따른 분류 정확률

Test data 200	컴플리먼트 나이브 베이즈	나이브 베이즈 멀티노미얼	SVM
TF-IDF	44.06%	43.31%	6.25%
CTF-IECDF	73.26%	75.40%	69.06%
CDF-IDF	70.01%	58.64%	25.96%
CDF-IECDF	72.40%	72.61%	58.63%

기존의 TF-IDF는 3가지 분류 학습기를 이용한 결과 모두에서 50%가 되지 않는 분류 정확률을 보였다. TF-IDF는 분류 정보가 반영되지 않고, 키워드의 단순 빈도와 단순한 문서에서의 IDF를 이용하여 문서를 분류하여 낮은 정확률을 보이는 것으로 분석되었다. 컴플리먼트 나이브 베이즈로 학습한 CTF-IECDF는 TF-IDF와 29.2%의 차이로 더 높은 분류 정확률을 보였다. CDF-IDF와 CDF-IECDF는 약 2% 차이로 전체적으로 비슷한 정확률을 보였다. 나이브 베이즈 멀티노말로 학습을 한 결과에서 CTF-IECDF가 3가지 방식의 실험 중 75.40%로 가장 높은 정확률을 보였고, 반대로 CDF-IDF는 가장 낮은 결과를 보였다. 전체적인 실험 결과는 CTF-IECDF가 다른 방식들보다 가장 높은 정확률을 보였다. 그리고 SVM으로 학습을 한 결과는 다른 분류 학습기를 사용 했을 때 보다 비교적 낮은 정확률을 보였다.

본 연구에서는 실험 결과에서 정확률이 높은 상위 6개를 이용하여 블로그 자동 분류 시스템을 구축하였다. [그림 2]는 실행 예를 보인다. 시스템에서 사용자가 포스트를 작성한 뒤 저장 버튼을 누르면 자동으로 추천 카테고리를 제시하는 방법으로 구동된다. 시스템에서는 6개의 분류 기법에서 얻어진 분류를 투표 방식(voting)을 적용하여, 가장 많이 추천된 분류부터 순서대로 사용자에게 제시된다. 실행 예에서 포스트는 육아와 관련한 책을 소개하는 글로서, [문학-책]과 [육아_결혼]의 분류를 추천하는 결과를 볼 수 있다.



[그림 2] 블로그 포스트를 시스템에 입력한 결과

5. 결론

문서 분류에 TF-IDF를 변형하여 새로운 알고리즘을 만드는 연구는 다양하게 존재한다. 하지만 기사와 같이 의도가 분명한 글에 관한 분류와 달리 다양한 개성을 가진 사용자들의 글이 존재하는 블로그 문서에 대한 연구는 드물다.

본 논문에서는 블로그 포스트를 자동으로 분류하기 위해 TF와 IDF를 분류로 확장시킨 특성 추출 알고리즘을 사용하여 정확률을 실험하였다. 제안된 방식에서 나이브 베이즈를 사용한 분류 모델이 비교적 높은 정확률을 보였고, IDF를 사용한 것 보다 IECDF를 사용했을 때 블로그 문서의 분류가 더 정확하게 수행되었다. TF보다는 분류로 확장한 CDF나 CTF를 사용하는 것이 더 정확한 결과를 보였다.

블로그 문서는 정형적인 텍스트가 아니기 때문에 오타나 신조어 등에 민감 할 수 있는데 이러한 점은 형태소 분석기의 성능이 향상되거나 고유 명사 사전 등을 구축하면 해결 할 수 있을 것이라고 기대한다. 그리고 문서 자동 분류에서 문서 필터링으로의 확장이 추후 연구가 될 수 있다.

참고문헌

[1] Young-Ju Kim, "A Study on the Blog as a Media : Focused on Media Functions and the Problems of the Blog", Korean journal of journalism & communication studies, Vol.50, No.2, 2006.

[2] 박단호, 최원식, 김홍조, 이석룡, "한글 형태소 및 키워드 분석에 기반한 웹 문서 분류", 정보처리학회 논문지(D), pp.263-270, 2012.

[3] 강윤희, 박용범, "SVM을 이용한 디렉토리 기반 기술 정보 문서 자동 분류시스템 설계", 전기전자학회 Vol.8.No.2, pp. 186-194, 2004.

[4] Hae Young Kim, "An Experimental Study On Semi-Supervised Classification of Blog Genres", MS Thesis, Yonsei University, 2009.

코사인 유사도 기법을 이용한 뉴스 추천 시스템

김상모○, 김형준, 한인규
국민대학교 임베디드 시스템 연구실
simon@koama.net, dndnwjs@gmail.com, in1004kyu@gmail.com

SNS news Recommendation by Using Cosine Similarity

Simon Kim○, Hyung-Jun Kim, In-Kyu Han
Kookmin University, Embedded System Lab.

요 약

사용자별로 SNS/RSS 구독 뉴스 분석을 통해 사용자가 관심이 있는 새로운 뉴스를 추천해 주는 시스템을 설계하고 구현한다. 뉴스 추천 시스템의 설계를 위해 전체 시스템에서 사용자와 서버에서의 작업을 명세하고, 이중에 주요 기능을 담당하는 부분을 구현한다. 구현된 주요 기능은 선호 문서가 들어왔을 때 특징을 추출하고 이를 저장하는 것과 새로운 문서가 들어왔을 때 선호 문서군과 얼마나 유사한지 판별하여 문서에 대한 추천 여부를 결정하는 것이다. 선호 문서의 특징 추출에 대해서는 형태소 분석을 통해 단어와 빈도를 추출하고 이를 누적하여 저장한다. 또한, 새로운 문서가 들어왔을 때 코사인 유사도를 계산하여 사용자가 선호하는 학습문서와의 유사도 비교를 통해 문서 추천 여부를 결정한다. 구현된 시스템에서 실제로 연관된 선호 문서군을 학습시키고, 연관된 새로운 문서 혹은 연관되지 않은 새로운 문서에 대한 추천 여부를 비교하는 것으로 시스템 정확도를 파악한다.

주제어: Naïve Baysian, 코사인 유사도, 형태소 분석, 문서 벡터

1. 서론

오늘날 사용자들은 각종 매체에서 많은 뉴스를 접한다. 이중 원하는 뉴스를 골라보기 위해서 전체 뉴스들을 살펴 보아야 하는 번거로움이 있다. 이 때문에 자동적으로 사용자가 좋아할 만한 뉴스를 제공해 주는 서비스가 구글에서 제공됐다[1]. 하지만 이는 사용자가 어떤 뉴스를 클릭해서 들어갔는지에 대한 패턴 조사에 따른 추천 시스템이다. 단순히 뉴스를 본 것에 그치는 것이 아니라 사용자 자신이 Facebook의 "좋아요"를 클릭한 행동에 따라 추천해 준다면 더 정확한 뉴스의 추천이 가능할 것이다. 또한 현재 Facebook, Google Plus와 같은 소셜 네트워크를 통한 뉴스의 제공이 늘어나고 있는 추세이다. 그림 1에서 보는 것처럼 소셜 네트워크로부터의 뉴스 제공량이 2010에서 2012년까지 10%의 상승폭을 보이고 있다[2]. 이러한 이유로 인해 사용자 선호도 측정에 사용할 뉴스를 소셜 네트워크에서 제공되는 것을 기준으로 잡고 타 소셜 네트워크(Twitter)와 RSS에서 제공되는 뉴스에서 사용자가 좋아할만한 뉴스를 추천해서 보여주는 시스템을 개발하고자 한다.

여기서 해당 시스템의 사용자가 10만 명이라고 가정할 때 각 사용자의 평균 뉴스 구독 수를 5개, 사용자가 업로드하는 뉴스를 20개라고 가정하면, 하루에 생성되는 뉴스의 양은 약 1,000만개가 된다. 생성된 모든 뉴스 중 사용자의 관심을 끌만한 뉴스를 일일이 찾아본다는 것은 매우 번거러운 작업이다. 하지만 이미 선호하는 뉴스에 대한 정보는 SNS/RSS 구독 정보를 통해 얻을 수 있다. 따라서 우리는 각 사용자 별 SNS/RSS 구독 정보에 대한 분석을 통해 많은 양의 새로운 뉴스가 생성되었을 때 이 중 사용자가 관심을 가질 가능성이 높은 뉴스를 기존에 관심이 있었던 문서들과의 유사도를 기준으로 추천해 주는 시스템을 설계하고 구현하고자 한다.

More Americans Getting News Digitally and from Social Networks

Where did you get news yesterday?	2010 %	2012 %	Change
NET Online/Mobile*	34	39	+5
Online	34	34	0
Cell, tablet, other mobile	--	17	--
Social networking sites like Facebook, Google Plus	9	19	+10
Twitter	2	3	+1
Email	14	16	+2

그림 1. 2012 News Consumption Survey

2. 관련 연구

본 연구에서의 주요 기능들은 자연어처리 기법을 기반으로 구현되었다. 문서의 내용을 분석하고자 대상 어절의 모든 가능한 분석 결과를 출력하는 형태소 분석이나 문서 내용 분석을 기반으로 문서의 범주를 이미 정의된 범주에 할당하는 문서 분류 같은 것이 이에 해당된다. 형태소란 자연언어에서 '의미를 갖는 최소의 단위'이며 형태소 분석은 하나의 말마디를 여러 형태소로 분리한 후에 형태적 변형을 처리하는 것이다[3,4,5].

문서 분류는 인터넷 상의 블로그나 SNS 등 때문에 증가하는 문서의 양으로 인해. 효율적이며 정확한 문서 범주화를 요구하게 되며 발전하였다[6,7,8,9]. 문서 내용에 등장하는 용어에 기초한 통계적 방법, 문장의 의미를 파악하는 의미적 해석 방법, 문장 구조에 의존하는 구조적 방법으로 구별된다. 문헌에 나타난 용어의 특성 분석을 바탕으로 하는 자동 분류 시스템의 연구가 가장 활발하게 이루어지고 있다. 또한, 한글과 영어는 형태론적 구조가 다르므로 형태소 추출 과정은 다르지만 추출된 용어의 빈도 등의 특성 분석은 본질적으로 유사하다[10,11].

문서 분류는 문서 내용 분석을 통해 용어 집합을 통해 문서를 분류하는 방법과 분류 모델을 생성하여 문서를 분류하는 방법이 있으며, 또한 문장구조를 이용하는 방법도 존재한다[12,13]. 본 연구에서는 뉴스가 하나의 문서이기 때문에 사용자가 선호할 뉴스와 선호하지 않을 뉴스에 대한 분류 기술이 필요하다. 문서 분류 기법 중에서 나이브 베이지안 방법은 조건부 확률을 단순 확률곱 연산과 나눗셈 연산으로 간단화시키는 방법이다. 어떠한 문서가 있을 때 그 문서가 추천하는 클래스에 속해 있을 확률이 조건부 확률이다. 이 확률이 높을수록 해당 문서는 추천할 만하다고 판단한다. 이 확률을 쉽게 구하기 위해 Naïve Bayesian을 이용하면 독립적인 곱셈 연산과 나눗셈 연산으로 단순화시키는 벡터 연산이 가능하다.

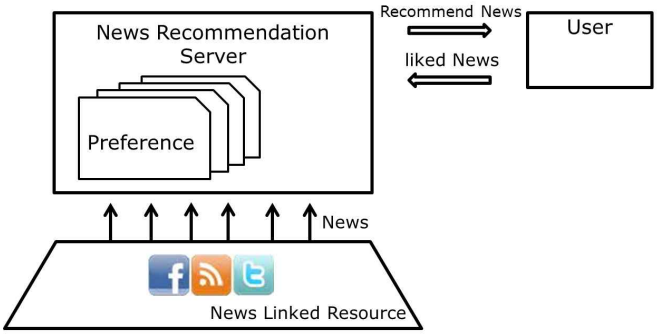


그림 2. 뉴스 추천 시스템의 서비스 구조

3. 뉴스 추천 시스템 구조

본 연구에서 설계한 시스템의 서비스 구조는 그림 2와 같다. 사용자는 단말에서 서버가 제공하는 2가지 서비스를 받는다. 추천된 뉴스를 확인하거나, 자신이 선호하는 뉴스를 선택할 수 있는 것이다. 그리고 서버는 개인별로 선호하는 뉴스에 대한 데이터베이스를 가지고, 새로운 뉴스를 SNS/RSS를 통해 얻고, 이 중 사용자가 관심을 가질 만한 뉴스를 추천해 주는 것이다.

중요 구성요소 별 시나리오는 크게 2가지로 나뉜다. 먼저 사용자가 선호 뉴스를 등록하면 선호 뉴스로 등록된 뉴스는 문서로 간주하고 해당 문서에 대한 특징 벡터를 추출한다. 추출된 특징 벡터는 개인 데이터베이스에 누적되어 저장된다. 매일 새로운 뉴스가 업로드되면 개인별 데이터베이스의 누적된 문서 특징 벡터를 이용하여 추천 여부를 판별한다. 새로운 뉴스의 내용 또한 특징 벡터를 추출하고 누적된 문서 특징 벡터와 유사도를 비교하여 추천 여부를 판단한다. 추천 여부에 따라 개인별로 관심을 가질 만한 문서가 추천된다.

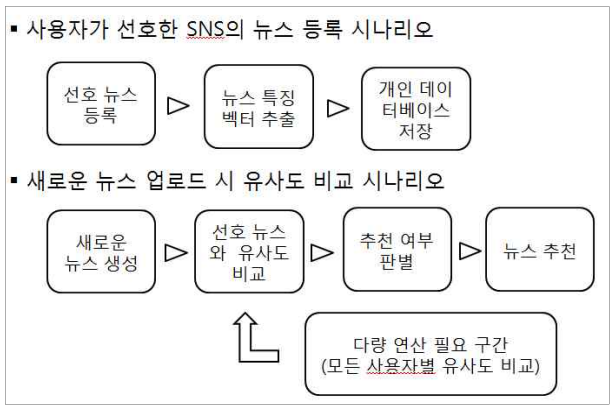


그림 3. 뉴스 추천 서비스 시나리오

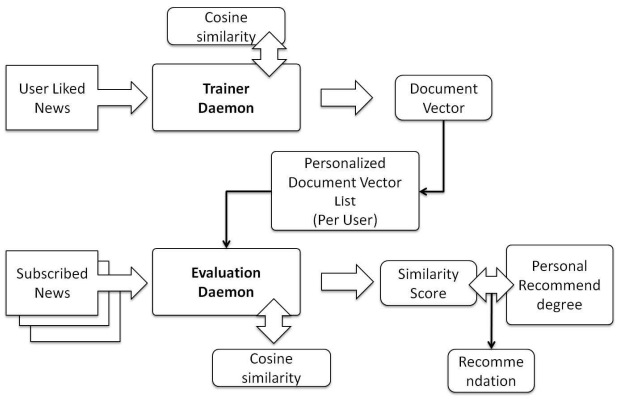


그림 4. 뉴스 추천 시스템의 구조

3.1 뉴스 추천 시스템의 구조

그림 4는 중요 구성 요소를 구현하기 위한 전체 시스템 구조를 나타낸다. 사용자가 선호한다고 표시한 뉴스를 User Liked News로 정의하고, RSS/SNS로부터 유입되는 뉴스를 Subscribed News라 정의한다. User Liked News를 입력 문서로 하는 Trainer Daemon 컴포넌트는 특징 벡터(한 문서에서 사용된 단어와 해당 단어의 개수의 매핑 벡터)인 Document Vector를 추출하여 Personalized Document Vector List(PDVL) 라는 데이터 저장소에 저장한다. 이 저장소는 각 개인 별로 존재하며 이를 기반으로 사용자 별 추천여부를 계산한다. 매일 새로 생성되는 Subscribed News와 PDVL을 기반으로 Evaluation Daemon 컴포넌트에서는 새로운 뉴스에 점수를 매긴다. 계산된 점수는 선호 뉴스들과의 유사도를 나타내는 Similarity Score로 정의된다. 이 점수와 사용자 별 유사도 정도를 나타내는 Personal Recommend degree와 비교하여 추천 여부를 결정한다.

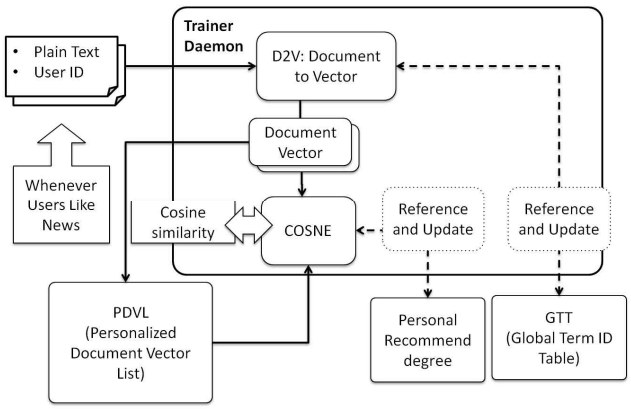


그림 5. 학습 데몬의 내부구조도

3.2 학습 데몬

학습 데몬(Trainer Daemon)은 사용자가 선호하는 문서에 대한 정보들을 수집하고 저장하는 모듈이다. 사용자가 선호하는 뉴스가 들어오면 이는 문서로 취급되며 D2V(Document to Vector)라 불리는 문서에서 특징 벡터를 추출하는 모듈에 입력된다. D2V 모듈은 문서 내부 단어들을 고유 ID값과 단어의 빈도수를 나타내는 COUNT값으로 이루어진 벡터인 Document Vector로 만들어서 PDVL에 저장한다. 모듈은 필수 기능 외에 고유 ID값을 다루는 모듈이 필요하다. 따라서 시스템 전체 단어를 관리하는 테이블이 필요하다. 이 테이블을 다루는 기능을 담당하는 모듈은 Global Term ID Table(GTT)로 정의하였다.

D2V는 문서를 벡터화하는 과정에서 GTT를 참조하며, 새로운 단어가 들어오면 GTT를 업데이트 하여 새로운 ID값을 제공받는다. Training을 하고 나서 현재 PDVL의 데이터와 선호 뉴스와의 유사도 계산 방식으로 뉴스 추천 기준값 Similarity Score를 계산한다. 이 기준값은 이후 새로 들어온 뉴스와의 유사도 계산 점수와 비교하는 기준값이 된다. 해당 점수는 PDVL과 사용자가 선호하는 뉴스와의 유사도 점수이기 때문에 이보다 높으면 추천 정도가 높다고 볼 수 있다. 뉴스 추천 기준값 Similarity Score는 개인별 데이터베이스에 저장한다. 만약 저장된 Score가 있다면 둘 사이의 평균값을 계산하여 새로 저장한다. 이 점수를 개인별 추천 정도를 나타내는 Personal Recommend degree라고 정의한다.

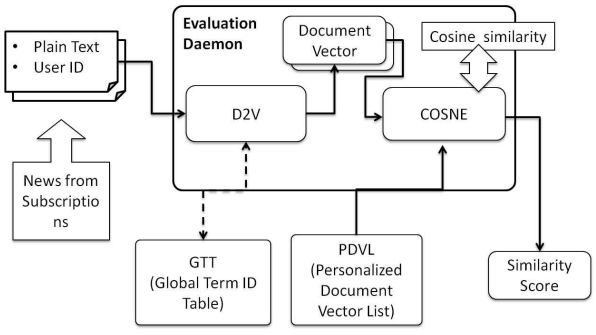


그림 6. 평가 데몬의 내부 구조도

3.3 뉴스 추천 및 평가

Evaluation Daemon은 새로 생성된 뉴스가 사용자가 관심을 가질 만한 뉴스인지 판별하는 모듈이다. 새로운 뉴스가 업데이트 되는 것을 감지하여, 문서를 가져오는 기능은 구현하지 않는다. 새로운 뉴스가 업데이트 되었다고 가정하고 다양한 문서들을 모듈의 입력 데이터로 사용한다. 입력된 데이터는 D2V를 이용하여 특징 벡터를 추출하고, PDVL과 함께 COSNE 모듈에서 코사인 유사도 연산을 통해 선호 뉴스들과의 유사도 정도를 도출해 낸다. 이 값은 Similarity Score로 정의되며 저장된 Personal Recommend degree와 비교하여 추천여부를 결정한다. Personal Recommend degree 보다 높으면 강력한 추천으로 추천하며, Recommend degree의 50% 보다 높은 상위 10개의 문서는 보통 추천으로 추천한다.

$$\mathbf{a} \cdot \mathbf{b} = \|\mathbf{a}\| \, \|\mathbf{b}\| \cos\theta$$

$$\text{similarity} = \cos(\theta) = \frac{A \cdot B}{\|A\|\|B\|} = \frac{\sum_{i=1}^{n} A_i \times B_i}{\sqrt{\sum_{i=1}^{n} (A_i)^2} \times \sqrt{\sum_{i=1}^{n} (B_i)^2}}$$

그림 7. 코사인 유사도 계산식

PDVL과 새로 들어온 뉴스의 Document Vector를 가지고 문서 분류 연산 중 하나인 코사인 유사도 알고리즘을 사용하여 추천 문서를 추천하는 모듈이다. 코사인 유사도는 그림 7과 같은 공식을사용한다. 두 백터 a, b 사이의 각도를 나타내는 $cos(\theta)$는 두 벡터 a와 b 사이의 각도를 나타내기 때문에 이를 유사도 정도로 볼 수 있다. 여기서 $cos(\theta)$ 그림-7 의 연산으로 구할 수 있다. 각 문서의 특징을 추출해 낸 D2V는 벡터로 표현이 가능하기 때문에 이와 같은 코사인 유사도 연산에 사용이 가능하다. 본 프로젝트에서 a는 비교할 문서의 벡터, b는 PDVL의 벡터가 되고 $cos(\theta)$ 는a와 b의 유사도 정도가 된다.

4. 실험 및 성능 평가

성능 측정을 위한 실험 환경은 표 1과 같다.

표 1. 실험 환경

CPU	Intel® Core™ i5-3570 CPU @ 3.40GHz
RAM	8GB DDR3 SDRAM
OS	Ubuntu 12.04

실험 데이터는 주로 IT 관련 뉴스 30개로 Training 시키고 임의의 800개의 뉴스에 대하여 추천 정도를 테스트한다. 실험 결과 학습에 사용한 30개의 뉴스와 추천 기준값이 되는 유사도보다 높은 6개의 뉴스를 선호도가 매우 높다고 판단되는 강력한 추천 대상으로 하였다. 추천 기준값의 50%보다 높은 뉴스는 약 300개 정도가 나왔고 그 중 상위 10개를 보통 추천 뉴스로 하였다. 그리고 나

머지 뉴스는 추천 대상에서 제외하였다. 추천 대상이 된 뉴스의 주제는 주로 삼성, LG, 구글과 같은 IT 업계와 관련된 뉴스였으며 추천 기준값의 50% 보다 높은 뉴스 중 상위 10개의 뉴스들 중 6개는 IT와 관련된 주제의 뉴스였지만, 나머지 4개는 IT 용어가 나오긴 하지만 경제나 사회에 관한 뉴스였다. 그 외 기준 유사도 점수의 50% 이상에 해당 하는 뉴스 중 상위 10개의 뉴스를 제외한 뉴스 들은 IT와 관련된 뉴스가 있긴 했지만 대부분 정치, 경제, 사회에 대한 뉴스였다. 결과적으로 유사도 점수가 높게 나온 뉴스는 대부분 사용자가 선호할 만한 뉴스였고, 낮게 나온 뉴스는 대부분 선호할 만한 뉴스가 아니였다.

표 2. 뉴스 추천 시스템의 실험 결과

추천 정도	강력 추천 (기준 유사도 이상)	보통 추천 (기준 유사도의 50% 이상 중 상위 10개)		유사도의 50 % 이상	비추천 (그 외)
추천된 뉴스	6개 (IT 회사, 제품)	6개 (IT 회사, 제품)	4개 (경제, 사회 - IT 관련정책)	300개 (IT, 기타)	나머지 기타

5. 결론

본 연구에서는 사용자가 좋아할 만한 뉴스를 가져오면 유사도 점수를 계산하여 새로 유입되는 뉴스와의 유사도 점수와 비교하여 추천하는 시스템을 구현하였다. 단순히 문서의 단어에 대한 빈도 계산만 가지고도 어느 정도 추천하는 시스템이 완성되었기 때문에 단어 빈도뿐만이 아닌 단어의 가중치를 고려한다면 더욱 정교한 추천 시스템이 구현될 것이다. 여기에 뉴스를 수집하는 모듈을 추가하여 실시간으로 추천되는 시스템으로 확장하면, 실제 서비스가 가능해지고 이를 웹페이지, 안드로이드, 아이폰 앱으로 확장이 가능할 것이다.

참고문헌

[1] Personalized News Recommendation Based On Click Behavior, http://googlenewsblog.blogspot.kr/2011/04/automatic-personalization-and.html

[2] Trend in News Consumption 1991-2012, http://www.people-press.org/

[3] 강승식, "다층 형태론과 한국어 형태소 분석 모델", 제6회 한글 및 한국어 정보처리 학술발표 논문집, pp.140-145, 1994.

[4] 강승식, 이하규, 손소현, 홍기채, 문병주, “조사 유형 및 복합명사 인식에 의한 용어 가중치 부여 기법”, 한국정보과학회 가을 학술발표논문집, 28권 2호, pp. 196-198, 2001.

[5] 최재혁, "형태소 분석을 통한 한영 자동 색인어 추출", 정보과학회논문지(B), 제23권, 제12호, pp. 1279-1288, 1996.

[6] Yang, Y. and Xin Liu, “A Re-examination of Text Categorization Methods”, In Proc. of Conference on Research and Development in Information Retrieval(SIGIR 99), pp.42-49, 1999.

[7] Y. Yang and J. P. Pedersen, “A Comparative Study on Feature Selection in Text Categorization,” In Jr. D H. Fisher(Ed.), 14th International Conference on Machine Learning, Morgan Kaufmann, pp.412-420, 1997.

[8] F. Sebastiani, “Machine Learning in Automated Text Categorization”, Technical Report IEI-B4-31-1999, Istituto di Elaborazione dell'Informazione, Consiglio Nazionale delle Ricerche, Pisa, IT, 1999.

[9] V. Vapnik, “The Nature of Statistical Learning Theory”, Springer-Verlag, 1995.

[10] 고영중, 박진우, 서정연, “문장 중요도를 이용한 자동 문서 범주화”, 정보과학회 논문지: 소프트웨어 및 응용, 29권 6호, pp.417-423, 2002.

[11] 이지행, 조성배, “전자우편 문서의 자동분류를 위한 다중 분류기 결합”, 정보과학회 논문지: 소프트웨어 및 응용, 29권 3호, pp.192-201, 2002.

[12] 이경찬, 강승식, “용어 가중치와 역범주 빈도에 의한 자동 문서 범주화”, 제15회 한글 및 한국어 정보처리 학술발표 논문집, pp.14-17, 2003.

[13] K. C. Lee, S. S. Kang, K. S. Hahn, “A Term Weighting Approach for Text Categorization”, AIRS 2005, pp.673-678, 2005.

어휘지도(UWordMap)를 이용한 용언의 다의어 중의성 해소

배영준○, 옥철영
울산대학교 컴퓨터정보통신공학
young4862@nate.com, okcy@ulsan.ac.kr

Word Sense Disambiguation of Polysemy Predicates using UWordMap

Young-Jun Bae○, Cheol-Young Ock
Dept. of Computer Engineering & Information Technology, Ulsan University

요 약

한국어 어휘의 의미를 파악하기 위하여 어휘의 의미 중의성을 해결하는 것은 중요한 일이다. 본 논문에서는 한국어 다의어 기반의 어휘 의미망과 용언의 논항정보 등의 관계가 포함된 어휘지도(UWordMap)를 사용하여 용언의 의미 중의성 해소에 대한 연구를 진행한다. 기존의 의미 중의성 해소 연구와 같은 동형이의어 단위가 아닌 다의어 단위의 용언 의미 중의성 해소 시스템을 개발하였다.

실험결과 실험말뭉치로 품사 태그 부착 말뭉치를 사용했을 때 동형이의어 단위 정확률은 96.44%였고, 다의어 단위 정확률은 67.65%였다. 실험말뭉치로 동형이의어 태그 부착 말뭉치를 사용했을 때 다의어 단위 정확률은 77.22%로 전자의 실험보다 약 10%의 높은 정확률을 보였다.

주제어: 용언 의미 중의성 해소, 다의어, 어휘지도, UWordMap

1. 서론

지식 정보화 사회에서 고도화된 맞춤형 정보와 지식을 추출하기 위해 다양한 종류의 데이터·정보를 정확히 분석하기 위한 연구가 진행되고 있다. 특히 데이터 중 텍스트로 된 언어자원과 관련된 연구에서는 단어의 의미 또는 문맥의 의미를 분석하기 위한 연구가 활발히 진행되고 있다. 그러나 특정 단어가 가지는 의미의 수가 다양할수록 그리고 이러한 단어들의 빈도가 높은 문서일수록 의미 분석 복잡도가 상승하기 때문에 정확한 의미 분석은 어려워지며, 이러한 의미 분석을 통해 제공되는 정보는 신뢰성이 떨어지는 경향이 있다. 예를 들면 정보검색에서 사용자가 '맛있는 배'를 검색어로 입력했을 경우 사람은 '과일 배'로 인지하여 의미에 맞는 문서를 결과로 제시할 수 있지만, 컴퓨터는 의미를 구분하지 못해 단순히 일치되는 단어 '배'{과일, 신체부위, 기계, …}가 포함된 모든 문서를 검색결과로 제시한다. 그러나 이를 적절히 분류하여 '과일 배'가 포함된 문서만을 결과로 제시한다면 보다 신뢰성이 높은 시스템이 될 수 있다. 이러한 문제를 해결하기 위해 형태적 처리부터 구문·의미적 처리까지 다양한 방법이 연구되고 있다. 하지만 형태적 처리 기술이 95%이상의 정확률을 보이는 반면, 의미적 처리 기술은 의미적 중의성을 가지는 특정 소수 단어만을 대상으로 처리하여도 70~90%의 정확률을 보이는 등 현재까지 많은 한계를 가지고 있다.

의미 중의성 해소를 위한 의미 분석 단위는 동형이의어(同形異義語)와 다의어(多義語) 두 단위로 구분할 수 있다. 동형이의어는 형태는 동일하나 의미들이 전혀 다른 말을 뜻한다. 예를 들면 '배_01(과일)', '배_02(기계, 운송수단)', '배_03(신체부위)'와 같다. 다의어는 하나의 말이 둘 이상의 다르면서 어원적(語源的)으로 관련 있는 의미를 가지고 있는 말을 가리킨다. 즉, '배_03(신체부위)' 중에서도 "사람이나 동물의 몸에서 위장, 창자, 콩팥 따위의 내장이 들어 있는 곳으로 가슴과 엉덩이 사이의 부위."라는 의미의 '배'가 있으며, "아이가 드는 여성의 태내(胎內)."와 같은 의미의 '배'가 있다. 이 둘은 포괄적 의미로 신체부위를 나타내지만, 세부적 의미는 다른데 이를 다의어라 한다.

<표 1>을 동형이의어와 다의어 단위로 구분하면 2개의 동형이의어{다리[1], 다리[2]}와 4개의 다의어{다리[1]①, 다리[1]②, 다리[2]①, 다리[2]②}로 구분할 수 있다.

표 1. '다리'의 뜻풀이

단어	뜻풀이
다리[1]	①사람이나 동물의 몸통 아래 붙어 있는 신체의 부분.
	②물체의 아래쪽에 붙어서 그 물체를 받치거나 직접 땅에 닿지 아니하게 하거나 높이 있도록 버티어 놓은 부분.
다리[2]	①물을 건너거나 또는 한편의 높은 곳에서 다른 편의 높은 곳으로 건너다닐 수 잇도록 만든 시설물.
	②두 사물이나 사람 사이를 이어 주는 역할을 하는 것.

다의어 단위의 의미 중의성 해소는 동형이의어 단위의 의미 중의성 해소보다 세부적인 의미로 분류해야하기 때문에 분석이 더욱 어렵다. 그렇지만 다의어 단위로 정확히 의미 분석을 할 수 있다면 더욱 신뢰성 높은 정보를 제공할 수 있다. 예를 들면 "**다리**를 못 쓰는 장애인들을 위해 주부들이 운전봉사자로 나섰다."라는 문장에서 동형이의어 '다리[1]'{①사람의 다리, ②물건의 다리}를 의미 중의성 해소 결과로 제시하면 포괄적인 의미는 이해할 수 있지만, 다의어 '다리[1]①'{사람의 다리}를

결과로 제시하면 정확한 의미를 이해할 수 있다.

현재 대부분의 의미 중의성 해소 연구에서 의미 분석 단위는 동형이의어 단위이다. 그러나 부분적으로 다의어 단위를 포함하는 말뭉치인 SENSEVAL-2를 실험말뭉치로 사용하는 연구[1, 2]도 있었지만, SENSEVAL-2는 실험말뭉치에 출현하는 빈도가 1인 다의어를 제외하면 전체 10개의 어휘 중 어휘 '점' 하나만 다의어 처리 대상이 된다. 그러므로 SENSEVAL-2는 다의어 단위라기 보단 동형이의어 단위의 말뭉치로 볼 수 있다.

신뢰성 높은 정보를 획득 및 제공하기 위해서는 다의어 단위의 의미 중의성 해소가 필요하다. 그렇지만 현재 다의어 단위의 자원이 부족하기 때문에 자원의 구축이 필요하며, 구축된 자원을 바탕으로 다의어 기반의 의미 중의성 해소 방법에 대한 연구가 필요하다.

2. 관련 연구

단어의 의미 중의성 해소를 위한 많은 연구들이 있었다. 크게 두 부류의 연구로 나눌 수 있는데, 대용량 말뭉치를 바탕으로 베이지안 분류기, 결정 트리, 신경망, CRF, SVM등의 기계학습 기법의 통계정보에 기반을 둔 연구와 기계가독형 사전, 시소러스, 온톨로지, 의미망, 연어 등의 정보를 이용한 지식정보에 기반을 둔 연구로 나눌 수 있다.

그러나 기존의 의미 중의성 해소와 관련된 논문의 대부분이 동형이의어 중의성 해결을 위한 논문이었으며, 학습된 말뭉치를 기반으로 한 의미 중의성 해소를 시도하는 논문이었다[1,2,3,4].

용언의 의미 중의성 해소에 관한 논문 중 종속격 정보를 적용한 동사 의미 중의성 해소와 관련된 연구가 있었다. 보다 정확한 의미 중의성 해소를 위해 공기 빈도와 더불어 종속격(목적격, 부사격, 보격) 정보를 활용하였다[4]. 이 논문은 12개의 동사를 대상으로 98.7%의 정확률을 보였지만, 동형이의어 단위의 의미 중의성 해소 연구였다.

한국어 동사의 의미 중의성 해소에 관한 논문으로 문맥에서 추출한 가중치 정보를 이용한 모델을 제안하는 연구가 있었다. 약 300만 어절의 사전에서 추출한 공기관계의 문맥에서 얻은 품사정보와 거리정보를 사용하였다. 논문에서는 다의어 수준의 동사(감다, 피우다, 빠지다, 타다) 4개를 대상으로 실험하였으며, 실험결과 84%의 내부실험 정확률과 75%의 외부실험 정확률을 보였다[5]. 이 논문의 결과를 이후 비교 실험 결과로 사용한다.

본 논문에서는 다의어 단위의 의미 중의성 해소를 위해 대용량 말뭉치 대신 다의어 단위로 구축된 한국어 어휘지도 UWordMap을 활용하여 용언의 의미 분석에 대한 연구를 진행한다.

3. 한국어 어휘지도 UWordMap

한국어 어휘지도인 UWordMap은 표준국어대사전을 기반으로 한국어 어휘의 다의적 수준의 통사-의미 관계를 네트워크로 연결한 어휘지식베이스이다. UWordMap은 U-WIN[1]을 기반으로 구축되었다. U-WIN은 품사 간의 관계, 주로 같은 품사인 명사들 간의 다양한 관계에 중심이 맞춰져 구축되었다. UWordMap은 이러한 U-WIN의 관계를 기반으로 일부만 존재하던 다른 품사 간의 연결 정보에 의한 구문관계와 의미관계를 보다 확장하여 구축되었다.

UWordMap은 구문관계에 의한 용언과 명사(주격·목적격·부사격) 등의 하위범주화 정보 뿐 아니라, 용언과 부사의 관계, 부사와 부사의 관계와 의미관계에 의한 용언과 관련된 명사의 의미역 등이 확장 구축되고 있다. 구문 관계의 주요 대상이 되는 용언 중 말뭉치에 출현빈도가 높은 용언은 대부분 구축되었고 출현 빈도가 낮은 용언은 구축 중에 있다.

UWordMap의 구문정보는 용언과 명사어휘망과의 매핑을 통해 구축되었다. 명사어휘망의 명사를 선정할 때는 최소공통상위노드(LCS:least common subsumer)[6]를 선택하였다. 만약 최소공통상위노드의 하위노드 중 용언과 연결될 수 없는 어휘가 포함되어 있을 시에는 이 어휘를 N관계에 포함시켜 해당 용언과의 관계가 연결되지 않도록 설정하였다.

4. 다의어 단위의 용언 의미 중의성 해소

품사 태그가 부착된 말뭉치만을 기반으로 다의어 단위의 의미 중의성 해소를 시도하는 것은 많은 어려움이 있다. 대부분의 명사류 또는 용언류의 단어들은 의미적 중의성을 가지고 있으며, 이런 단어들의 관계를 파악하여 의미를 결정하는 것은 복잡도가 상당히 높은 작업이기 때문이다. 만약 공개된 다의어 기반의 의미 태그 부착 말뭉치가 있다면, 지도학습 기반의 기계학습방법을 사용하여 어느 정도의 성능을 보장할 수 있는 시스템 개발이 가능하지만, 공개된 다의어 기반의 말뭉치가 없기 때문에 기계학습에 의한 지도학습방법은 사용할 수 없다. 그래서 본 논문에서는 다의어 기반으로 구축된 UWordMap을 활용하여 용언의 다의어 분석을 진행하였다.

다의어 분석을 위한 방법으로 두 가지 말뭉치(품사 태그 부착 말뭉치, 품사/동형이의어 태그 부착 말뭉치)를 대상으로 실험을 진행하였다. UTagger[2]는 형태소 분석뿐 아니라 동형이의어 분석까지 가능하기 때문에 동형이의어 분석의 유무에 따른 다의어 분석 정확률의 차이를 알아보기 위해서 두 가지 말뭉치로 실험을 진행하였다.

실험을 위한 과정은 [그림 1]과 같다. 문장이 입력되면 형태소분석과 동형이의어분석을 진행한 후 동사(VV)와 형용사(VA)를 찾고 그 앞의 논항을 검색한다. 용언 바로 앞의 조사 중 주격조사(JKS), 목적격조사(JKO), 부

1) U-WIN(User Word Intelligent Network)은 한국어의 공통적이고 개별적인 속성을 바탕으로 한국인의 보편적인 인지 체계와 개념 관계를 파악하여 이를 어휘의 의미적·개념적 네트워크를 형성한 온톨로지적 의미망이다. 핵심 구축 대상은 명사, 동사, 형용사이며, 나머지 품사는 부수적인 구축 대상이다. 2013년 현재 48만여 명사에 대해 상·하위 관계가 구축되어 있다.

2) UTagger는 울산대학교 한국어처리연구실에서 개발한 품사 및 동형이의어 태깅시스템이다.

사격조사(JKB)가 검색되면 그 앞의 명사를 해당 논항 명사로 용언과의 쌍을 저장한다. 논항의 수를 보다 많이 확보하기 위해 복수를 뜻하는 접미사 '들/XSN' 이 나타날 경우 그 앞의 명사를 용언과 쌍으로 만들어 저장한다. 저장된 쌍을 UWordMap과의 비교를 통해 다의어를 분석한다. 실제 UWordMap은 다의어 단위(밥_010001, 밥_010002, 밥_020000…)로 구축되어 있고, 입력된 단어는 원형(밥) 및 동형이의어(밥_01) 단위이기 때문에 원형 또는 동형이의어와 일치하는 모든 다의어를 UWordMap에서 찾아서 해당 용언과 관계를 확인한 후 용언의 다의어를 선택한다.

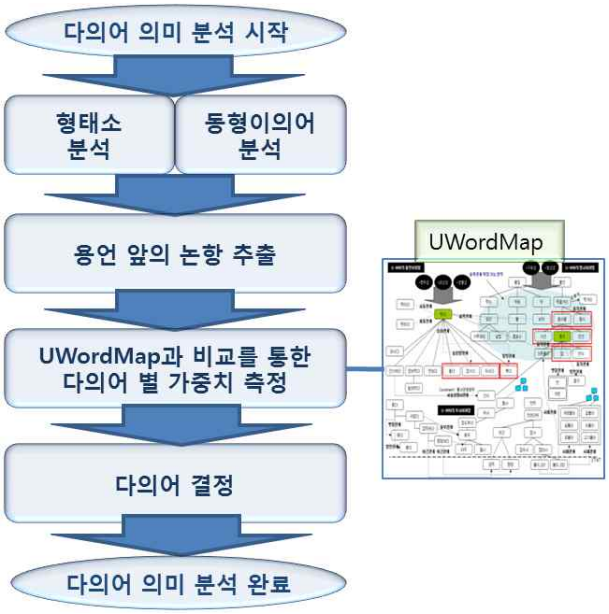


그림 1. 다의어 의미 분석 과정

UWordMap의 하위범주 정보는 최소상위공통노드로 설정되어 있기 때문에 직접적으로 연결된 관계가 찾아지지 않을 수 있다. 그래서 논항 명사의 명사어휘망 내의 위치 확인 후 상위탐색을 통한 확장 검색을 실행한다. [그림 2]를 보면 동사 '먹다_020101' 의 '을' 논항에 대해 5개의 최소상위공통노드가 설정되어있다. 예를 들어 실험 데이터로 "영수가 밥을 먹다." 라는 문장이 입력되었을 때 '밥' 이라는 단어는 직접적으로 연결되어 있지 않지만 '밥' 의 명사어휘망 내의 위치에서 상위탐색을 통해 논항 관계가 설정되어 있는 '음식물' 을 발견하면 이를 근거로 하여 실험 데이터의 '먹다' 를 다의어 '먹다_020101' 로 설정할 수 있다.

5. 실험 및 평가

실험에 사용된 용언은 표준국어대사전의 용언 중 옛말·북한말·방언을 제외한 다의어 52,224개 중 UWordMap에 포함된(말뭉치 상에 고빈도로 출현) 15,514개의 다의어 용언을 대상으로 실험하였다.

본 논문에서는 UWordMap을 이용한 다의어 분석의 가능성을 판단하기 위한 실험이기 때문에 재현율을 고려하지 않고 실험을 진행하였다. 용언이 문장의 첫 어절로 나타

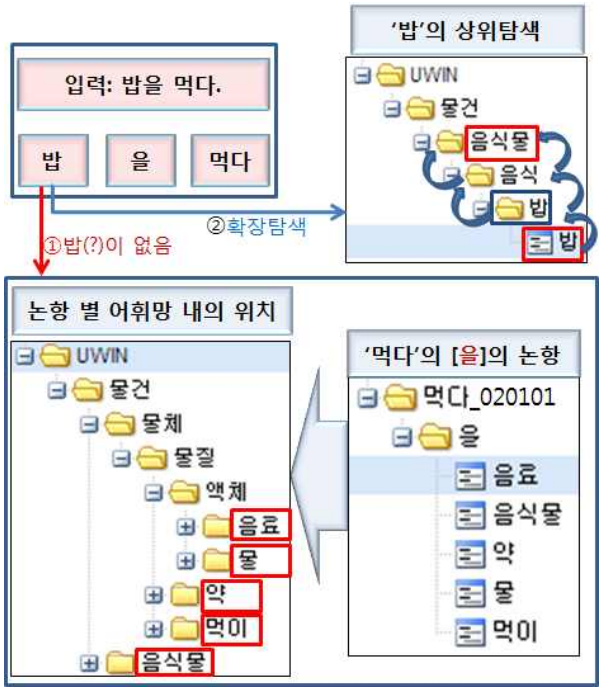


그림 2. 용언의 다의어 분석 예(밥을 먹다)

나는 경우 앞의 논항이 없어 정보를 추출할 수 없기 때문에 분석 대상에서 제외하였고, 용언의 바로 앞의 단어가 조사 및 부사가 아닌 경우도 분석 대상에서 제외하였다.

실험 및 정답 말뭉치는 표준국어대사전의 용례 216,089개의 문장을 대상으로 하였다. 실험 말뭉치는 품사 태그 부착 말뭉치와 UTagger를 이용한 품사/동형이의어 태그 부착 말뭉치를 대상으로 하였고, 정답 말뭉치는 다의어 의미 태그가 부착된 용례를 대상으로 하였다. 정답 말뭉치는 용례에 다의어 의미 태그를 반자동으로 부착한 뒤 전문가가 확인 후 수정하는 방식으로 구축되었다.

실험말뭉치에 따라 품사 태그만 부착된 실험말뭉치를 사용하는 실험과 품사/동형이의어 태그가 부착된 실험말뭉치를 사용하는 실험으로 나누었다. 전자에서는 다의어 및 동형이의어 두 단위의 정확률을 측정하였고(<표2>의 A와 B), 후자에서는 이미 동형이의어 태그가 부착된 상태이기 때문에 다의어 단위의 정확률만 측정하였다(<표2>의 C). 실험 결과 동형이의어 단위의 분석(<표2>의 B)은 UTagger의 정확률과 비슷한 성능을 보였다. 다의어 단위의 분석에서 품사 태그와 동형이의어 태그까지 포함된 실험결과가 품사 태그만 사용한 실험 결과보다 9.57% 높은 성능을 보였다. 이러한 결과가 나타나는 원인은 다의어 분석 전에 동형이의어 범위를 제약함으로써 결과로 나타날 수 있는 다의어의 경우의 수를 줄이는 효과가 있기 때문이다. 하지만 약 10% 정도의 차이를 보인다는 것은 동형이의어 분석 때문에 무거워 보일 수 있는 시스템인 UTagger를 충분히 사용할 가치가 있다고 판단하는 근거가 될 수 있다. 그리고 다의어 분석 시 원시 문장에서 바로 다의어 분석을 시도하는 것이 아니라 그 전 단계로 동형이의어 분석 단계를 포함하는 것도 고려해볼 가치가

있음을 의미한다.

표 2. 다의어 분석 정확률

[A]:품사 태그 부착 말뭉치의 용언 다의어 분석 정확률, [B]:품사 태그 부착 말뭉치의 용어 동형이의어 분석 정확률, [C]:품사/동형이의어 태그 부착 말뭉치의 용언 다의어 분석 정확률

A	B	C
67.65	96.44	77.22

실험 결과 나타난 오류 중 대부분이 다의어 중의성에 의한 오류였다. 용언뿐만 아니라 추출된 논항도 다의어 의미 중의성을 가지기 때문이다. “예를 가르치다.”와 같은 문장의 경우 ‘예’의 다의어는 27개나 되기 때문에 다의어 ‘예’가 많이 포함된 논항의 용언으로 의미가 결정된다. 이러한 오류를 줄이기 위해 문장에서 용언 앞의 논항탐색 범위를 늘리거나, 다른 품사들과의 공기정보와 같은 통계정보를 활용하면 성능향상에 도움이 될 것이다. 그리고 접속조사 ‘와/과’에 의한 논항의 확장을 통해 논항 간의 최소상위공통노드를 설정하여 다의어 의미 중의성 해소에 활용할 수 있을 것이다.

다음으로 많이 나타난 오류는 용언 다의어 가중치 값이 동일하여, 적절한 의미로 결정을 하지 못하는 경우였다. 이러한 오류도 위에서 설명한 방법을 통해 일부 해결이 가능할 것이다.

실험말뭉치와 동사의 의미 수가 다르기 때문에 정확한 비교가 되지 않지만, 의미 태그 부착 말뭉치를 사용하지 않고 실험을 진행한 임수종(1998)의 연구와 비교를 시도하였다. 비교결과를 살펴보면 <표 3>과 같이 본 논문이 중의성 해소가 필요한 용언의 의미 수가 전체적으로 23개 더 많았지만 정확률에서 약 2%정도의 차이만 보였다. 구분해야하는 의미 수가 배로 더 많지만 전체 분석 성능은 비슷하였다. 임수종(1998)에서는 용언의 의미의 수가 증가할수록 낮은 정확률을 보였지만, 본 논문에서는 동사 ‘빠지다’의 경우만 낮은 정확률을 보였다. 원인을 분석해 보면 동사 ‘빠지다’의 경우 의미 중의성이 다양한 한 음절의 명사와 논항으로 빈번히 결합하는 것을 볼 수 있었다. 예를 들면 {살(다의어 개수:20), 힘(17), 발(27),…}과 같은 명사가 자주 논항으로 나타났다. 반면 임수종(1998)에서는 문맥 상 나타나는 주위의 명사도 같이 고려하기 때문에 동사 ‘빠지다’에 대해서 본 논문의 결과보다 높은 정확률이 측정된 것이라 판단된다.

표 3. 다의어 중의성 해소 실험 결과 비교

	임수종(1998)		본 논문	
동사	의미 수	정확률	의미 수	정확률
감다	3	90.6	7	88.34
피우다	4	88.0	6	85.17
빠지다	11	66.0	17	41.03
타다	13	58.6	24	80.5
전체	31	75.8	54	73.76

6. 결론

본 논문에서는 말뭉치에 일정 빈도 이상 사용되는 용언을 대상으로 의미 중의성 해소를 시도하였다. 포괄적 의미를 분석하는 동형이의어 기반이 아닌 세부적 의미를 분석하는 다의어 기반의 의미 중의성 해소 연구였다. 적절한 학습말뭉치가 없기 때문에 다의어 기반으로 구축된 UWordMap을 활용하여 용언과 격의존관계에 있는 논항 명사의 연결 관계를 바탕으로 의미 중의성 해소를 진행하였다.

실험결과 품사 태그 부착 말뭉치를 사용했을 때 동형이의어 단위 정확률은 96.44%였고, 다의어 단위 정확률은 67.65%였다. 동형이의어 태그 부착 말뭉치를 사용했을 때 다의어 단위 정확률은 77.22%로 전자의 실험보다 대략 10% 높은 결과가 나타났다. 이는 UTagger와 같은 형태소뿐만 아니라 동형이의어 단위까지 분석하는 시스템을 사용하는 것이 다의어 분석에 보다 높은 성능을 보일 수 있다는 것을 보여준다.

대용량 말뭉치를 사용하지 않고 어휘지도만의 사용으로도 대용량 말뭉치를 사용한 것과 유사한 분석 결과 도출이 가능하다는 것을 실험을 통해 알 수 있었다. 다의의 분석의 초기단계인 본 논문의 실험결과를 살펴봤을 때 UWordMap은 다의어 의미 중의성 해소에 유용한 자원이라 판단된다.

향후 보다 적절한 논항정보를 획득하기 위해 다양한 규칙 기반의 의존구문파서를 적용하여 UWordMap을 통한 다의어 분석을 진행할 것이다. 그리고 용언뿐만 아니라 명사도 분석 대상에 포함하여 통계정보, 공기정보 등을 통한 다의어 분석을 진행할 것이며, 중의성을 가지는 논항 간의 최적의 최소상계노드를 추출할 수 있는 방법 등에 대한 연구를 진행할 것이다.

참고문헌

[1] 김민호, 권혁철, "한국어 어휘의미망의 의미 관계를 이용한 어의 중의성 해소", 정보과학회논문지,제38권, 제10호, pp.554-564, 2011

[2] 이호, 백대호, 임해창, "분류 정보를 이용한 단어 의미 중의성 해결", 한국정보과학회, 정보과학회논문지(B), 제24권, 제7호, pp.779-789, 1997

[3] 허정, 옥철영, "사전의 뜻풀이말에서 추출한 의미정보에 기반한 동형이의어 중의성 해결 시스템", 한국정보과학회논문지, 제28권, 제9호, pp.688-698, 2001

[4] 박요셉, 신준철, 옥철영, 박혁로, "종속격 정보를 적용한 동사 의미 중의성 해소", 정보처리학회논문지 Part(B), 제18권, 제4호, pp.241-248, 2011

[5] 임수종, 박영자, 송만석, "가중치 정보를 이용한 한국어 동사의 의미 중의성 해소", 한국정보과학회 언어공학연구회 학술발표 논문집, pp.425-429, 1998

[6] Resnik, P., "Using information content to evaluate semantic similarity in a taxonomy", Proceedings of the 14th International Joint Conference on Artificial Inteliigence, Montreal, pp.448-453, 1995

모바일 기기에서 일정 관리를 위한 개체명 인식

장은서○, 강승식, 이재원†, 김도현††
국민대학교, 성신여자대학교†, 삼성전자††
{akdangz, sskang}@gmail.com, jwlee@sungshin.ac.kr, dh1022.kim@samsung.com

Named Entity Recognition for Schedule Management in Mobile Devices

Eun-Seo Jang, Seung-Shik Kang, Jae-Won Lee, Do-Hyun Kim
Kookmin University, Sungshin University, Samsung Electronics

요 약

본 논문은 모바일 기기에서 일정을 메모하거나 음성 인식 등의 인터페이스로부터 일정 관리, 약속과 관련된 문구가 입력되었을 때 입력 문자열로부터 개체명을 인식하여 시간, 장소, 참석자 등을 일정 관리 시스템에 자동으로 등록하는 개체명 인식 시스템을 개발하는 방법에 관한 연구이다. 일정 관리의 편의성을 위한 개체명 인식 시스템을 개발하기 위하여 개체명 사전을 구축하고, 자연어 처리 기술을 이용하여 정확하고 향후 발전 가능성이 높은 시스템을 개발하고자 한다.

주제어: 일정 관리, 개체명 인식, 모바일 기기, 인명, 지명, 기관명, 자연어처리

1. 서론

스마트폰, 타블렛 PC 등 스마트 기기가 널리 보급되고 활용되면서 전 세계의 사람들의 삶의 방식에 변화를 가져왔다. 한편, 사용자들이 점점 스마트 기기를 다루는데 익숙해짐과 동시에 스마트 기기의 터치 스크린과 같이 직접 입력하는 방식이 아닌 음성 인식이나 필기 인식과 같은 내장 센서와 관련 기술들을 이용하여 기기를 다루는 인터페이스에 대한 연구 개발의 수요 또한 높아지고 있다. 스마트 기기의 기능 중 활용도가 높은 일정 관리 시스템과 자연어 처리 인터페이스, 대화형 인터페이스 등을 결합하여 일정 관리라는 정형화된 도메인 안에서 자연어 처리 기술을 적용함으로써 사용자의 편의성을 강화할 수 있다.

개체명 인식 기술은 정보검색 분야에서 대용량의 정보 자료로부터 관심의 대상이 되는 특정 유형의 정보만을 추출하기 위한 목적으로 연구되어 왔으며 대표적인 개체명으로는 인명, 지명, 기관명이다. 일반적인 개체명 인식은 “누가, 언제, 어디서, 무엇을, 어떻게, 왜”와 관련된 5W1H에 해당하는 개체명을 인식하고자 하는 것이다. 그런데 이 중에서 명확하게 개체명으로 인식이 가능한 것을 중심으로 문장에 출현한 어휘들을 개체명 단위로 구분하여 태깅을 하는, 기계적으로 개체명을 인식하는 것이 문서 자료에서 주요 관심이 되는 정보를 추출하는 목적을 달성하기 위한 가장 기초적인 작업인 것이다.

날짜와 시간도 중요한 개체명에 속하는데 영어의 경우에는 날짜와 시간을 인식하는 것은 정규식으로 쉽게 인식되기 때문에 관심의 대상에서 제외되었다. 그러나 한국어의 날짜, 시간 표현은 아라비아 숫자와 한글이 혼용되기 때문에 매우 다양한 표현이 가능하여 날짜와 시간과 관련된 개체명을 인식하는 것도 개체명 인식의 주요 기능에 포함되어야 한다.

본 논문은 모바일 기기에서 일정을 메모하거나 음성 인식 등의 인터페이스로부터 일정 관리, 약속과 관련된 문구가 입력되었을 때 입력 문자열로부터 개체명을 인식하여 시간, 장소, 참석자 등을 일정 관리 시스템에 자동으로 등록하는 개체명 인식 시스템을 개발하는 방법에 관한 것이다. 즉, 일정 관리의 편의성을 위한 개체명 인식 시스템을 개발하기 위하여 개체명 사전을 구축하고, 자연어 처리 기술을 이용하여 정확하고 향후 발전 가능성이 높은 시스템을 개발하고자 한다.

2. 개체명 분류

2.1 일정관리 문맥 패턴

스마트 기기는 매우 개인화된 단말기이다. 하나의 단말기는 여러 명의 사용자가 공용으로 사용하기 보다는 한 명의 사용자가 주로 사용하는 특성을 가지며, 이러한 특성을 이용하여 개체명 인식 시스템의 정확도를 한 단계 향상시킬 수 있다. 사용자의 습관, 사회적인 관습 등 사용자의 메모 패턴을 분석하여 개체명 인식과 사전 구축에 활용한다. 또한 개체명 인식이 완료되었을 때 각 개체명들의 배치 순서들을 저장하여 추후 분석에 활용하는 사용자 피드백에 의한 개인화된 학습 방법론을 적용한다. 표 1은 사용자의 습관 또는 사회적 관습과 같이 일정한 패턴을 분석하여 개체명 인식에 적용하는 예시를 나타낸다.

표 1. 일정관리 관련 문구의 특성과 개체명 인식

분류	내용
사용자의 습관	<시간><장소명><이벤트명> 으로 메모한 사용자는 추후 같은 순서로 메모할 가능성 높음
사회적 관습	- 일반적으로 약속은 <시간> <장소명> 패턴으로 메모하는 경향이 강함 - 정규식 등을 이용하여 <시간> 태깅을 가정하면 <시간> 뒤에 <장소명>이 올 가능성이 큼

또한, 인식 속도 및 성능 향상을 위하여 문맥 패턴과 분류 체계를 함께 분석한다. 예를 들면 개체명 인식을 통해 부장, 대리와 같은 직위 개체명과 2호실, 제3회의실 등의 지역(호실) 개체명이 인식된 경우 이벤트 개체명은 회의, 미팅과 관련될 확률이 높으므로 인식 모듈이 탐색을 수행해야 할 도메인을 제한할 수 있다.

2.2 개체명 범주 분류

개체명은 인명, 시간, 장소명, 이벤트명 등 일정한 범주에 의해 분류된다. 기본적으로 휴대폰에 저장되는 일정 관련 내용을 점심/저녁 식사 모임 등 개인별 사적인 약속, 세미나 일정, 경조사 일정, 여행 예약 등 일상적인 일정들을 유형별로 분류한다. 그 예는 아래 표 2와 같다. 개체명 범주는 각 일정의 유형 및 세부 관리 항목에 따라 세분화할 지 여부를 결정한다.

개체명 범주들을 효율적으로 설계하고 시스템에 새로운 키워드가 입력되었을 때 키워드를 기존의 범주에 삽입할 지 새로운 범주를 생성할 지를 결정하는 등의 개체명 범주 설계 및 관리에 대한 연구를 수행한다. 예를 들면, 위키백과의 범주화된 데이터들을 연구하는 것으로 시스템의 분류 체계를 구성하는데 도움을 받을 수 있다.

표 2. 개체명 범주 분류와 예제

개체명 범주	예제
인명(Person)	이름만(Fname): 준섭, 광혁 성만(Lname): 김씨 직위(Position): 부장, 과장, 대리 직업(Occupation): 의사, 교수
지역(Location)	건물(Building): 삼성병원, 더하우스웨딩홀, 강남CGV 호실(Room): 2호실, 4호실, 대강당 위치(Location): 강남역 2번출구, 강남구청, 서울역 노원 롯데백화점 정문, 정문, 고대 정문 돈암동 중앙 버스 정류장
이벤트(Event)	호주어학연수설명회, 생일, 생일파티, 조 모임, 택배도착, 저녁식사(식사모임), 콘서트, 백화점 쎄일, 수강신청, 부부동반모임, 서적구입, 휴강, 병원치료(진료 예약), 렌터카 예약, 배송 예정(택배 수령) 수선, 맥주 한잔, 칵테일 뷔페, 면접
날짜(Date)	2월21일, 2013/03/16
시간(Time)	08:00-10:00 사이, 6시, 세시, 아침, 저녁 8시, 3주후, 세 시간 후(뒤)
요일(Weekday)	목요일

3. 개체명 인식

3.1 개체명 인식 시스템 구조

개체명 인식 시스템은 다양한 카테고리의 훈련 데이터를 저장하고 있는 개체명 사전과 실제로 개체명 인식을 수행하는 개체명 인식 모듈, 입력 스트링에서 메모 패턴을 분석하여 패턴 사전을 갱신하는 패턴 추출 모듈, 개체명 인식 결과를 개체명 사전에 입력 및 갱신하는 개체명 갱신 모듈로 이루어져 있다. 개체명 인식을 완료한 후 사전을 갱신하여 추후의 인식 결과를 개선할 수 있다. 그림 1은 시스템의 전체적인 구조도와 실행 흐름을 나타낸다.

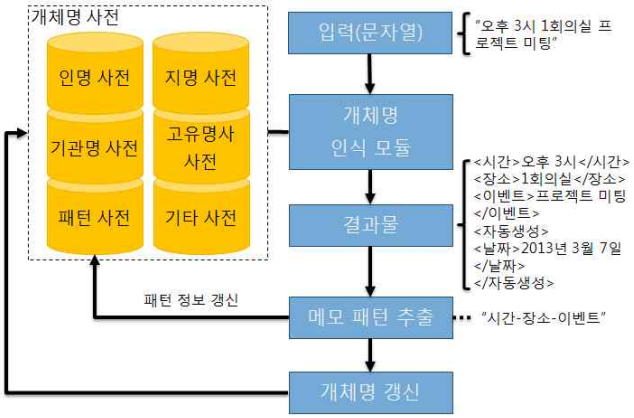


그림 1. 개체명 인식 과정

3.2 개체명 사전

개체명은 정규식 등을 이용하여 비교적 쉽게 인식할 수 있는 것과 그렇지 못한 것으로 구분할 수 있다. 날짜나 시간, 이메일 주소, 전화번호 등은 쉽게 인식할 수 있지만 특정 개체명이라는 외재적 요소가 명확하지 않은 경우에는 인식하기 어렵다. 인명, 지명, 기관명 등 대부분의 개체명이 이에 해당되며, 개체명과 그 범주를 정리해 놓은 개체명 사전을 구축함으로써 효과적인 개체명 인식을 수행할 수 있다.

외부에 공개된 자료 혹은 스마트폰에서 얻을 수 있는 부가 자료를 개체명 사전 구축에 활용할 수 있다. 표 3은 개체명 사전 구축에 외부 데이터를 활용하는 예제를 정리한 것이다. 공개되어 있는 양질의 데이터를 적절히 활용하여 대규모 개체명 사전을 빠르고 효율적으로 구축할 수 있다. 스마트폰 내부의 데이터를 활용하여 각 사용자의 특성에 맞추어진 개인화된 개체명 사전을 구축할 수 있다는 장점도 있다.

표 3. 개체명 자료: 외부 데이터 목록

분류	내용
위키백과/위키사전 데이터 활용	범주화된 방대한 데이터 활용 가능
지역 검색 API 활용	지역명 추정 키워드의 검증 가능
스마트기기에서 획득 가능한 부가정보 활용	주소록 정보를 이용한 인명 사전 구축
국립국어원의 연구보고서 자료 활용	"국어 어휘의 분류 목록에 대한 연구"
정보통신부 고시 우편번호 DB정보 활용	주소 정보, 아파트 및 건물명에 대한 데이터를 이용하여 개체명 사전 구축

날짜와 시간은 정규표현식을 이용하여 대부분 인식 가능하다. 하지만 '오늘', '내일'과 같이 상대적 표현

을 저장하고 있는 사전을 구축하여 활용하면 개체명 인식을 보다 정확하게 처리할 수 있다. 표 4는 날짜와 시간을 인식하는 정규표현식 예제와 관련 용어 사전의 예시이다.

표 4. 날짜와 시간 인식

분류	내용	
정규표현식 사용	([0-9]{4})년 ([0-9]{1,2})월 ([0-9]{1,2})일	예) 2013년 3월 7일
	([0-9]{1,2})월 ([0-9]{1,2})일	예) (2013년) 5월 15일
	([0-9]{1,2})일	예) (2013년 4월) 9일
	[0-9]{1,2}/[0-9]{1,2}	예) 5/3
	[0-9]{1,2}[:시]([0-9]{1,2}(분){0,1}){0,1}	예) 12시, 5시30분, 19:00
관련 용어 사전 구축	'오늘', '내일', '모레'와 같이 상대적인 날짜를 나타내는 단어 포함	
	'오전', '오후', '정오', '자정'과 같이 시간을 나타내는 단어 포함	

인명을 인식하는 작업 또한 사용자의 일정 관리를 위해 필수적이다. 인명 인식을 위해 성씨 사전을 구축하고 사전과 표 5에 설명되어 있는 언어 처리 규칙을 적용하여 인명을 인식할 수 있다. 또한 스마트폰 주소록과 연계하여 인명을 인식하는 방법도 있다. 스마트폰 주소록의 활용은 모바일 기기에서 동작하는 사용자에게 특화된 시스템으로서의 의미가 있다.

표 5. 인명 인식 방법

분류	내용
성씨 사전 구축 및 활용	어절을 구성하는 음절수를 조사, 어절의 앞부분을 성씨사전과 대조하여 인명사전에 등록
스마트기기의 주소록 정보 활용	사용자의 지인과 약속을 가질 확률이 높으므로 스마트 기기의 주소록 정보를 이용하여 인명사전 구축

3.2 자연어 문장의 개체명 인식

일정 관리 메모는 "오후 1시 김민수 대리 결혼식"의 예와 같이 개체명들만 나열하는 것이 일반적이다. 그러나 음성인식을 통한 일정관리 등에서 "승화야 내일 3시에 만나자"와 같이 문법형태소를 생략하지 않고 자연스러운 문장으로 입력되는 경우도 있다. 일반적인 텍스트 문장 입력은 형태소 분석 등 언어처리 기법을 활용하여 개체명을 인식한다. 그러나 스마트폰에서 일정 메모 환경은 개체명 나열을 우선으로 처리하고 개체명이 인식되지 않은 경우에 간단한 조사 절단 과정을 거쳐 개체명을 인식하는 방법을 사용한다.

표 6은 실제로 개체명 인식이 되는 예시를 보여준다. 입력 스트링으로부터 정규표현식, 개체명 사전 탐색, 패턴 및 규칙 매칭 등의 방법을 이용하여 개체명을 인식하고 해당 문자열의 앞·뒤에 개체명 태그 정보를 추가하여 인식 결과를 사용자에게 출력한다.

표 6. 개체명 인식 예제

메모 원본	처리 과정	결과 예시
"25일 3시 ㅇㅇ회의실 과제 미팅"	정규식을 통해 시간 정보 획득	<시간>25일 3시</시간> <장소>ㅇㅇ회의실</장소> <이벤트>과제 미팅</이벤트>
	개체명 사전 이용, 일치 문자열 탐색	
"김철수 사장 미팅"	규칙 적용 검색(인명): 성씨 사전 + 해당 어절의 음절수 정보 이용	<인명>김철수</인명> <직함>사장</직함> <이벤트>미팅</이벤트>
	입력 패턴 검사: 인명 뒤 단어는 직합으로 추정, 사장을 단서 어휘로 사용하여 해당 개체명 사전 조사	

모바일 기기 내부의 데이터를 다루다 보면 아래 표의 내용과 같은 예외 상황들이 발생한다. 표 7의 예시를 포함한 다양한 예외 상황들을 처리하여 사용자의 신변 정보들을 인식 시스템에 적용, 개체명 인식의 정확도를 높일 수 있다.

표 7. 예외 상황 및 처리 방안

메모 원본	예외 상황	처리 방법
"승화야 내일 3시에 만나자"	스마트폰 주소록에 성을 포함한 이름이 저장(고승화)	개체명 사전 구축시 성과 이름을 구분하여 저장
"오후 1시 김민수 대리 결혼식"	스마트폰 주소록에 '김민수'가 2명 이상인 경우	과거의 분석 패턴을 조사 (김민수 '대리'로 인식한 메모 패턴과 대조)

표 8은 고빈도 조사를 분리하는 방법에 의해 조사가 분리되는 비율을 조사한 것이다. 이 표에 의하면 고빈도 조사를 상위 30여개로 한정하더라도 대략 95% 정도의 조사를 처리할 수 있다. 조사 절단 과정은 개체명 사전에 등록되지 않은 문자열을 대상으로 한다. 개체명 사전에서 개체명을 검색했을 때 검색되지 않은 문자열의 끝부분이 조사일 가능성이 있는 문자열에 대해 조사로 사용될 수 있는지를 확인한다. 이 문자열의 끝부분이 조사로 사용가능한 경우에 조사를 분리한 후에 개체명 인식 과정을 적용한다.

표 8. 고빈도 조사 통계

말뭉치%	논문요약	신문기사	국민학교 교과서	문학작품	평 균
70%	8	9	9	9	9
90%	16	20	20	22	20
95%	25	31	32	39	32
99%	49	65	68	93	69

4. 결론

본 논문은 모바일 기기에서 일정관리 편의성을 위해 일정관리와 관련된 개체명을 인식하는 방법을 제안하였다. 위키백과, 우편번호DB와 같이 공개되어 있는 유용한 외부 데이터와 모바일 기기 내부의 부가 정보들을 활용하여 개체명 사전들을 구축하는 방법을 소개하였다. 또한 구축한 개체명 사전, 정규표현식, 메모 패턴 분석 등의 방법을 이용하여 입력 스트링으로부터 개체명들을 인식하는 방법을 소개하였다.

본 논문에서 소개한 개체명 인식 시스템은 개체명 인식의 역할을 수행할 뿐 아니라 인식 결과를 이용하여 개체명 사전을 갱신, 시스템을 개선한다. 또한 모바일 기기의 정보를 개체명 사전의 구축 및 갱신에 이용하기 때문에 시스템을 이용할수록 사용자에 특화된 사전이 구축되어 정확도를 높일 수 있을 것으로 기대한다.

참고문헌

[1] Y. Wang, “Annotating and Recognising Named Entities in Clinical Notes,” In Proc. of ACL-IJCNLP 2009 Student Research Workshop, pp.18-26, 2009.

[2] C. Li, J. Weng, Q. He, Y. Yao, A. Datta, A. Sun, and B. S. Lee, “TwiNER: Named Entity Recognition in Targeted Twitter Stream,” SIGIR'12, pp.721-730, 2012.

[3] A. Ritter, S. Clark, M and O. Etzioni, “Named Entity Recognition inTweets: An Experimental Study,” In Proc. of EMNLP 2011, pp.1524-1534, 2011.

[4] J. Polifroni, I. Kiss, M. Adler, “Bootstrapping Named Entity Extraction for the Creation of Mobile Services,” In Proc. of LREC 2010, pp.1515-1520, 2010.

[5] T. Finin, W. Murnane, A. Karandikar, N. Keller, J. Martineau, and M. Dredze, “Annotating named entities in Twitter data with crowdsourcing,” In Proc. of the NAACL HLT 2010 Workshop on Creating Speech and Language Data with Amazon’s Mechanical Turk, pages 80-88, 2010.

[6] X. Liu, S. Zhang, F. Wei, and M. Zhou, “Recognizing Named Entities in Tweets,” In Proc. of ACL, pp.359-367, 2011.

[7] D. Nadeau and S. Sekine, “A Survey of Named Entity Recognition and Classification,” Linguisticae Investigationes, 30(1), pp.3-26, 2007.

[8] D. M. Oliveira, A H. Laender, A Veloso, A. S. Silva, FS-NER: A Lightweight Filter-Stream Approach to Named Entity Recognition on Twitter Data, In Proc. of WWW 2013, pp.597-604, 2013.

접속 부사의 사용에 따른 설득문과 보도문의 대응 분석

김혜영○, 강범모
줌 인터넷, 고려대학교
simpson15@korea.ac.kr, bmkang@korea.ac.kr

Correspondence Analysis of Reports and Persuasives based on a Newspaper Corpus

Hye-Young Kim○, Beom-Mo Kang
Zum internet, Korea University

요 약

본 논문은 동아, 조선, 중앙, 한겨레 신문의 2000~2011년 신문 사설과 보도문에서서 나타나는 접속부사의 사용에 대한 분석이다. 구체적으로, 텍스트 구조를 드러내는 표지의 기능을 하는 접속부사에 대해 논의하고자 한다. 12년 동안 출현한 고빈도 접속부사 '그러나, 하지만, 그런데, 그리고, 따라서, 그래서, 그렇지만, 그러면, 그러므로, 하물며'를 대상으로 보도문에서의 빈도 변화와 신문 사설에서의 빈도 변화를 대응 분석과 군집 분석을 통해 객관적, 통계적, 통시적으로 분석하였다. 연구 결과, 나열의 구조에서 보도문은 '그리고'를 선호하고 신문 사설은 '하물며'를 선호하여 사용하며, 대조의 표지로서 보도문은 '하지만'을 신문 사설은 '그러나, 그렇지만'을 선호하여 사용하였다. 화제 전환을 나타낼 때 보도문은 '그러면'을 사용하는 반면 신문 사설은 '그런데'를 사용하고, 문제에 대한 결과를 제시할 때 '보도문'은 '그러므로, 그래서'를 신문 사설은 '따라서'를 더 많이 사용하는 경향이 나타났다.

주제어: 신문 사설, 보도문, [물결 21] 코퍼스, 접속 부사, 대응 분석, 군집 분석

1. 서론 및 관련 연구

담화에서 문장의 첫머리에 등장하는 항목은 화자나 작가가 앞으로 이야기하려는 대상으로서, 가장 관심을 두고 있는 부분이다[1]. 이러한 항목(주제; theme)은 명사구, 부사구, 전치사구, 접속부사 등이 될 수 있다. 이 중 접속부사는 'that is, moreover, meanwhile, likewise, in other words' 등과 같이 선행 텍스트와 해당 문장 사이를 연관시키는 것을 말한다[1]. 즉 작가가 말하는 내용간의 관계를 알리면서 전달하고자 하는 메시지를 시작할 때 이는 문장의 시작 부분에서 등장하기 쉽다. 또한 접속부사는 텍스트의 구조 자체를 나타내는 표지가 될 수 있다[2]. 즉, 접속부사를 살펴봄으로써, 단락과 단락 사이의 의미 구조 관계의 양상을 살펴볼 수 있고 이러한 구조적 파악을 통해 주제 파악 또한 가능한 것이다. [3]은 논증문에서 접속부사가 주된 논증 관계를 가장 잘 보여준다며 시간관계(temporal connectives: first, when, now, then 등), 인과관계(causal-conditional connectives; so, consequently, moreover, as a result of 등), 비교관계(comparative connectives; however, rather, whereas 등), 첨가관계(additive connectives; also, but, and, in addition, while 등)로 구분하여 설명한다. [4]는 이러한 텍스트 구조를 구성하는 표지를 명제의 연결 관계와 문장 단위 연결, 내용 단락 단위의 구조를 살펴서 '나열, 강조, 인과, 예시-정의, 포괄-분류, 대조-비교, 전환' 의 일곱 가지 종류의 표지로 분석한다.[1] 따라서 본 절에서는 텍스트 구조를 드러내는 표지의 기능을 하는 접속부사에 대해 논의하고자 한다. 12년 동안 신문에 출현한 고빈도 접속부사 '그러나, 하지만, 그런데, 그리고, 따라서, 그래서, 그렇지만, 그러면, 그러므로, 하물며' 를 대상으로 보도문에서의 빈도와 사설에서의 빈도를 대응 분석과 군집 분석을 통해 분석, 기술할 것이다.

2. 연구 대상 및 연구 방법

본 연구에서는 [물결 21] 코퍼스를 주요 연구 대상으로 한다. [물결 21] 코퍼스는 대규모 신문 언어 자료를 통한 한국의 언어, 사회, 문화적 특성과 그 변화를 연구하기 위해서 고려대학교 민족문화연구원에서 수행 중인 [물결 21] 사업에서 구축된 신문 텍스트이다. 이는 동아일보, 조선일보, 중앙일보, 한겨레신문과 협약을 맺어서 2000년 이후의 전체 기사문을 제공받아 코퍼스로 구축한 것이다[5].

[물결 21] 코퍼스는 현재 2000년~2011년까지의 자료, 즉 약 5억 어절로 구성되어 있으며, 이 중에서 본 연구에서 다루는 신문 사설 수는 약 4만 천여 개, 어절 수는 약 천백만 어절에 이른다. 신문 사설의 수는 전체 기사문 수의 약 1.8%를 차지하고, 어절 수는 약 2.1%를 차지하고 있다.

1) 나열구조 표지는 '등, 또, 또한, 다시, 아울러, 더불어' 등이, 강조구조 표지는 '가장, 특히, 매우, 특히' 등이, 인과구조 표지는 '따라서, 그러므로, 그러면, 그래서, 비로소' 등이, 예시-정의구조 표지는 '즉, 말하자면, 실제' 등이, 포괄-분류구조로는 '모두, 함께, 각각' 등이, 대조-비교구문은 '그러나, 하지만, 혹은, 반면' 등이, 전환구조로는 '그런데, 한편, 물론' 등이 있다[4].

표 1 신문 사설과 연구대상인 보도문의 기사 수 및 어절 수

사설 수	사설 어절 수	사설 1개당 평균 어절 수	보도문 기사 수(일부)	보도문 기사 어절 수(샘플링)
41,482	10,631,033	256.28	163,349	36,406,459

연구 대상으로 삼은 샘플링된 보도문의 기사 수(전체 기사문의 1/10 규모)는 신문 사설의 약 4배의 크기이다. 표 2는 보도문과 신문 사설에 12년 동안 출현한 접속부사의 빈도를 정리한 것이다. 연구 대상이 되는 접속부사는 코퍼스에서 고빈도 순으로 선정하였다.

표 2 보도문과 신문 사설에서의 접속부사 사용 빈도(십만 어절 당)

목록	2000	2001	2002	2003	2004	2005	2006	2007	2008	2009	2010	2011
A그러나	53,747	49,707	49,257	48,141	46,972	42,850	37,993	36,048	35,687	33,340	29,728	28,968
A따라서	6,683	6,632	6,083	6,259	5,741	5,652	5,299	5,448	4,915	4,933	4,433	3,415
A그리고	8,739	9,182	9,347	9,128	8,338	9,672	9,654	9,308	8,193	7,322	8,825	8,463
A그런데	4,075	4,522	4,509	5,127	5,004	6,080	6,656	7,448	6,937	6,534	7,912	7,463
A하지만	18,553	21,400	22,304	22,147	24,613	24,795	28,547	29,685	33,836	38,043	37,256	40,923
A그래서	4,626	4,762	4,973	5,329	5,608	6,364	6,529	6,830	6,245	5,890	7,117	6,513
A그렇지만	1,188	1,078	887	973	1,014	1,212	1,356	951	769	655	794	681
A그러므로	179	287	244	227	203	294	616	532	238	231	290	222
A그러면	708	848	794	834	835	837	841	954	814	720	868	947
A하물며	68	71	60	52	45	59	61	66	45	50	77	47
B그러나	52,048	51,681	48,716	44,643	43,064	46,929	46,929	47,370	48,944	42,589	38,860	30,913
B따라서	10,723	10,562	9,208	9,249	10,291	6,820	6,820	3,535	4,555	3,892	4,834	2,749
B그리고	10,410	10,487	11,996	8,539	7,055	7,204	7,204	5,997	6,250	7,733	6,466	7,351
B그런데	10,097	10,988	10,314	14,263	13,981	13,482	13,482	16,751	14,415	13,063	14,885	15,951
B하지만	9,575	9,007	13,324	16,873	18,511	17,096	17,096	17,572	17,232	23,249	28,489	31,595
B그래서	4,305	4,014	3,586	3,892	4,531	6,210	6,210	5,661	5,590	5,905	4,129	6,862
B그렇지만	913	1,179	1,262	1,122	690	836	836	926	1,430	1,151	1,248	1,068
B그러므로	757	702	354	69	43	90	90	231	264	406	192	261
B그러면	391	301	221	343	820	655	655	968	726	508	480	875
B하물며	391	401	531	229	345	294	294	274	176	169	64	227

표 2에서 A는 보도문을, B는 신문 사설을 의미하며, 십만 어절 당 상대빈도로 재계산하였다.

대응 분석은 분할표 자료에 대해 행과 열 범주간의 유사성, 연관 관계, 상호 관련성 등을 파악하기 위한 통계적 분석 방법이다[6]. 즉, 범주형으로 관측된 두 변수간의 관계를 다차원 공간상(perception map)에서 시각적·도식적으로 파악하고자 할 때 유용하다. 대응 분석은 이원 분할표에 기재된 행과 열의 모든 범주들 사이의 상대적 분포에 대해서 통계적인 상관관계와 군집 유형을 보여준다. 이 때, 행렬의 산점도에 대해서 참조선을 표시하여 정규화할 수 있다. 열 범주와 행 범주간의 관계를 살펴보기 위해서, 열 범주 좌표를 원점에 대해 화살표로 나타내고, 행 좌표를 이들 두 축에 직교하는 점선을 표시하여 상관성을 해석할 수 있다. 주로 대응 분석은 통계 모형이나 가설에 대한 관찰 자료의 검증이 아닌 관찰 자료의 분포적 특성에 대한 해석을 목적으로 할 때, 살펴보고자 하는 범주의 상대 분포를 통합적으로 고려할 때, 그리고 시각적인 자료 해석을 목적으로 할 때 사용한다[7].

본 연구에서는 대응 분석을 R 통계 프로그램을 이용하였다. R 통계 프로그램에서 대응 분석은 ca package를 설치한 후 plot(ca())함수를 사용한다[7]. 활용 예시는 아래 그림과 같다.

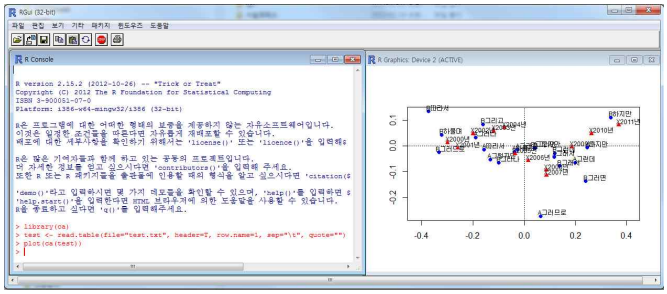


그림 1 R 통계 프로그램을 활용한 대응 분석 예시

대응 분석에서는 차원 축소된 공간에서의 행과 열 범주 좌표값이 정규화 방법에 따라 계산된다. 그림 1에서처럼 행 범주에서 한 차원 좌표평면 상에 위치한 범주들 간의 거리는 유사성 정도를 나타내서 가까이 위치한 범주일수록 유사성이 크다고 해석할 수 있다. 열 범주 좌표차원에서는 열 범주 좌표를 원점에 대해서 화살표로 나타내서 이들 두 축에 직교하는 점선을 표시한 후 더 가까운 쪽과 유사하다고 해석할 수 있다.

3. 연구 결과

표 2에서 12년 동안 추출된 보도문에서의 사용 분포는 '그러나>그런데>따라서>그리고>그래서>하지만>그렇지만>그러므로>그러면>하물며' 의 순으로, 신문 사설에서는 '그러나>하지만>그런데>그리고>따라서>그래서>그렇지만>그러면>그러므로>하물며' 의 순으로 나타났다. 또한 '그러나' 는 신문 사설과 보도문 모두에서 가장 많이 사용된 접속부사로 나타났다. '그러나' 는 텍스트 생산자의 주장을 효과적으로 제시하고 논의의 범위를 확대시키거나 한정하는 방법으로 텍스트를 발전시키고 전개하는 등, 텍스트의 주제를 드러내는데 매우 중요한 기능을 한다[8]. 특히 논증문에서는 주제를 전개하는 표지로 텍스트 생산자의 신념이나 의견을 펼치는 데에 사용된다. 대체로 설명을 나타내는 글에서는 논리적 대등관계를 형성하는 나열 구조를 많이 쓴다고 알려져 있지만, 실제로는 보도문과 신문 사설 모두 '그러나' 가 가장 많이 사용되는 것으로 드러났다.

(1) 보도문과 신문 사설에서의 ‘그러나’ 사용 예시

ㄱ. 보도문에서의 사용 예시

- 잡스로서도 당시 IT업계의 황제였던 MS의 게이츠를 견제하기 위해 구글의 슈밋이 필요했다. **그러나** 잡스와 슈밋의 밀월은 오래가지 못했다.

ㄴ. 신문 사설에서의 사용 예시

- 최근 여야 구분 없이 정치권 곳곳에서 복지를 외치는 목소리가 높아지는 것도 자연스러운 현상이다. **그러나** 아무리 좋은 말이고, 아무리 절박한 상황이라 하더라도 국가의 정책이라면 당연히 원칙이 분명하고 현실성이 있어야 한다.

(1)에서 보도문과 신문 사설에서 사용된 ‘그러나’의 사용 의미는 상이하다. 대체로 선행문에서 기대하는 바는 일반적으로 기대할 수 있는 정도에 부합하는데, 이러한 기대가 선행 문장으로 인해 야기되긴 하지만 후행 문장에서 진술하는 사실이 그와 부합하지 않을 때, 보도문에서는 ‘그러나’를 사용해서 두 문장을 접속하고 있다. 반면 신문 사설에서 ‘그러나’는 작자의 가치 평가 속에서 부정적인 것으로 필자의 주관에 의해 결정될 때 사용되고 있음을 알 수 있다. 이때 ‘그러나’는 ‘안타깝다, 다를 것 없다’ 등의 술어와 함께 앞의 사실에 의한 대조보다는 훨씬 작가의 주관적인 인식을 반영하고 있다. 즉, 보도문에서의 ‘그러나’의 사용은 함축적인 성향을 가지지만, 신문 사설에서의 ‘그러나’는 주관적인 성향을 나타낸다.

다음으로 ‘그러나, 하지만, 그렇지만’은 보도문보다 신문 사설에서 더 많이 사용된 접속부사로 그림 2에서 나타났다. ‘그러나’와 ‘그렇지만’은 1.1배 가량, ‘하지만’은 3.5배 가량 신문 사설에서 그 쓰임이 더 많다. 이들은 모두 앞에서 기술한 내용과 반대 내용을 지시할 때 사용되는 표지로[4], 이들의 12년간 사용 분포는 다음과 같다.

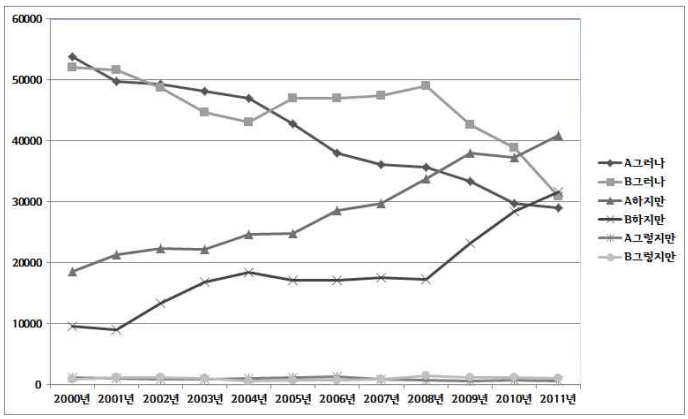


그림 2 ‘그러나, 하지만, 그렇지만’의 12년간 사용 분포(십만 어절 당)

그림 2에서 A는 보도문에서의 사용을, B는 신문 사설에서의 사용을 보여준다. 흥미로운 점은 세 개의 접속부사 모두 문제가 되는 상황을 제시한 이후에 원인을 진단, 평가하거나 구체적, 혹은 좀 더 심각하게 제시할 때 사용하기 위해 쓰인다는 점이다. 2000년대 초반에는 ‘그러나’가 이러한 역할을 대표하듯이 사용이 압도적으로 많았고, ‘하지만’은 그 다음으로, ‘그렇지만’은 미미한 사용을 보였다. 하지만 2011년에 이르러서는 보도문과 신문 사설에서 ‘그러나’보다 ‘하지만’의 사용이 더 많다. 즉, 신문 텍스트에서 과거에는 대조를 나타내는 문맥에서 ‘그러나’를 주로 사용했다면, 최근에는 ‘하지만’을 선호해서 사용함을 알 수 있는 부분이다.

[2]에서는 논증텍스트 구조는, 문제는 기존의 상황에서 발생하므로 문제제기 단락 앞에는 문제 발생 전의 상황을 제시하는 단락이 오고, 이후 배경제시 부분은 ‘배경-[문제-해결]-평가’의 3분 구조의 첫 부분을 담당한다고 기술한다. 이후에 필요한 상황을 보여주거나 사건에 대한 견해를 첨언하는 내용이 따르고 화자의 주장이나 단언을 나타내는 해결을 제시한 후(‘문제-해결’), 정리, 요약을 통해 텍스트를 마무리하여(마무리 부분) ‘정반합’ 구조를 신문 사설에서는 보여준다. 특히 ‘배경’과 ‘평가’ 부분보다는 텍스트의 절대적 비중은 내용의 구조를 나타내는 ‘문제-해결’ 부분에 있다고 그녀는 주장한다. 이러한 논증문의 구조 단위에 따라 텍스트의 구조를 구성하는 표지를 분류해보았다.

표 3 논증 구조에 따른 접속부사의 표지 구분

배경	배경 제시 및 나열구조 표지	그리고, 하물며
문제-해결	문제 진단을 위한 대조 표지	그러나, 그렇지만, 하지만
	화제 전환 및 문제 야기	그런데, 그러면
평가	문제 해결 및 결과 제기	그래서, 그러므로, 따라서

표 3에 따른 보도문과 신문 사설에서의 표지별 사용 분포는 그림 3과 같다.

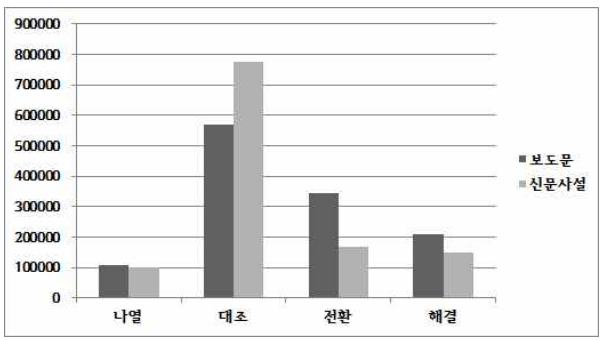


그림 3 접속표지에 따른 보도문과 신문 사설의 사용 분포(십만 어절 당)

대조를 나타내는 표지인 ‘그러나, 그렇지만, 하지

만’ 의 사용은 신문 사설에서 더 많이 사용되고, 그 밖에 배경을 제시하거나 문제를 제기하고 문제의 결과나 평가를 나타내는 표지들은 보도문에서 더 많이 사용하는 것으로 나타났다. 이 중 나열 구조를 나타내는 표지인 ‘그리고’ 와 ‘하물며’ 는 다른 접속 표지와는 달리 문단과 문단, 문장과 문장의 연결 고리 역할 뿐만 아니라, 문장 중간에도 출현하는 경향을 보여 다른 접속부사와는 신문텍스트에서 상이한 양상을 보였다.

그림 4는 2000년부터 2011년까지의 발간 년도에 따라 보도문에서 접속부사와 발간 연도 사이의 대응 분석 결과를 보여준다.

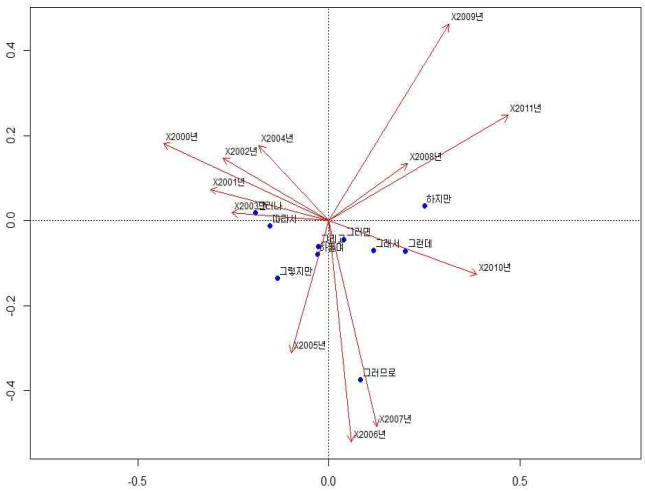


그림 4 보도문에서 접속부사의 사용 양상

2사분면은 2000년부터 2004년까지의, 3사분면은 2005년의, 4사분면은 2006년부터 2007년과 2010년의, 1사분면은 2008년부터 2011년의 분포 공간으로 나타났다. 이 때 각 사분면에 위치하는 접속부사들은 해당 연도의 보도문에서 가장 우세한 분포 성향을 보인다. 즉, ‘그러나’ 와 ‘따라서’ 는 2000년대 초반에 우세하게 쓰인 반면, 2000년 후반에는 ‘그러면, 그래서, 그런데, 하지만’ 이 우세하게 사용되고 있다. 또한 접속부사와 원점, 발간 시기 사이의 각도를 측정하여 접속부사와 발간 시기의 상관성을 해석할 수 있는데, ‘그러나, 따라서’ 는 감소 추세를, ‘그러면<그래서<그런데<하지만’ 의 순으로 점차 증가 추세를 보이고 있다. 두 발간 년도 중간에 위치한 ‘그렇지만, 하지만, 그러면’ 은 두 발간 시기 사이의 분포 차이가 크지 않은 반면, 발간 연도의 선에 분포해 있는 ‘그러나, 그리고, 하물며, 그러므로, 그런데’ 는 해당 발간 연도와 분포 차이가 현저한 특징을 드러낸다. 또한 2000년부터 2004년까지와 2006년부터 2007년 사이는 각도가 적게 나타나서 해당 연도 사이에는 보도문의 분포적 유사도가 높은 것으로 나타났다. 원점은 대응 분석에 의한 결과에서 전체 접속부사에 대한 전체 발간 시기의 일반적인 분포를 나타낸다. 따라서 원점으로 가장 멀리 위치하는 ‘그러므로’ 는 2007년에 분포 강도가 더 큰 것으로 해석할 수 있다. 반면 원점에 근접하는 ‘그리고, 하물며, 그러면’ 은 2005~2007년에 대한 분포적인 상관성이 있지만 다른 발간 연도의 분포와 비교해서 그 차이가 크지는 않은 것이다.

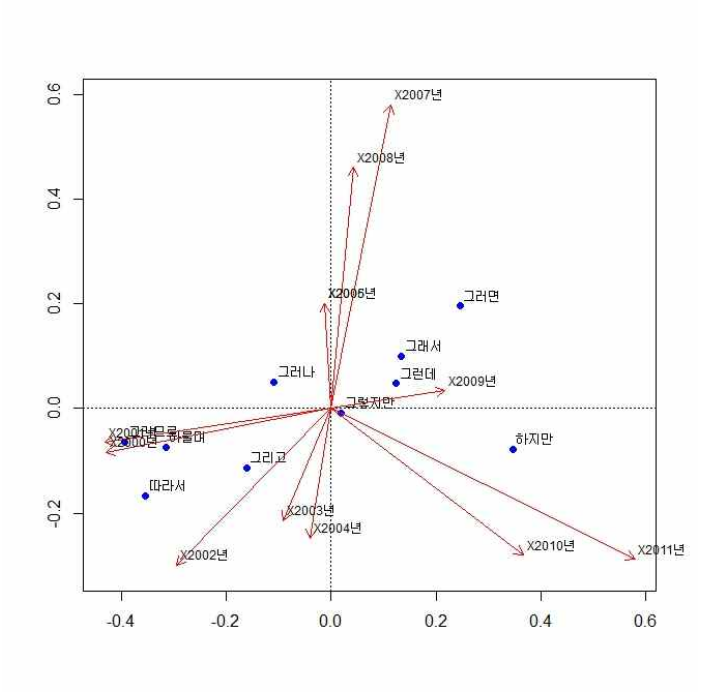


그림 5 신문 사설에서 접속부사의 사용 양상

그림 5는 2000년부터 2011년까지의 발간 연도에 따른 신문 사설에서의 대응 분석 결과를 보여준다. 3사분면은 2000년부터 2004년까지의, 1사분면은 2006년부터 2009년까지의, 4사분면은 2010년부터 2011년의 분포 공간으로 나타났다. 2000년대 초반에는 ‘그러므로, 하물며, 따라서, 그리고’ 와 같은 문제의 배경이나 결과를 설명하는 접속부사들이 우세하게 쓰인 반면, 2000년 후반에는 ‘그렇지만, 그런데 하지만’ 등 문제와 해결을 나타내는 접속부사가 우세하게 사용되고 있다. ‘그러므로, 하물며, 따라서, 그리고, 그러나’ 는 12년 동안 감소 추세를 나타낸 반면, ‘그런데, 그래서, 그러면, 하지만’ 은 증가 추세를 보인다. 2000년부터 2001년까지와 2003년부터 2004년 사이, 그리고 2006년부터 2008년 사이는 각도가 적게 나타나서 해당 연도 사이에는 신문 사설의 분포적 유사도가 비교적 높은 것으로 나타났다. 보도문과 신문 사설 각각에서 우세한 접속부사의 표지별로 구분하면 표 4와 같다.

표 4 보도문과 신문 사설에서 더 우세한 접속부사의 분류

	보도문	신문 사설
배경 제시 및 나열구조 표지	그리고	하물며
문제 진단을 위한 대조 표지	하지만	그러나, 그렇지만
화제 전환 및 문제 야기	그러면	그런데
문제 해결 및 결과 제기	그러므로, 그래서	따라서

보도문은 2000년~2001년 사이에 '그리고'를 중심으로 텍스트 구조를 형성하여 이유, 나열의 기술을 사용하는 특성이 강하게 나타났음을 알 수 있다. 이후에는 보도문의 성격이 '그러면'과 '하지만'을 활용하여 화제를 전환하거나 문제를 제시하는 성격이 점차 짙어졌음을 알 수 있다. 반면 신문 사설에서는 '그러나, 그렇지만' 처럼 문제를 진단하는 성격의 표지를 2000년대 초반에 주로 사용하여 점차 '그런데'를 사용하는 특성을 보이고 있다. 즉, 신문 사설에서는 항상 문제를 제시하거나 진단하는 특성을 보여주는 것이다. 이는 시간이 지날수록 보도문과 신문 사설의 성격 차이가 문제 제시를 중점적으로 나타내는 방향으로 귀결되어 가고 있고, 한때 벌어졌던 텍스트 구조의 차이가(2003~2008년) 시간이 지날수록 점차 줄어들고 있음을 보여주는 부분이다.

다음 그림은 켄달 지수에 따른 보도문과 신문 사설에서의 접속사 군집 분석 결과를 보여준다.[1)]

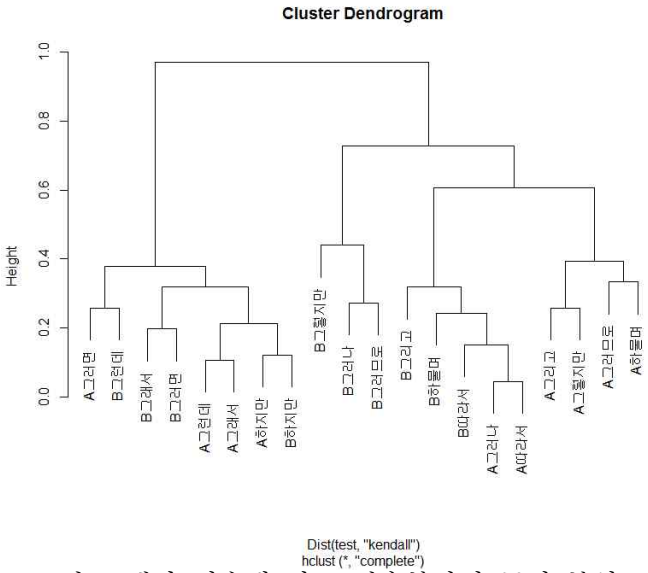


그림 6 켄달 지수에 따른 접속부사의 군집 분석

보도문에서 사용되는 접속부사 '그리고, 그렇지만, 그러므로, 하물며'와 신문 사설에서 사용되는 '그리고, 하물며, 따라서'는 수직선의 길이가 유사하여 텍스트에서의 중요도 면에서 비슷함을 알 수 있다. 다만 접속사 사용의 의도나 의미상에 두 그룹은 차이를 보여준다. 또한 '그러나'의 사용 정도는 보도문에서는 '따라서'와 상관성을 보이지만, 신문 사설에서는 '그러므로'와 상관성을 보였다는 점이 흥미롭다. 이는 신문 사설이나 보도문 모두 문제에 대한 진단을 하기 위한 대조 표현으로 '그러나'를 사용하지만, 문제에 대한 해결을 제시하기 위해서 신문 사설은 '그러므로'를, 보도문은 '따라서'를 사용한다고 해석할 수 있겠다. 이러한 사용의 예시는 다음과 같다.

(2) 신문 사설에서 '그러므로', 보도문에서 '따라서'의 사용 예시

ㄱ. 신문 사설에서의 '그러므로'의 사용 예시

・많은 이가 법규를 지키느라 애를 쓰는데 법규를 어긴 사람을 주기적으로 사면해 주면 형평에 맞지 않다. 그리고 "제재를 받아도 얼마 안 있어 사면될 것"이라는 풍조가 새기면 준법의식은 심각한 도전을 받게 된다. **그러므로** 대통령의 사면권은 이러한 부작용을 최소화하도록 신중하고 엄격하게 행사돼야 한다.

ㄴ. 보도문에서의 '따라서'의 사용 예시

・노조가 예정대로 파업에 돌입한다면 서울지하철 사상 처음으로 합법 파업이 된다. 국민 생활에 미치는 영향이 크 s사업장에서 노동위원회가 직권으로 파업을 못하게 막았던 직권중재 제도는 지난 연말로 폐지됐다. **따라서** 예전처럼 지하철 파업을 불법으로 규정하고, 공권력을 투입해 업무를 정상화하는 방식은 어렵게 됐다.

4. 결론

본 연구에서는 [물결 21] 코퍼스를 사용하여 신문 사설과 보도문에서의 어휘 사용에 대해서 살펴 보았다. 그 결과, '그러나'는 신문 사설과 보도문 모두에서 가장 많이 사용된 접속부사로 나타났다. 즉, 보도문과 설명문 모두 대조의 표현이 자주 사용됨을 알 수 있다. 보도문에서만 많이 쓰이는 접속부사는 '그런데'로 2.2배 정도 더 사용이 많았는데 이는 보도문이 텍스트의 화제를 바꾸는 경향성이 신문 사설보다 더 많음을 알 수 있게 하였다. 평가를 표시하는 접속부사 '따라서, 그래서, 그러므로'는, 신문 사설에서 '따라서'와 '그러므로'가 각각 2배 정도 더 많이 사용된 반면, '그래서'는 보도문에서 더 많이 사용되었다. 각각의 접속사를 텍스트에서의 기능에 따라 분류하여 살펴보면 대조를 나타내는 표지인 '그러나, 그렇지만, 하지만'의 사용은 신문 사설에서 더 많이 사용되고, 그 밖에 배경을 제시하거나 문제를 제기하고 문제의 결과나 평가를 나타내는 표지들은 보도문에서 더 많이 사용하는 것으로 나타났다. 결과적으로 나열의 구조에서 보도문은 '그리고'를

1) 두 관찰치 사이의 유사성을 측정할 때 거리를 측정하는 방법으로는 유클리디안 거리(두 관찰치 사이의 거리 측정), 유클리디안 제곱거리(유클리디안 거리의 제곱값), 맨해튼 거리(두 관찰치 사이의 거리로 각 변수값 차이의 절대값의 합) 등이 있고, 유사성을 측정하는 방법으로는 코사인값(두 벡터 X와 Y 사이의 코사인값으로 유사성을 측정), 상관계수(X와 Y 사이의 상관계수로 유사성 측정), 켄달 지수(등위로 상정한 결과값 측정) 등이 있다. 군집 분석은 R 통계 패키지에서 amap package를 설치한 후 hclust()를 사용하였다[9]. 또한 관찰치들의 군집화 과정에서 관찰치들을 순차적으로 유사성의 순서에 따라 연결할 때의 연결방법에는 최단연결법(single linkage), 최장연결법(complete linkage), 중심연결법(centroid linkage), 평균연결법(average linkgage), 워드의 방법(Ward's method)가 있는데 여기서는 각 관찰치가 가장 큰 값(유사성이 가장 작은 값)을 기준으로 연결하는 최장연결법을 사용하였다[10]. 본 연구에서는 여러 연구방법 중 가장 그룹 차이가 선명하게 드러난 켄달 지수를 소개한다.

선호하고 신문 사설은 ‘하물며’를 선호하며, 대조의 표지로서 보도문은 ‘하지만’을 신문 사설은 ‘그러나, 그렇지만’을 선호하여 사용하였다. 화제 전환을 나타낼 때 보도문은 ‘그러면’을 사용하는 반면 신문 사설은 ‘그런데’를 사용하고, 문제에 대한 결과를 제시할 때 ‘보도문’은 ‘그러므로, 그래서’를 신문 사설은 ‘따라서’를 더 많이 사용하는 경향이 나타났다.

‘하지만’과 ‘그러나’는 공통적으로 대조를 나타내는 표지로 사용된다. 신문에서 이들의 사용은 ‘그러나’를 2000년대 초반에 주로 사용하다가 ‘하지만’의 사용으로 변화하는 경향이 보이는데, 이는 보도문보다 신문 사설에서의 변화가 좀 더 늦게 나타난다. ‘하지만, 그래서, 그러면, 그런데’는 다른 접속 부사들에 비해서 구어성이 더 높은데, 2000년대 후반에 올수록 이들의 사용이 선호되는 것으로 연구 결과 나타나서 신문 문체가 구어적으로 변화하고 있음을 알 수 있었다. 특히 보도문이 먼저 변화하고, 신문 사설은 좀 더 늦게 나타남을 연구를 통해 알 수 있었다.

신문은 새로운 정보를 제공하고 여론을 형성하는 중요한 역할을 한다. 따라서 여느 글에 비해 문어의 특징을 강하게 지니고 잘 다듬어진 글이다. 본 연구의 의의는 통계적인 방법으로 이러한 신문 사설과 보도문의 접속 부사 사용의 (2000년대)변화 양상과 쓰임의 차이, 글의 구조의 차이에 대해서 경험적, 객관적, 통시적으로 제시하였다는 점이다.

참고문헌

[1] R. Fowler, Language in the news: Discourse and ideology in the press, NewYork: Routledge, 1991.

[2] 신지연, “국어 텍스트의 거시구조 접속”, 시학과 언어학 11, 7-23, 2006.

[3] P. Knapp and M. Watkins, *Genre, Text, Grammar: technologies for teaching and assessing writing,* University of new South Wales, 2005.

[4] 양태영, “설명텍스트의 표지와 텍스트구조 분석”, 한국어의미학 31, 109-142, 2010.

[5] 강범모, 김흥규, “명사 빈도의 변화, 관심의 사회적 트렌드: 물결 21 코퍼스 [2000-2009]”, 언어학 61, 3-38, 2011

[6] 양경숙, SPSS 최적 범주형 자료분석, 한나래아카데미, 2009.

[7] M. Greenacre, Correspondence Analysis in Practice, Boca Raton: Chapman & Hall/CRC, 2007.

[8] 신지연, “논증텍스트에서의 ‘그러나’의 주제 전개 기능”, 텍스트언어학 16, 41-63, 2004.

[9] R. H. Baayen, Analyzing Linguistic Data: A Practical Introduction to Statistics Using R, Cambridge University Press, 2008.

[10] 이용구, 김성수, 김현중, 다변량 분석 입문, KNOW PRESS, 2010.

한글 및 한국어 정보처리 학술대회 제25권 제1호

인 쇄: 서기 2013년 10월 05일
발 행: 서기 2013년 10월 06일
발행인: 강승식
편집인: 강승식, 이재원, 남기춘
발행처: 사단법인 한국정보과학회
서울특별시 서초구 방배3동 984-1 (머리재빌딩 401호)
전화: 1588-2728 FAX: (02)521-1352
e-mail: kiise@kiise.or.kr
홈페이지: http://www.kiise.or.kr
인쇄처: 유성기획 (02)943-9666

2013년도 제25회 한글 및 한국어 정보처리 학술대회

한글 및 한국어 정보처리