Bi-LSTM 기반 감성분석을 위한 대용량 학습데이터 자동 생성 방안

An Automatic Method of Generating a Large-Scale Train Set for Bi-LSTM based Sentiment Analysis

저자 (Authors)
최민성, 온병원
Min-Seong Choi, Byung-Won On

출처 (Source)
정보과학회논문지 46(8), 2019.8, 800-813(14 pages)
Journal of KIISE 46(8), 2019.8, 800-813(14 pages)

발행처 (Publisher)
한국정보과학회
The Korean Institute of Information Scientists and Engineers

URL
http://www.dbpia.co.kr/journal/articleDetail?nodeId=NODE08762604

APA Style
최민성, 온병원 (2019). Bi-LSTM 기반 감성분석을 위한 대용량 학습데이터 자동 생성 방안. 정보과학회논문지, 46(8), 800-813

ISSN 2383-630X(Print) / ISSN 2383-6296(Online)
Journal of KIISE, Vol. 46, No. 8, pp. 800-813, 2019. 8
https://doi.org/10.5626/JOK.2019.46.8.800

Bi-LSTM 기반 감성분석을 위한 대용량 학습데이터 자동 생성 방안

(An Automatic Method of Generating a Large-Scale Train Set for Bi-LSTM based Sentiment Analysis)

최 민 성 †　　온 병 원 ††
(Min-Seong Choi)　　(Byung-Won On)

요 약 딥러닝을 이용한 감성분석에서는 감성이 레이블 된 많은 양의 학습데이터가 필요하다. 그러나 사람이 직접 감성을 레이블 하는 것은 시간과 비용에 제약이 있고 많은 데이터에서 감성분석에 적합한 충분한 양의 데이터를 수집하는 것은 쉽지 않다. 본 논문에서는 이러한 문제점을 해결하기 위해 기존의 감성사전을 활용하여 감성점수를 매긴 후 감성 변환 요소가 존재하면 의존 구문 분석 및 형태소 분석을 수행해 감성점수를 재설정하여 감성이 레이블 된 대용량 학습데이터를 자동 생성하는 방안을 제안한다. 감성 변환 요소로는 감성 반전, 감성 활성화, 감성 비활성화가 있으며 감성점수가 높은 Top-k의 데이터를 추출하였다. 실험 결과 수작업에 비해 짧은 시간에 대용량의 학습데이터를 생성하였으며 학습데이터의 양이 증가함에 따라 딥러닝의 성능이 향상됨을 확인하였다. 그리고 감성사전만을 사용한 모델의 정확도는 80.17%, 자연어처리 기술을 추가한 제안 모델의 정확도는 89.17%로 9%의 정확도 향상을 보였다.

키워드: 감성분석, 딥러닝, 학습데이터, 감성사전, 의존 구문 분석, 형태소 분석

Abstract Sentiment analysis using deep learning requires a large-scale train set labeled sentiment. However, direct labeling of sentiment by humans is time and cost-constrained, and it is not easy to collect the required data for sentiment analysis from many data. In the present work, to solve the existing problems, the existing sentiment lexicon was used to assign sentiment score, and when there was sentiment transformation element, the sentiment score was reset through dependency parsing and morphological analysis for automatic generation of large-scale train set labeled with the sentiment. The Top-k data with high sentiment score was extracted. Sentiment transformation elements include sentiment reversal, sentiment activation, and sentiment deactivation. Our experimental results reveal the generation of a large-scale train set in a shorter time than manual labeling and improvement in the performance of deep learning with an increase in the amount of train set. The accuracy of the model using only sentiment lexicon was 80.17% and the accuracy of the proposed model, which includes natural language processing technology was 89.17%. Overall, a 9% improvement was observed.

Keywords: sentiment analysis, deep learning, train set, sentiment lexicon, dependency parsing, morphological analysis

· 이 논문은 2019년도 정부(과학기술정보통신부)의 한국연구재단의 개인기초연구사업(No. NRF-2019R1F1A1060752)의 연구비 지원으로 수행하였습니다.

† 학생회원 : 군산대학교 소프트웨어융합공학과
alstjd517@kunsan.ac.kr
†† 종신회원 : 군산대학교 소프트웨어융합공학과 교수
(Kunsan Nat'l Univ.)
bwon@kunsan.ac.kr
(Corresponding author임)

논문접수 : 2019년 3월 12일
(Received 12 March 2019)
논문수정 : 2019년 5월 16일
(Revised 16 May 2019)
심사완료 : 2019년 5월 20일
(Accepted 20 May 2019)

정보과학회논문지 제46권 제8호(2019. 8)

1. 서 론

기업에서 더 나은 서비스와 제품을 개발하기 위해서는 고객들의 선호도 및 의견을 조사하여 분석하는 것이 중요하다. 과거에는 이와 같은 여론 조사를 위해 설문지나 전화를 이용하는 등의 방법을 사용하였다. 그러나 보통 이러한 방법은 조사원을 고용하고 통계 전문가를 활용해 분석을 진행하는데 이 과정에서 많은 비용이 소모된다[1]. 또한 많은 사람들의 의견을 취합하기에 어려움이 있으며 빠른 시간 안에 결과를 얻기란 어렵다. 최근에는 모바일 기기와 인터넷의 발달로 어디서든 블로그, SNS, 쇼핑몰, 뉴스 댓글 등에 자신의 의견을 자유롭게 표현할 수 있게 되면서 텍스트, 사진, 영상과 같은 비정형 데이터가 실시간으로 생성되고 있다. 이 과정에서 생성되는 텍스트를 이용하면 기존의 방법에 비해 짧은 시간에 많은 양의 데이터를 수집하여 더욱 객관적인 정보를 얻을 수 있으며 마케팅 등 여러 분야에서 유용하게 활용이 가능해 사람들의 선호도 및 의견을 분석할 수 있는 감성분석(Sentiment Analysis) 연구도 활발히 이루어지고 있다.

그림 1은 기존 감성분석 연구와 본 논문의 제안 방안을 비교하여 나타낸 도표이다. 그림 1(a)는 기존 감성분석 연구에 대한 흐름도로 감성분석은 나이브 베이즈(Navie Bayes), 서포트 벡터 머신(Support Vector Machine) 등 전통적인 기계학습 기법을 이용해 많이 연구되어져 왔다. 그러나 전통적인 기계학습 기법의 경우 학습데이터의 특성에 영향이 커서 특정 영역의 데이터로 학습된 모델이 다른 영역에 적용될 경우 영역 적응 문제로 인해 저조한 성능을 보인다[2].

최근에는 딥러닝의 발달로 딥러닝을 이용한 감성분석이 많이 연구되어지고 있으며 뛰어난 성능을 보인다. 그러나 딥러닝은 모델 복잡도가 높기 때문에 과적합(Over-fitting) 문제가 발생할 수 있다. 딥러닝에서 이와 같은 문제를 해결하려면 많은 양의 학습데이터가 필요하다. 또한 전통적인 기계학습의 경우 학습데이터 양이 증가해도 성능이 정체되는 현상을 보이지만 딥러닝의 경우 학습데이터가 증가할수록 좋은 성능을 나타내어 딥러닝에서의 학습데이터 양과 질은 성능을 결정하는 중요한 요소이다[3]. 이처럼 딥러닝을 이용한 감성분석의 좋은 성능을 위해서는 감성이 긍정과 부정으로 레이블 된 많은 양의 학습데이터가 필요하다. 그러나 사람이 직접 모든 데이터의 감성을 레이블 하는 것은 시간과 비용에 많은 제약이 있다. 또한 빅데이터 시대의 많은 데이터에서 감성분석에 적합한 충분한 양의 데이터를 수집하는 것 또한 쉽지 않다.

감성분석을 위한 학습데이터 생성을 위해 사람이 직접 수작업으로 감성을 레이블하는 방법이 있다. 이러한 경우 정확한 학습데이터를 생성할 수 있지만 시간과 비용의 제약이 있어 많은 양의 학습데이터 생성이 어렵다. 영화 리뷰와 같은 평점을 기준으로 감성이 레이블 된 학습데이터를 생성하는 방법이 있다. 이러한 경우 수작업에 비해 데이터 생성에 시간을 절약할 수 있지만 반드시 평점과 내용의 감성이 같지 않아 오류 발생 가능성이 크다. 데이터에서 후보 단어 등 자질을 추출하여 직접 감성사전(Sentiment Lexicon)을 구축하고 그것을 기준으로 감성이 레이블된 학습데이터를 생성하는 방법이 있다. 이러한 경우 해당 데이터에 적합한 감성사전을 구축할 수 있지만 감성사전 구축에 많은 시간과 비용이 소요된다는 단점이 있다.

그림 1(b)에서는 기존에 존재하는 감성사전을 활용하여 감성사전의 긍정과 부정 어휘의 빈도만을 이용해 감성을 판단하였다[4]. 기존에 존재하는 감성사전을 활용하면 감성사전 구축에 의한 시간과 비용을 절약할 수 있다. 그러나 이러한 경우는 감성 어휘를 많이 가지는 학습데이터는 생성할 수 있지만 단순히 감성 어휘의 빈도로만 감성을 판단해 감성이 반전되거나 감성이 활성화, 비활성화 되는 경우와 같은 감성 변환 요소가 고려되지 않아 정확한 감성 파악이 어렵다. 또한 여기에서는 문장을 입력으로 받아서 감성을 판단하였는데 한 문장에 여러 감성이 존재하는 경우 감성을 확실히 정의하기 어렵다는 단점이 있다.

이와 같은 단점을 보완하기 위해 그림 1(c)와 같이 감성분석에서의 학습데이터 부족 문제 해결을 위한 기존의 감성사전과 의존 구문 분석 및 형태소 분석을 활용한 절 단위의 대용량 학습데이터 생성 방안을 제안한다. 이와 같은 목적을 가지고 본 논문에서는 3.1장에서 '입력 문장을 절로 나누는 방법'과 3.2장에서 '감성사전을 활용한 감성 점수 매기는 방법'을 제안하고, 3.3장에서는 3.2장에서 제안한 방법의 문제점에 대해 설명한다. 그리고 3.4장과 3.5장에서는 그 문제를 해결하기 위한 '의존 구문 분석과 그 관계 및 형태소 분석을 통해 감성이 변환되는 경우'를 제안하며 3.6장에서는 '감성 점수에 따른 감성을 레이블 하는 방법'을 최종적으로 제안한다.

이를 위해 본 연구에서는 네이버 카페 "한국 형태소 분석 등 NLP 연구개발 자료"에서 제공하는 1억2천3백만 가량의 단어로 구성된 한국어 원시 말뭉치(약1천만 문장)[5]를 사용하였으며, 문장을 절 단위로 나누는 과정을 수행하였다. 그리고 이 절을 입력으로 감성사전을 통해 어절 단위로 감성 점수를 매긴다. 감성사전을 활용하면 특정 데이터가 특정 감성으로 분류되었을 시 그 이유를 설명할 수 있는 장점이 있다[6]. 그 후 의존 구문 분석 및 형태소 분석을 통해 감성 반전, 감성 활성

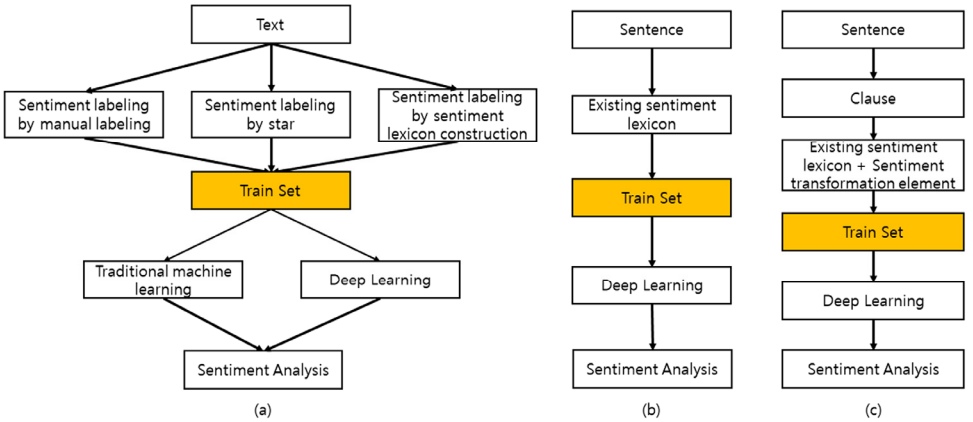

그림 1 감성분석 방법

Fig. 1 Method of sentiment analysis

화, 감성 비활성화 되는 감성 변환 요소를 찾고 그 결과를 감성 점수에 다시 적용하여 최종적으로 입력 절의 감성을 판별한다. 의존 구문 분석을 사용하면 한국어의 생략 및 어순 자유성에 의한 문제점을 해결 가능해 한국어 구문 분석에 잘 활용할 수 있다. 이와 같이 감성이 판별된 절을 감성 점수가 높은 순대로 긍정과 부정 각 150,000개 씩 300,000개를 추출하여 딥러닝을 이용한 감성분석의 학습데이터로 사용한다. 이 때 딥러닝 기법은 앞 뒤 문맥을 살펴볼 수 있는 Bi-LSTM(Bi-directional LSTM)을 사용하였다. 이후 성능 평가를 위해 수작업 및 감성사전만을 사용한 모델과 비교하여 제안 방안의 우수성을 입증하였다.

이 논문의 구성은 다음과 같다. 제 2장에서는 제안 방안의 이해를 돕기 위해 관련 연구에 대해 기술한다. 제 3장에서는 제안 방안인 '감성사전과 의존 구문 분석 및 형태소 분석을 이용한 대용량 학습데이터 생성 방안'에 대해 서술하고 제 4장에서는 제안한 방안에 대한 실험 및 결과에 대해 분석한다. 마지막으로, 제 5장에서는 결론 및 향후 방안에 대해 다룬다.

2. 관련연구

사람이 직접 수작업으로 학습데이터를 생성하는 것은 시간과 비용에 많은 제약이 있다. 그래서 이벤트 추출에서 수작업으로 레이블 할 때의 한계점을 해결하기 위해 위키피디아 기사를 이용해 학습데이터를 자동으로 생성한 연구가 있다[7,8]. [7]에서는 기존에 존재하는 구조화된 지식베이스로부터 해당 이벤트의 레이블을 자동으로 부여하여 학습데이터를 생성하였다. 실험 결과 많은 양의 학습데이터가 이벤트 추출에 효과적임을 보였다. [8]에서는 Freebase와 FrameNet을 이용하여 이벤트를 표현할 수 있는 문장에 자동으로 레이블을 지정하여 학습데이터를 생성하였다. 실험 결과 자동으로 레이블이 지정된 대규모 데이터의 품질이 수작업으로 정교하게 레이블 한 데이터와 경쟁할 수 있음을 보였다. 제로 대명사(zero pronoun)의 해결을 위한 감독학습에서 학습데이터를 자동으로 생성한 연구가 있다[9]. 레이블 된 학습데이터를 생성하는데 인력을 투입해야하는 어려움을 해결하기 위해 학습데이터 자동 생성 방법을 제안하였으며 실험 결과 제안 방안이 효과적임을 보였다. 이처럼 학습데이터를 자동으로 생성하여 사용하는 것은 이벤트 추출, 제로 대명사 분야에서 좋은 성능을 보이며 수작업에 비해 시간과 비용을 절약할 수 있다.

본 논문에서는 감성분석 분야에서의 학습데이터 생성을 제안하며 [10]에서는 트윗에 적용되는 감성분석에서 학습데이터를 수작업으로 레이블 할 경우의 어려움과 웹에서 직접 수집한 데이터에 존재하는 노이즈 문제를 해결하기 위해 원격 지도학습과 필터링 기술을 결합하여 학습데이터를 생성하였다. 그 결과 성능향상을 보였지만 트윗의 해시태그를 기준으로 학습데이터를 생성하기 때문에 해시태그가 존재하는 트윗에만 국한된다는 단점이 있다.

기존의 감성분석에 관한 연구는 전통적인 기계학습 기법이 많이 이용되어져 왔다. [11]에서는 트윗에서 자질을 추출한 뒤 서포트 벡터 머신과 나이브 베이즈를 이용하여 감성을 분류하고 그 성능에 대해 평가하였다. 그러나 전통적인 기계학습 기법의 경우 학습데이터의 특성에 영향이 커서 영역 적응 문제로 인해 저조한 성능을 보인다. 그래서 최근에는 딥러닝을 이용한 감성분석의 우수한 성능이 입증되면서 딥러닝을 이용한 감성분석이 많이 연구되고 있다[2].

그러나 딥러닝의 경우 학습데이터의 양과 질은 성능을 결정하는 중요한 요소이다. 그래서 딥러닝의 성능을 향상시키기 위해 학습데이터를 자동으로 생성한 연구가 있다. [12]에서는 전자상거래에서의 검색서비스 향상을 위해 딥러닝 기반 키워드 인식기 모형을 제시하였다. 그리고 SSD 모형이 학습데이터마다 직접 정답 레이블을 해야 되는 문제를 해결하기 위해 학습데이터 자동 생성 프로그램을 개발하였으며 수작업과 비교하여 시간과 비용을 대폭 절감할 수 있음을 보였다. 그리고 [13]에서는 인공신경망의 일반화 능력을 향상시키기 위해 학습데이터 생성 알고리즘을 제안하여 실험하였으며 실험 결과 추가된 학습데이터가 신경망의 일반화 능력을 크게 향상시키는 것을 확인하였다. 이처럼 딥러닝에서의 학습데이터는 중요한 요소이며 성능을 향상시키는 것을 확인할 수 있다.

주로 딥러닝을 이용한 감성분석으로는 영어 텍스트를 이용한 연구가 많이 진행되었다[14-16]. 한국어의 경우 어순의 제약 등으로 인해 영어에 비해 감성분석에 어려움이 있다.

감성분석을 위한 학습을 위해서는 긍정과 부정으로 감성이 레이블이 된 데이터가 필요하다. 감성분석을 위한 데이터 생성을 위해 2인 1조로 구성하여 사람이 직접 데이터의 감성을 레이블 한 연구가 있다[17]. 사람이 직접 감성을 레이블 할 경우 정확한 학습데이터를 생성할 수 있지만 많은 시간과 비용이 소요되고 대량의 데이터를 레이블하기엔 많은 제약이 있다. 그리고 [18,19]에서는 평점에 따라 감성을 긍정과 부정으로 분류한 네이버 영화 리뷰 데이터 셋(nsmc)[20]을 학습데이터로 사용하였다. 평점을 이용해 감성을 판단할 경우 수작업에 비해 학습데이터 생성에 시간을 절약할 수 있지만 반드시 평점과 내용의 감성이 같지 않은 경우가 많아 오류 발생 위험이 크다.

그리고 보통 감성분석의 경우 감성 어휘를 지닌 감성사전을 구축하여 그것을 자질로 감성분석이 이루어진다. 다양한 도메인에 맞추어 특성에 맞게 감성분석을 할 수 있도록 도메인 별 맞춤형 감성사전을 구축한 연구가 있다[21]. 이 경우 도메인에 맞는 감성사전을 구축하여 해당 도메인에 좋은 성능을 보일 수 있지만 감성사전을 직접 구축하는 일은 많은 비용과 시간이 소요된다. 그래서 영어로 이루어진 감성사전을 한국어로 번역하여 그것을 자질로 감성분석을 수행한 연구가 있다[22,23]. 그러나 이러한 경우 잘못된 번역으로 인한 오류가 발생할 수 있다는 단점이 있다.

이와 같이 감성이 레이블이 된 많은 양의 데이터를 구하기는 어렵다. 그래서 [24]에서는 반감독 학습을 통해 감성 레이블이 있는 데이터를 활용하여 감성 레이블이 없는 데이터의 레이블을 확정하는 방법을 사용하였다. 그러나 반감독학습의 경우 레이블이 있는 데이터의 오류가 계속 다른 데이터에 큰 영향을 줄 수 있다는 단점이 있다.

본 논문에서는 이와 같은 문제를 해결하기 위해 기존에 존재하는 감성사전과 의존 구문 분석 및 형태소 분석을 활용한 대용량 학습데이터 자동 생성 방안을 제안한다. 기존에 존재하는 감성사전을 활용함으로써 감성사전 구축을 위한 시간과 비용을 절약할 수 있으며 의존 구문 분석을 통해 한국어가 가지는 문제를 해결한 감성 레이블을 가진 대용량의 학습데이터를 생성할 수 있다.

3. 제안 방안

그림 2는 제안 방안의 흐름도이다. 한 문장에는 여러 감성이 존재할 수 있으며 이러한 경우 문장의 감성을 긍정이나 부정 하나로 레이블하기 어렵다. 그래서 본 논문에서는 문장을 절(Clause)로 나누는 방법을 제안해 절 단위로 감성을 레이블 하였다. 예를 들면, “연기는 훌륭했지만 전체적인 영상미가 좋지 않았다.”라는 문장이 있을 경우 ‘지만’을 기준으로 두 절로 나누게 된다. 문장을 절로 나누는 방법에 대해서는 3.1절에서 자세히 다룬다.

다음으로, 절을 감성사전과 매칭해서 어절 단위로 긍정 어휘에는 1점, 부정 어휘에는 -1점을 매기게 된다. 이때 감성사전의 원형과 어근을 둘 다 자질로 사용한다. 예를 들면, ‘훌륭했지만’과 ‘좋지’는 긍정을 나타내 1점을 매기게 된다. 그러나 ‘좋지’의 경우 뒤에 나오는 부정소 ‘않았다’로 인해 부정을 나타내지만 단순히 감성사전과 매칭하게 되면 긍정으로 분류되는 문제점이 발생한다. 감성사전으로 감성 점수 매기는 방법과 이러한 방법의 문제점에 대해서는 3.2절과 3.3절에서 자세히 다룬다.

감성사전으로만 감성 점수를 판단하는 것으로 인해 생기는 문제점을 해결하기 위해 감성 변환 요소가 존재하면 의존 구문 분석 및 형태소 분석을 수행해 감성 점수를 재설정하게 된다. 이때 감성 변환 요소로는 감성 반전, 감성 활성화, 감성 비활성화가 있다[25]. 감성 반전과 감성 활성화의 경우 의존 구문 분석을 통한 의존 관계를 이용하게 되고 감성 비활성화의 경우 형태소 분석을 이용한다. 의존 관계를 이용하면 한국어의 생략 및 어순 자유성으로 인한 문제점을 해결가능하다. 이러한 의존 구문 분석과 그 관계 및 형태소 분석을 통한 감성 변환 요소에 대해서는 3.4절과 3.5절에서 자세히 다룬다.

마지막으로, 위 과정을 통해 최종적으로 확정된 감성 점수를 기준으로 감성 점수가 양수이면 긍정, 감성 점수가 음수이면 부정으로 감성을 레이블 한다. 그리고 감성이 레이블 된 데이터 중 감성 점수가 높은 Top-k의 데이터를 추출해 감성분석에 적합한 대용량 학습데이터를 생성한다. 생성한 학습데이터를 Bi-LSTM으로 학습을

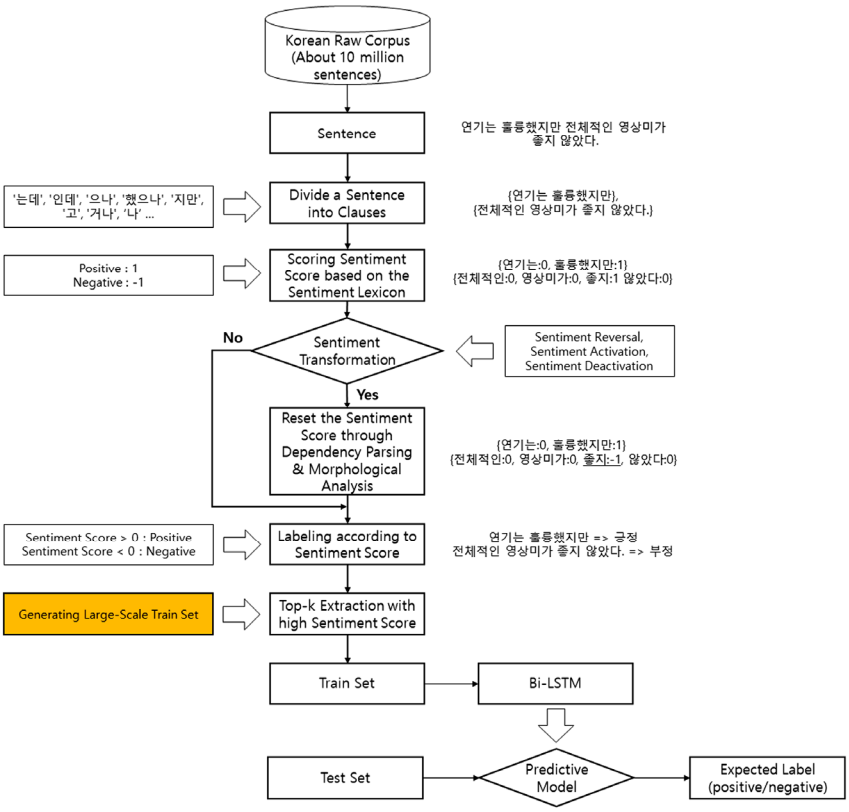

그림 2 제안 방안의 흐름도

Fig. 2 Flow chart of the proposed model

진행해 그 모델을 바탕으로 감성분석을 수행한다. 3.6절에서 감성 점수를 기준으로 감성을 레이블 하는 방법에 대해 자세히 다룬다.

3.1 입력 문장을 절로 나누는 방법

한 문장은 여러 감성을 가질 수 있으며, 여러 감성을 가지게 될 경우 그 문장의 감성을 긍정이나 부정으로 확실히 정의하기 어렵다.

예를 들면, 그림 3이 입력 문장으로 주어졌을 때 '이번 영화에서 배우들의 연기는 훌륭했지만'은 긍정 감성을 가지고, '전체적인 영상미가 아쉬웠다.' 같은 경우에는 부정 감성을 가진다고 볼 수 있다. 그렇기 때문에 문장 전체의 감성을 긍정이나 부정 하나로 정의하기 어렵다. 이러한 문제를 해결하기 위해 감성을 정의하기 전에 문장을 절로 나누는 방법을 제안한다.

이번 영화에서 배우들의 연기는 훌륭했지만 전체적인 영상미가 아쉬웠다.

그림 3 입력 문장 예시

Fig. 3 An example of an input sentence

'는데', '인데', '으나', '했으나', '지만', '고', '거나', '나', '면서', '으면서', '해서', '어서', '며', '으며', '였으며', '다면', '면', '니', '더니', '다가', '라', '든'

그림 4 형태소 목록

Fig. 4 Morpheme list

그 방법은 이와 같다. 문장을 형태소 분석을 한 후 그림 4의 목록 중에 형태소 품사 태그가 연결어미를 뜻하는 'EC'를 가지는 것을 기준으로 나눈다. 이때 형태소 분석기는 Mecab 형태소 분석기[26]를 사용하였다. 그러나 "문제점이 끊임없이 발생하고 있다."라는 문장이 있을 경우 '발생하고'에서 '고'가 'EC' 태그를 가져 "문제점이 끊임없이 발생하고"와 "있다"로 나눠지는 문제점이 생긴다. 그래서 이때 글자 수가 5개 미만인 경우에는 나누지 않고 앞 문장에 붙이도록 하였다.

3.2 감성사전을 활용한 감성 점수 매기는 방법

입력 데이터를 감성사전을 활용해 감성 점수 매기는 방법에 대해 설명한다. 이 때 감성사전으로는 'KNU 한국어 감성사전[27]'을 사용하였다.

표 1 감성사전 매칭 알고리즘 예시
Table 1 An example of sentiment lexicon matching algorithm

	Input	Morphological analysis	Sentiment lexicon matching
(1)	실망스러운 하루를 보내 우울하다.	실망/NNG+스럽/XSA+ㄴ/ETM 하루/NNG+를/JKO 보내/VV+어/EC 우울/NNG+하/XSA+다/EF+./SF	{실망스러운:-1, 하루를:0, 보내:0, 우울하다.:-1}
(2)	예쁜 꽃을 봐서 좋다.	예쁘/VA+ㄴ/ETM 꽃/NNG+을/JKO 보/VV+아서/EC 좋/VA+다/EF+./SF	{예쁜:1, 꽃을:0, 봐서:0, 좋다:1}
(3)	나는 기분이 상쾌하지 않았다.	나/NP+는/JX 기분/NNG+이/JKS 상쾌/XR+하/XSA+지/EC 않/VX+았/EP+다/EF+./SF	{나는:0, 기분이:0, 상쾌하지:1, 않았다.:0}

KNU 한국어 감성사전은 군산대학교에서 구축한 한국어 감성사전으로 특정 도메인에 영향을 받지 않는 독립적인 감성 어휘와 온라인 텍스트 데이터에서 사용되는 신조어, 이모티콘으로 이루어져 있다[28]. 긍정, 부정, 중립으로 이루어져 있으며 본 논문에서는 긍정과 부정을 나타내는 어휘만 추출하여 감성 점수를 매기는 수단으로 사용하였다.

전 단계에서 절로 나누어진 데이터를 꼬꼬마 형태소 분석기[29]를 통해 형태소 분석을 하게 된다. 그리고 감성사전의 어휘도 형태소 분석기를 통해 어근을 추출하게 된다. '순수한', '순수하게', '순수하고', '순수하다' 등의 어휘가 있을 때 이 어휘들은 모두 '순수/NNG'라는 어근을 가지며 같은 의미를 나타낸다. 감성사전 특성상 같은 어근을 가지는 모든 어휘를 담기 어렵기 때문에 어근을 추출하여 사용한다. 여기서 추출한 어근 중에 한 글자로 된 어근을 제외하였는데 예를 들어 '날조하여'라는 부정적인 어휘는 어근으로 '날/NNG'이 추출이 되는데 이러한 경우에 '소풍가는 날'에서의 '날'이 부정으로 판단된다는 오류가 있다. 이와 같은 이유로 한 글자로 이루어진 어근은 감성 어휘의 실질적인 의미를 담기 어렵다고 판단해 제외한다. 그러나 한 글자로 이루어진 어근을 감성 판단에서 제외하면 그에 해당하는 감성을 판단하기 어렵다는 문제가 발생한다. 그래서 감성사전 어휘의 원형도 감성 판단 자질로 어근과 함께 사용한다.

그림 5와 같은 알고리즘이 수행되면 입력 절을 띄어쓰기 단위로 나눈 토큰 리스트를 생성한다. 그리고 이 토큰을 감성사전 어휘의 어근과 형태소 분석한 데이터를 비교하고 감성사전 어휘의 원형과 입력 데이터를 비교해 긍정 어휘에는 1점, 부정 어휘에는 -1점을 매긴다.

예를 들면, 표 1의 (1)은 감성 어휘의 어근 중 부정을 나타내는 '실망/NNG', '우울/NNG'을 가진다. 그래서 감성사전 매칭 알고리즘 수행 결과 총 감성 점수가 -2점이 되고 부정으로 분류된다. (2)는 어근 중 긍정을 나타내는 '예쁘/VA'를 가진다. 그리고 한 글자로 이루어진 어근은 제외하였으므로 어근만으로 판단하면 긍정을 나타내는 '좋다'는 한 글자로 이루어진 어근 '좋/VA'을 가지므로 제외된다. 이러한 문제를 해결하기 위해 어휘 원형까지 같이 감성 판단 자질로 사용하였으며 '좋다'라는 어휘 원형이 사전에 긍정 어휘로 존재해 감성 점수 1점을 매기게 된다. 그래서 감성사전 매칭 알고리즘 수행 결과 총 감성 점수는 2점이 된다. (3)은 감성 어휘의 어근 중 긍정을 나타내는 '상쾌/XR'를 가진다. 그래서 감성사전 매칭 알고리즘 수행 결과 총 감성 점수가 1점이 되고 긍정으로 분류된다. 그러나 (3)같은 경우에는 부정소 '않았다'로 인해 감성이 긍정에서 부정으로 반전되어야 하지만 단순히 긍정과 부정 어휘의 빈도로만 판단하게 되면 긍정으로 분류되는 문제가 발생한다.

Algorithm 1 : 감성사전 매칭

```
T : Input clause's token list
D : Token dictionary
neg_dic : Negative sentiment dictionary
pos_dic : Positive sentiment dictionary
T[i]_w: Word of the i-th token
T[i]_mor: Morpheme of the i-th token
T[i]_id: Number of the i-th token
for i in range(len(T)):
    D[ T[i]_id ] = 0
    if ( T[i]_w or T[i]_mor ) in neg_dic:
        D[ T[i]_id ] = -1
    if ( T[i]_w or T[i]_mor ) in pos_dic:
        D[ T[i]_id ] = 1
```

그림 5 감성사전 매칭 알고리즘
Fig. 5 Algorithm of sentiment lexicon matching

3.3 감성사전 매칭 알고리즘의 문제점

단순히 감성 어휘의 빈도만을 이용해 데이터의 감성을 레이블 하는 방법의 경우 감성 변환 요소로 인한 문제점이 발생한다. 예를 들면, 표 2의 (1)은 부정을 나타내지만 '좋지'라는 긍정 어휘만 고려하고 감성사전만을 사용할 경우 뒤에 나오는 부정소를 고려하지 않아 긍정으로 분류된다. (2)의 경우 긍정을 나타내지만 '오류'는 부정, '극복'은 긍정으로 인해 중립으로 분류된다. 그리고 (3)의 경우 정도 부사 '너무'가 '많다'와 결합하여 부정을 나타내지만 감성을 가지는 어휘가 없어 중립으로 분류된다. (4)의 경우 부정을 나타내지만 '좋게'라는 긍정 어휘와 '경망한'이라는 부정 어휘로 인해 중립으로 분류된다.

표 2 감성사전 매칭 알고리즘의 문제점
Table 2 Problems of sentiment lexicon matching algorithm

	Input	Algorithm1 result	Solution label
(1)	그렇게 좋지 않은데요	positive	negative
(2)	요즘 학생들은 선배들의 오류를 많이 극복했다고 본다.	neutral	positive
(3)	사람이 너무 많다고 생각한다.	neutral	negative
(4)	그것은 아무리 좋게 생각해도 경망한 짓이었소.	neutral	negative

본 논문에서는 부정소와 특정 어근으로 인해 감성이 반전되고, 정도부사 '너무'로 인해 감성이 활성화되고, 특정 형태소로 인해 감성이 비활성화되는 유형을 찾고 문제점을 해결하고자 한다.

3.4 의존 구문 분석

구문 분석은 문장의 구조를 파악하여 문장에 포함된 단어들 간의 관계를 찾아내는 작업으로 자연어 처리에서 중요한 작업이다. 구문 분석 방법으로는 구구조 분석(Phrase Structure Parsing)과 의존 구문 분석(Dependency Parsing)이 있다. 구구조 분석은 여러 언어 요소가 모여 구문 요소를 만들고 여러 구문 요소가 모여 더 큰 구문 요소를 만드는 방법이고 의존 구문 분석은 두 언어요소 사이의 의존관계를 파악함으로써 문장을 분석하는 것이다. 한국어는 영어에 비해 어순이 자유로우므로 구구조 분석보다 의존 구문 분석을 사용하는 것이 적합하다. 의존 구문 분석을 사용하면 한국어의 어순 자유성에 의한 문제점을 해결 가능하고 구성요소의 불연속성이나 생략 등과 같은 현상에 큰 영향을 받지 않는다[30]. 의존관계는 지배소와 의존소 사이에 존재하며, 이 때 지배소는 의미의 중심이 되는 요소이고 의존소는 지배소가 갖는 의미를 보완해주는 요소이다.

그림 6은 의존 문법을 이용한 구문 구조이다. 예를 들면, "나는 밥을 먹는다."라는 문장이 있을 때 '나/NP+는/JX'은 체언(NP)이면서 주어(SBJ)를, '밥/NNG+을/JKO'은 체언(NP)이면서 목적어(OBJ)를 나타내고 '먹/VV+는다/EF+./SF'는 VP가 되어 동사구를 나타낸다[31]. 그리고 '나/NP+는/JX', '밥/NNG+을/JKO'은 '먹/VV+는다/EF+./SF'를 지배소로 가진다. 이와 같이 의존 구문 분석을 이용하면 문장의 의미와 형태를 파악하는데 도움이 된다. 본 논문에서는 한국어 의존 구문 분석에서의 지배소와 의존소 관계를 이용해 감성 반전과 감성 활성화되는 요소를 찾고 감성 점수에 적용하였다. 한국어 의존 구문 분석을 위해 한국어 형태소 및 구문 분석기 모음인 KoalaNLP[32]에서 제공하는 꼬꼬마 형태소 분석기와 의존 구문 분석기 API를 사용해 의존 구문 분석을 수행하였다.

그림 6 의존 구문 분석 예시
Fig. 6 An example of dependency parsing

3.5 감성 변환 요소

의존 구문 분석 및 형태소 분석을 이용한 감성 반전, 감성 활성화, 감성 비활성화를 통해 감성이 변환되는 경우에 대해 알아본다. 그림 7은 본 논문에서 제안하는 감성 변환 요소에 대해 나타낸 그림이다. 감성 반전은 긍정 감성이 부정으로, 부정 감성이 긍정으로 반전되는 것을 말하며 부정소와 소멸, 해소, 상실의 의미를 가진 어근으로 인해 이루어진다. 감성 활성화는 감성을 가지지 않는 중립 어휘가 감성을 가지게 되는 것으로 정도부사 '너무'로 이루어진다. 그리고 감성 비활성화는 감성 활성화와 반대로 감성을 가지던 어휘가 감성을 가지지 않는 중립 어휘가 되는 것을 말하며 특정 형태소로 인해 선행절의 감성이 비활성화 된다. 그림 8의 알고리즘 2는 의존 구문 분석에 의한 감성 반전과 감성 활성화를 나타내고 그림 14의 알고리즘 3은 특정 형태소에 의한 감성 비활성화에 대해 나타낸다.

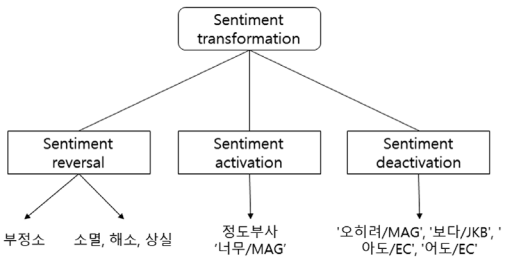

그림 7 감성 변환 요소
Fig. 7 Sentiment transformation element

3.5.1 감성 반전

감성 어휘와 결합하여 감성을 반전시키는 부정 성분에 대해 설명한다. 한국어의 부정문은 부정소의 위치에 따라 단형 부정문과 장형 부정문으로 나누며, 부정소의 쓰임에 따라 '안' 부정문과 '못' 부정문으로 나눌 수 있다[33].

단형 부정문은 "나는 수학을 안 좋아한다."와 같이 부정소가 서술어 앞에 위치하며, 장형 부정문은 "나는 수학을 좋아하지 않는다."와 같이 서술어인 용언의 어간에 연결어미 '-지'가 붙고 뒤에 부정소가 위치한다. 그리고 이와 같은 경우에는 '좋아하/VV'라는 긍정을 나타내는 어근이 있음에도 전체 문장을 보면 부정 감성을 나타낸다. 이와 같이 부정소와 감성 어휘가 수식이 되면 감성 반전이 이루어진다.

Algorithm 2 : 감성 반전, 감성 활성화

T : Input clause's token list
D : Token dictionary
neg_dic : Negative sentiment dictionary
pos_dic : Positive sentiment dictionary
$T[i]_w$: Word of the i-th token
$T[i]_{mor}$: Morpheme of the i-th token
$T[i]_{root}$: Root of the i-th token
$T[i]_{id}$: Number of the i-th token
$T[i]_{hid}$: Head's number of the i-th token

```
for x in range(len(T)):
  for y in range(len(T)):
    if ( T[i]_w or T[i]_mor ) in neg_dic:
      if ( T[i]_id == T[x]_hid and '안/MAG' in T[x]_root
      or T[i]_hid == T[x]_id and '지/EC' in T[i]_mor and T[x]_root in ('않/VX', '아니하/VX', '못/MAG', '못하/VX')
      or T[i]_hid == T[x]_id and '지/EC' in T[x]_mor and T[x]_hid == T[y]_id and T[y]_root in ('않/VX', '아니하/VX', '못/MAG', '못하/VX')
      or T[i]_hid == T[x]_id and T[x]_root in ('없/VG', '사라지/VV', '해결/NNG', '극복/NNG', '풀리/NNG')):
        D[ T[i]_id ] = 1
    elif ( T[i]_w or T[i]_mor ) in pos_dic:
      if ( T[i]_id == T[x]_hid and '안/MAG' in T[x]_root
      or T[i]_hid == T[x]_id and '지/EC' in T[i]_mor and T[x]_root in ('않/VX', '아니하/VX', '못/MAG', '못하/VX')
      or T[i]_hid == T[x]_id and '지/EC' in T[x]_mor and T[x]_hid == T[y]_id and T[y]_root in ('않/VX', '아니하/VX', '못/MAG', '못하/VX')
      or T[i]_hid == T[x]_id and T[x]_root in ('없/VG', '사라지/VV', '어렵/VA', '떨어지/VV', '상실/NNG', '꺾이/VV', '힘들/VA', '낮/VA', '잃/VV')):
        D[ T[i]_id ] = -1
    else:
      if ( T[i]_id == T[x]_hid and '너무/MAG' in T[x]_root ):
        D[ T[i]_id ] = -1
```

그림 8 감성 반전, 감성 활성화 알고리즘
Fig. 8 Algorithm of sentiment reversal and sentiment activation

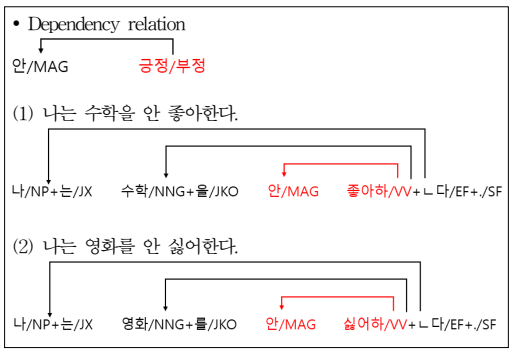

그림 9 감성 반전 예시 1
Fig. 9 An example of sentiment reversal 1

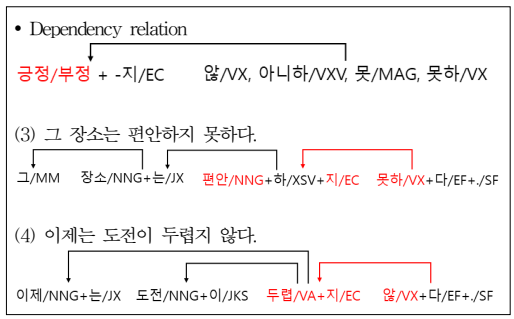

그림 10 감성 반전 예시 2
Fig. 10 An example of sentiment reversal 2

그림 9는 단형 부정소에 의한 감성 반전으로 부정소 '안/MAG'이 감성 어휘를 지배소로 가지면 감성 어휘의 감성이 반전된다. 예를 들면, (1)의 경우 부정소 '안/MAG'이 긍정 감성을 가지는 '좋아하/VV'를 지배소로 가지게 되어 감성이 긍정에서 부정으로 반전되고 (2)의 경우에는 '안/MAG'이 부정 감성을 가지는 '싫어하/VV'를 지배소로 가지게 되어 감성이 부정에서 긍정으로 반전된다.

그림 10은 장형 부정소에 의한 감성 반전으로 감성을 가지는 용언의 어간에 연결어미 '지/EC'가 붙고 부정소를 지배소로 가지면 용언의 어간의 감성이 반전된다. 예를

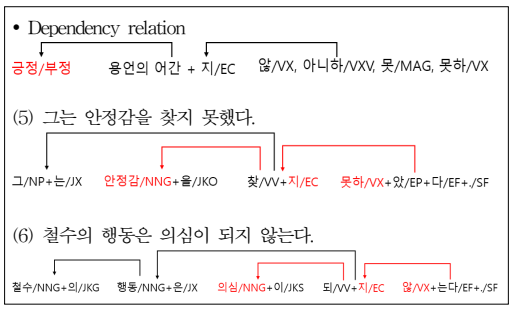

그림 11 감성 반전 예시 3

Fig. 11 An example of sentiment reversal 3

들면, (3)의 경우 긍정 감성을 가지는 '편안/NNG'에 연결어미 '지/EC'가 붙고 부정소 '못하/VX'를 지배소로 가지게 되어 감성이 긍정에서 부정으로 반전된다. 그리고 (4)의 경우에는 부정 감성을 가지는 '두렵/VA'에 연결어미 '지/EC'가 붙고 부정소 '않/VX'을 지배소로 가지게 되어 감성이 부정에서 긍정으로 반전된다.

그림 11은 장형 부정소에 의한 또 다른 감성 반전으로 그림 10과 달리 감성 어휘의 용언인 지배소의 어간에 연결어미 '지/EC'가 붙게 된다. 그리고 그 용언이 부정소를 지배소로 가지면 감성 어휘의 감성이 반전된다. 예를 들면, (5)의 경우 긍정 감성을 가지는 '안정감/NNG'이 연결어미 '지/EC'를 가지는 용언 '찾/VV+지/EC'를 지배소로 가지게 되고 이 용언은 부정소 '못하/VX'를 지배소로 가지게 되어 감성이 긍정에서 부정으로 반전된다. 그리고 (6)의 경우 부정 감성을 가지는 '의심/NNG'이 연결어미 '지/EC'를 가지는 '되/VV+지/EC'를 지배소로 가지게 되고 이 용언은 부정소 '않/VX'을 지배소로 가지게 되어 감성이 부정에서 긍정으로 반전된다.

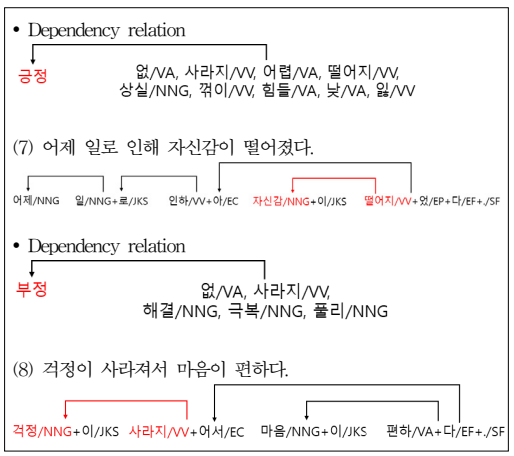

그림 12 감성 반전 예시 4

Fig. 12 An example of sentiment reversal 4

그림 12는 본 논문에서 제안하는 감성 어휘와 결합했을 경우 감성이 반전되는 어근으로 소멸, 해소, 상실의 의미를 가지고 있으며 감성 어휘가 이와 같은 어근을 지배소로 가질 경우 감성이 반전된다. 예를 들면, (7)의 경우 긍정 감성을 가지는 '자신감/NNG'이 '떨어지/VV'를 지배소로 가져 긍정에서 부정으로 감성이 반전된다. (8)의 경우 부정 감성을 가지는 '걱정/NNG'이 '사라지/VV'를 지배소로 가져 부정에서 긍정으로 감성이 반전된다.

3.5.2 감성 활성화

감성을 가지지 않는 어휘가 감성을 가지게 되어 감성이 활성화되는 경우에 대해 설명한다. 그 예로 정도부사 '너무'가 있다.

정도부사 '너무'는 이중적 의미를 가지고 있다. 정도부사 '너무'의 본래 의미는 뒤에 이어지는 성분에 대한 정도의 지나침으로 부정적 의미를 나타내며 그와 반대로 긍정 강조의 의미로도 쓰인다[34]. 본 논문에서는 정도부사 '너무'가 부정적 의미로 쓰여 감성을 가지지 않는 중립 어휘가 부정 감성을 가지게 되는 경우에 대해 알아본다.

'너무'가 문장에서 부정적으로 쓰이면 어떠한 기준에 대해 '지나친 나머지 오히려 바람직하지 못함'의 의미를 나타낸다. 예를 들면, 그림 13의 (1)은 사람이 앉아 있기에 적절한 정도가 기준이 되고 이 기준을 미치지 못해 '너무'라는 말로 이와 같은 상태를 강조하고 있으며 부정의 감성을 가지게 된다. 또한 정도부사 '너무'는 후행절을 부정적 의미로 이끄는 내용을 함축하고 있다. 예를 들면, (2)의 경우 '너무'가 후행 용언의 상태가 지나침을 강조하면서 생략된 후행절에 '그래서 걱정이다'라는 부정적인 의미를 함축하고 있다. 그러므로 정도부사 '너무'는 정도성을 나타내는 동시에 선행절이 지나친 나머지 부정적 결과를 초래하게 된다는 의미를 지니게 되어 두 가지 기능을 한다고 볼 수 있다[35].

이와 같이 정도부사 '너무'로 인해 부정적인 감성을 가지게 되는 어휘도 의존 구문 분석을 통해 파악이 가능

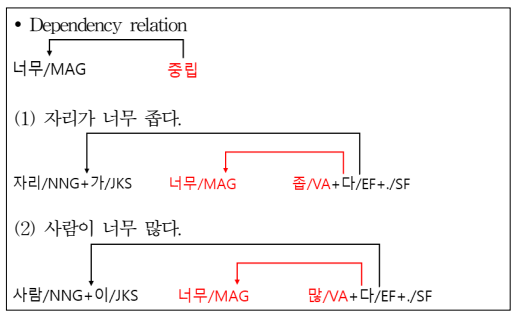

그림 13 감성 활성화 예시

Fig. 13 An example of sentiment activation

하다. (1)은 정도부사 '너무'가 '좁다'를 지배소로 가지는 구조이며 이때 지배소인 '좁다'가 부정 감성을 가지게 되고 (2)는 '너무'가 '많다'를 지배소로 가지는 구조이며 이때 지배소인 '많다'가 부정 감성을 가지게 된다. 본 논문에서는 정도부사 '너무'의 지배소가 감성사전의 긍정 어휘와 부정 어휘 둘 다 해당되지 않을 경우 부정으로 판단하도록 하였다.

3.5.3 감성 비활성화

정도부사 '너무'로 인해 감성을 가지지 않는 어휘가 감성을 가지게 되는 경우가 있는 반면 그 반대의 경우인 감성을 가진 어휘의 감성이 비활성화 되는 경우도 있다. 그림 14의 알고리즘을 적용할 경우 특정 형태소에 의해 선행절의 감성이 비활성화 된다.

예를 들면, 그림 15의 (1)은 '도가 넘치는 칭찬'의 경우 긍정 어휘로 인해 긍정으로 분류되지만 감성이 비활성화 되고 '오히려/MAJ'의 후행절인 '위험하다'의 감성만 판단해 부정으로 분류한다. (2)는 '귀엽다는 생각이 들기보다는'의 경우 긍정 어휘로 인해 긍정으로 분류가 되지만 감성이 비활성화 되고 '보다/JKB'의 후행절인 '불쌍하다는 생각이 들었다'의 감성만 판단해 부정으로 분류한다. (3)은 '그것은 아무리 좋게 생각해도'의 경우 긍정 어휘로 인해 긍정으로 분류되지만 감성이 비활성화 되고 '아도/EC'의 후행절인 '경망한 짓이었소'의 감성만 판단해 부정으로 분류한다.

Algorithm 3 : 감성 비활성화

```
    T : Input clause's token list
    D : Token dictionary
    T[i]_mor: Morpheme of the i-th token
    T[i]_id: Number of the i-th token
1:  if T[i]_mor in ('오히려/MAG', '보다/JKB', '아도/EC',
    '어도/EC') :
2:      deact = int( T[i]_id)
3:      for z in range(len(T)):
4:          if int( T[i]_id) <= deact :
5:              D[ T[i]_id] = 0
```

그림 14 감성 비활성화 알고리즘

Fig. 14 Algorithm of sentiment deactivation

(1) 도가 넘치는 칭찬은 오히려 위험하다.
• 도/NNG+가/JKS 넘치/VV+는/ETM 칭찬/NNG+은/JX
오히려/MAJ 위험/NNG+하/XSV+다/EF+./SF

(2) 귀엽다는 생각이 들기보다는 불쌍하다는 생각이 들었다.
• 귀엽/VA+다는/ETM 생각/NNG+이/JKS
들/VV+기/ETN+보다/JKB+는/JX
불쌍/XR+하/XSA+다는/ETM 생각/NNG+이/JKS
들/VV+었/EP+다/EF+./SF

(3) 그것은 아무리 좋게 생각해도 경망한 짓이었소.
• 그것/NP+은/JX 아무리/MAG 좋/VA+게/EC
생각/NNG+하/XSV+아도/EC 경망/NNG+하/XSV+ㄴ/ETM
짓/NNG+이/VCP+었/EP+소/EF+./SF

그림 15 감성 비활성화 예시

Fig. 15 An example of sentiment deactivation

3.6 감성 점수에 따른 감성 레이블

감성 점수에 따른 데이터의 감성을 레이블하는 방법에 대해 설명한다. 그림 16의 알고리즘과 같이 3.4절의 감성사전, 3.5절의 의존 구문 분석 및 형태소 분석을 통해 어절 별로 매겨진 점수를 모두 합하게 된다. 합해진 이 점수가 데이터의 감성 점수가 되며 감성 점수가 양수일 경우 긍정으로, 음수일 경우 부정으로 그 문장의 감성을 레이블한다.

이와 같은 과정을 통해 긍정과 부정으로 감성이 레이블 되며 감성 점수의 절댓값이 큰 Top-k의 절을 추출해 감성분석의 학습데이터로 사용한다.

Algorithm 4 : 감성 레이블

```
    Input : Token dictionary D, Sentiment Score S
    Output : Labeled data according to sentiment score
1:  for v in D.values():
2:      S = S + v
3:  if S > 0:
4:      positive
5:  elif S < 0:
6:      negative
```

그림 16 감성 레이블 알고리즘

Fig. 16 Algorithm of sentiment labeling

4. 실험

4.1 실험 환경

본 논문에서는 실험을 위해 네이버 카페 "한국 형태소 분석 등 NLP 연구개발 자료"에서 제공하는 1억2천3백만 가량의 단어로 구성된 한국어 원시 말뭉치(약1천만 문장)를 사용하였다. 그리고 이 말뭉치를 제안한 방법을 이용해 절로 나누어 사용했으며 그 통계는 표 3과 같다.

학습을 위한 실험 환경으로 batch size는 50, learning rate는 0.0001로 설정하였고 epoch은 12로 설정하였다. 그리고 최근 여러 딥러닝 연구에서 좋은 성능을 보이고 있는 adam 최적화 알고리즘[36]을 이용하여 파라미터를 최적화하였다. 실험을 위한 모든 모델은 구글에서 오픈소스로 공개한 라이브러리인 TensorFlow를 사용하여 구현하였다. 표 4는 실험에 사용한 컴퓨터의 사양을 나타낸다.

표 3 문장과 절의 개수

Table 3 Number of sentence and clause

Sentence	Clause
10,409,409	12,462,448

표 4 컴퓨터 사양
Table 4 Computer specification

OS	Ubuntu 16.04
CPU	Intel(R) Core(TM) i7-4790 CPU @ 3.60GHz
GPU	GeForce GTX TITAN X
RAM	31GB
HDD	901GB

4.2 학습데이터

표 5는 12,462,448개의 절에서 제안 방안을 통해 긍정과 부정으로 감성이 레이블 된 절에 대한 통계이다. 이 중 감성 점수가 높은 순대로 긍정과 부정 각 150,000개씩 300,000개를 추출하여 학습데이터로 사용하였다.

표 5 감성이 레이블 된 절의 수
Table 5 Number of clauses labeled sentiment

Sentiment	Number	Ratio
Positive	2,323,619	18.6%
Negative	1,994,709	16%

표 6은 감성 점수를 기준으로 추출한 학습데이터 문장의 예시이다. 추출된 학습데이터를 보면 비교적 긍정과 부정 감성에 맞는 데이터가 추출된 것을 볼 수 있다. 그리고 (1)의 경우 '안전', (2)의 경우 '재능'이라는 긍정 어휘가, (4)의 경우 '신경증'이라는 부정 어휘가 반복적으로 나타나 감성 점수가 높게 나왔다고 판단한다.

4.3 감성사전 자질 추출

표 7은 감성사전 자질 추출에 대해 평가한 결과이다. 감성사전 자질 중 어근만 추출하여 사용한 모델은 한 글자로 이루어진 어근으로 인한 오류가 발생한다. 그래서 그 분류 오류를 해결한 한 글자 어근을 제외한 모델에서 5.67%의 정확도 향상을 보였다. 그리고 한 글자로 된 어근으로 인해 제외된 감성 어휘를 보완하기 위해 감성사전 어휘의 원형을 추가한 모델에서 1.67%의 정확도 향상을 보여 제안한 감성사전에서 자질을 추출하는 방법의 타당성을 입증하였다.

표 7 감성사전 자질에 따른 성능 비교
Table 7 Comparison of sentiment lexicon feature

Sentiment lexicon feature	Accuracy
Root	0.8183
Root expect for 1-letter root	0.875
Root expect for 1-letter root + Word	0.8917

4.4 실험 결과

제안한 방안의 성능을 평가하기 위해 수작업과 비교를 진행하였다. 수작업은 전체 데이터 중 무작위로 15,000개를 추출해 그 데이터를 긍정과 부정으로 분류하였다. 3명이 분류에 참여하였으며 3명의 평가자 중 2명 이상이 동일한 감성으로 판단할 경우 긍정과 부정을 결정하였다. 표 8은 수작업으로 감성을 분류한 통계이다. 긍정과 부정 각 2,200개 씩 4,400개를 추출해 3,800개는 학습에 사용하였고 600개는 평가데이터로 이용하였다.

표 8 수작업 통계
Table 8 Manual labeling statistics

Sentiment	Number	Ratio
Positive	3,785	25.2%
Negative	3,620	24.1%

4.4.1 학습데이터 생성 시간

표 9는 수작업과 제안 모델의 학습데이터 생성에 소요된 절 당 평균 소요시간을 초단위로 나타낸 표이다. 수작업은 15,000개에서 3,800개의 학습데이터를 생성했으며 15,000개를 기준으로 3명의 평가자가 걸린 시간의

표 6 학습데이터 예시
Table 6 An example of a train set

Sentiment		Train set	Sentiment score
Positive	(1)	어린이안전과 노인안전 교통안전 지역안전 화재안전 수난안전 관광안전 스포츠안전 산악안전 사업장안전 등 10개 항목이다.	9
	(2)	재능기부 봉사 하려면 재능기부 봉사는 다른 사람에게 자신의 재능을 나눠준다는 점에서 매력적이다.	8
	(3)	이성과 지혜를 발달시켰으며 충분히 발달한 개인으로서 자연과 새로운 조화를 이루게 되었다고 한다.	8
Negative	(4)	신경증의 종류도 불안신경증을 비롯해 히스테리아 공포신경증 신경증성 우울증 신경증성 장애 등을 포함하고 있다.	-9
	(5)	활기를 잃어버린 일상 특히 실패 혹은 힘이 줄어드는 은퇴 시기에 대책 없이 떨어지는 허무는 당황스럽고 무겁다.	-8
	(6)	그 결과 정신적 소외감 고립감 신경질환 정신병 온갖 형태의 자기 분열과 자기 상실 속에서 고통받고 있습니다.	-8

표 9 학습데이터 생성 시간
Table 9 Time of generating train set

Model	Average time per clause (second)
Manual labeling	5.931
Proposed model	0.00095

평균을 이용해 나타내었다. 제안 모델은 전체 문장에서 추출한 절의 수인 12,462,448개에서 300,000개의 학습데이터를 생성하였으며 전체 절의 개수를 기준으로 전체 데이터의 감성 레이블에 소요된 시간을 이용해 나타내었다. 실험 결과, 제안 모델이 수작업보다 약 6,243배 빨리 학습데이터를 생성할 수 있음을 알 수 있다.

4.4.2 정확도

표 10은 수작업으로 추출한 평가데이터 600개를 기준으로 정확도를 나타낸 것이다. 정확도를 구하는 식은 아래와 같다. 수식에서 $|Solution \cap Prediction|$은 정답 레이블과 예측한 레이블이 일치하는 개수를 나타내고 $|Test\ Set|$은 평가데이터의 개수를 나타낸다.

$$Accuracy = \frac{|Solution \cap Prediction|}{|Test\ Set|}$$

수작업의 경우 사람이 직접 감성을 판단해 정확한 학습데이터를 생성할 수 있었지만 생성하는데 많은 시간이 소요되어 많은 양의 학습데이터를 생성하는데 한계가 있었다. 또한 적은 양의 학습데이터를 딥러닝 모델에 적용하였기 때문에 다른 모델에 비해 정확도가 높지 못한 결과를 보였다. 감성사전만을 사용한 모델은 감성사전 어휘의 빈도를 이용해 많은 양의 학습데이터를 생성하고 학습시켜 수작업에 비해 높은 정확도를 보였지만 감성 반전, 감성 활성화, 감성 비활성화와 같은 감성 변환 요소가 고려되지 않았다. 여기에서 발생하는 문제를 해결하기 위해 의존 구문 분석 및 형태소 분석을 통해 감성 변환 요소를 고려한 제안 모델의 경우 정확도가 9% 상승하여 제안한 모델의 타당성을 입증하였다.

표 10 테스트 정확도
Table 10 Accuracy of the test set

Model	Accuracy
Manual labeling	0.5181
Sentiment lexicon	0.8017
Proposed model	0.8917

표 11은 제안 모델이 평가데이터에서 올바르게 감성을 예측한 데이터의 예시이다. 긍정에서의 (1), (2), (3)과 부정에서의 (9), (10), (11)의 경우 본 논문에서 제안한 감성 반전이 반영됨을 볼 수 있다. (12)의 경우 감성 활성화가 반영되어 부정으로 잘 분류되었으며 (4)의 경우 정도부사 '너무'가 있지만 긍정 어휘로 인해 긍정으로 잘 분류된 것을 볼 수 있다. 긍정에서의 (7)과 부정에서의 (13), (14)의 경우 감성 비활성화가 반영되어 감성을 제대로 분류한 것을 볼 수 있다. 그리고 (7), (8), (15), (16)의 경우 감성 어휘의 빈도로 감성이 분류되었음을 볼 수 있다.

4.4.3 학습데이터 수에 따른 성능 비교

그림 17은 제안한 모델에서의 학습데이터 수에 따른 정확도를 그래프로 나타낸 것이다. 학습데이터 수가 증가함에 따라 딥러닝 모델의 성능이 향상됨을 볼 수 있다. 이를 통해 본 논문에서 제안한 방안인 딥러닝 모델을 위한 대용량 학습데이터를 생성하는 것은 성능을 향상시킬 수 있으며 제안 방안의 필요성과 우수성을 확인할 수 있다.

표 11 제안 모델의 결과
Table 11 Results of the proposed model

Sentiment	Test set	
Positive	(1)	걱정이 없어졌으니 말야
	(2)	요즘 학생들은 선배들의 오류를 많이 극복했다고 본다.
	(3)	서울이 이 모든 문제점을 한꺼번에 해결할 수 있는 선수 영입을 앞두고 있다.
	(4)	나는 이 소식을 들으니 너무 기뻤다.
	(5)	한편 적극적으로 인터뷰에 임한 나씨는 편견이나 불편한 시선에도 한국이 좋다고 말한다.
	(6)	이상하게도 어렸을 때부터 한국에 매력을 느꼈다.
	(7)	다리 위에서 기념품을 파는 노점이나 거리의 악사를 구경하는 것도 흥미롭다.
	(8)	풀타임을 뛰던 반게임을 하던 나이 든 선수의 경험은 경기에 큰 도움이 된다.
Negative	(9)	패싱 게임도 원활하지 않다.
	(10)	팀이 부진에 빠지면서 코일 감독 교체가 거론될 정도로 팀 분위기도 좋지 않다.
	(11)	올해는 그같은 이변을 기대하기 어려워 보인다.
	(12)	사람이 너무 많다고 생각한다.
	(13)	리더가 아홉 번 잘해도 한 번 실수하면 그대로 끝장일 수 있다.
	(14)	그것은 아무리 좋게 생각해도 경망한 짓이었소.
	(15)	그러나 법에도 없는 그런 모욕을 당하다는 것은 너무나 마음 아픈 일이옵니다.
	(16)	도대체 무엇을 믿어야 할지 혼란스럽다.

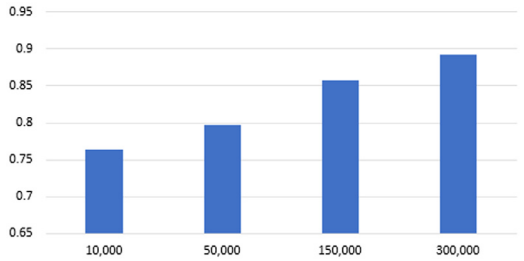

그림 17 학습데이터 수에 따른 성능 비교

Fig. 17 Performance comparison for the number of the train set

5. 결론 및 향후 연구

본 논문에서는 딥러닝을 이용한 감성분석에서의 학습데이터 부족 문제 해결을 위해 기존에 존재하는 감성사전과 의존 구문 분석 및 형태소 분석을 이용한 대용량 학습데이터 자동 생성 방안을 제안하였다. 감성사전으로는 다양한 도메인에 적용할 수 있도록 도메인에 영향을 받지 않는 'KNU 한국어 감성사전'을 사용하였다. 기존의 감성사전을 활용하여 의존 구문 분석에서의 의존 관계 및 형태소 분석을 통해 감성 변환 요소를 찾아 감성 점수를 계산하여 감성 점수에 따라 데이터를 긍정과 부정 감성으로 레이블 하였다. 이를 통해 딥러닝에서 사용 가능한 감성이 레이블 된 대용량의 학습데이터를 생성할 수 있었다. 또한 딥러닝에서의 학습데이터 수가 증가함에 따라 성능이 향상됨을 확인하고 수작업 및 감성사전만을 사용한 모델과 제안 모델을 비교하여 성능평가를 수행하였다.

본 연구에서는 기존의 감성사전을 활용해 감성분석에서의 감성사전 구축을 위한 비용과 시간을 절약할 수 있었다. 그리고 감성사전 어휘의 빈도만을 사용한 모델에 감성 반전, 감성 활성화, 감성 비활성화와 같은 감성 변환 요소를 고려해 보다 정확한 학습데이터를 자동으로 생성할 수 있었고 수작업에 비해 짧은 시간에 대용량의 감성이 레이블 된 학습데이터를 생성하였다. 제안 모델을 감성사전만을 사용한 모델과 비교함으로써 정확도 향상을 확인하였으며 학습데이터 수가 많아짐에 따라 딥러닝의 성능이 향상됨을 보여 제안 방안의 필요성과 우수성을 입증하였다.

향후에는 추가적인 감성 변환 요소와 구문 분석 및 의미 분석에 대한 결과를 추가하여 보다 정교한 대용량 학습데이터를 자동으로 생성할 것이다. 예를 들면, '슬프다'라는 어휘는 보통 부정의 의미로 쓰이지만 영화평에서는 부정의 의미로 쓰이지 않는다. 이와 같이 도메인에 의해 감성이 변환되는 경우나 정도 부사에 의해 감성이 강화, 약화되는 경우에 대해서도 탐구하여 성능을 개선할 예정이다. 또한 KNU 한국어 감성사전 외에 다른 한국어 감성사전을 활용하거나 Bi-LSTM 외에 다른 딥러닝 모델을 사용하여 비교 평가를 수행할 예정이다.

References

[1] S. Park and B. On, "Latent topics-based product reputation mining," *Journal of Intelligence and Information Systems*, Vol. 23, No. 2, pp. 39-70, 2017. (in Korean)

[2] S, Seo and J. Kim, "Sentiment Analysis Research Trend Based on Deep Learning," *The Korea Multimedia Society*, Vol. 20, No. 3, pp. 8-22, 2016. (in Korean)

[3] H. Koo, "Artificial Intelligence and Deep Learning Treads," *The Korean Institute of Electrical Engineers*, Vol. 67, No. 7, pp. 7-12, 2018. (in Korean)

[4] M. Choi and B. On, "A Method of Constructing Large-Scale Train Set Based on Sentiment Lexicon for Improving the Accuracy of Deep Learning Model," *Proc. of the 30th Annual Conference on Human & Cognitive Language Technology*, pp. 106-111, 2018. (in Korean)

[5] nlpkang [Online]. Available: https://cafe.naver.com/nlpkang/17 (downloaded 2018, Sep)

[6] S. Kim and N. Kim, "A Study on the Effect of Using Sentiment Lexicon in Opinion Classification," *Journal of Intelligence and Information Systems*, Vol. 20, No. 1, pp. 133-148, 2014. (in Korean)

[7] Y. Zeng, Y. Feng, R. Ma, Z. Wang, R. Yan, C. Shi, and D. Zhao, "Scale Up Event Extraction Learning via Automatic Training Data Generation," *Proc. of the Thirty-Second AAAI Conference on Artificial Intelligence*, Vol. 32, pp. 6045-6052, 2018.

[8] Y. Chen, S. Liu, X. Zhang, K. Liu, and J. Zhao, "Automatically Labeled Data Generation for Large Scale Event Extraction," *Proc. of the 55th Annual Meeting of the ACL*, pp. 409-419, 2017.

[9] T. Liu, Y. Cui, Q. Yin, W. Zhang, S. Wang, and G. Hu, "Generating and exploiting large-scale pseudo training data for zero pronoun resolution," *Proc. of the 55th Annual Meeting of the ACL*, pp. 102-111, 2017.

[10] S. Cagnoni, P. Fornacciari, J. Kavaja, M. Mordonini, and A. Poggi, A. Solimeo and M. Tomaiuolo, "Automatic creation of a large and polished training set for sentiment analysis on Twitter," *Proc. of the Third International Conference on Machine Learning, Optimization, and Big Data*, pp. 146-157, 2017.

[11] J. Lim and J. Kim, An Empirical Comparison of Machine Learning Models for Classifying Emotions in Korean Twitter," *Journal of Korea Multimedia Society*, Vol. 17, No. 2, pp. 232-239, 2014. (in Korean)

[12] K. Kim, W. Oh, G. Lim, E. Cha, M. Shin, and J. Kim, "The way to make training data for deep learning model to recognize keywords in product catalog image at E-commerce," *Journal of Intelli-*

gence and Information Systems, Vol. 24, No. 1, pp. 1-23, 2018. (in Korean)
[13] T. Kim and S. Kim, "Training Data Creating Algorithm to Increase Generalization Capability of Neural Networks," *Proc. of the KIISE 2014 Winter Conference*, Vol. 2014, No. 12, pp. 1281-1283, 2014. (in Korean)
[14] H. Shirani-Mehr, "Application of Deep Learning to Sentiment Analysis of Movie Reviews," *Technical Report*, Stanford University, 2014.
[15] H. Pouransari and S. Ghili, "Deep learning for sentiment analysis of movie reviews," *Technical Report*, Stanford University, 2015.
[16] Y. Wang, M. Huang, L. Zhao, and X. Zhu, "Attention-based LSTM for Aspect-level Sentiment Classification," *Proc. of the 2016 Conference on Empirical Methods in Natural Language Processing*, pp. 606-615, 2016.
[17] Y. Park, S. Kwak, D. Lee, B. Kim, Y. Yoon, and J. Lee, "Construction of Korean Test Collection for Social Media Text Sentiment Analysis," *Proc. of the KIISE 2012 Fall Conference*, Vol. 39, No. 2B, pp. 118-120, 2012. (in Korean)
[18] Y. Oh, M. Kim, and W. Kim, "Korean Movie-review Sentiment Analysis Using Parallel Stacked Bidirectional LSTM Model," *Journal of KIISE*, Vol. 46, No. 1, pp. 45-49, 2019. (in Korean)
[19] M. Kim, J. Byun, C. Lee, and B. On, "Multi-channel CNN for Korean Semtiment Analysis," *Proc. of the 30th Annual Conference on Human & Cognitive Language Technology*, pp. 79-83, 2018. (in Korean)
[20] L. Park. Naver sentiment movie corpus v1 [Online]. Available: https://github.com/e9t/nsmc
[21] D. Kim, T. Cho, and J. Lee, "A Domain Adaptive Sentiment Dictionary Construction Method for Domain Sentiment Analysis," *Proc. of the Korean Society of Computer Information Conference*, Vol. 23, No. 1, pp. 15-18, 2015. (in Korean)
[22] H. Seo, H. Kim, and J. Kim, "Sentiment Polarity Identification of Comments Using Machine Learning," *Proc. of the Korean Society of Marine Engineering Conference*, pp. 373-374, 2009. (in Korean)
[23] J. Hwang and Y. Ko, "A Korean Sentence and Document Sentiment Classification System Using Sentiment Features," *Journal of KIISE : Computing Practices and Letters*, Vol. 14, No. 3, pp. 336-340, 2008. (in Korean)
[24] S. Hong, Y. Chung, and J. Lee, "Semi-supervised learning for sentiment analysis in mass social media," *Journal of Korean Institute of Intelligent Systems*, Vol. 24, No. 5, pp. 482-488, 2014. (in Korean)
[25] G. Yoo and J. Nam, "A Study on Polarity Change Context for Sentiment Analysis of Social Network Corpus," *Proc. of the 32th Korean Association For Lexicography Conference*, pp. 133-143, 2018. (in Korean)
[26] Mecab [Online]. Available: https://bitbucket.org/eunjeon/mecab-ko
[27] park1200656. KnuSentiLex [Online]. Available: https://github.com/park1200656/KnuSentiLex
[28] S. Park, C. Na, M. Choi, D. Lee, and B. On, "KNU Korean Sentiment Lexicon - Bi-LSTM-based Method for Building a Korean Sentiment Lexicon -," *Journal of Intelligence and Information Systems*, Vol. 24, No. 4, pp. 219-240, 2018. (in Korean)
[29] D. Lee, J. Yeon, I. Hwang, and S. Lee, "KKMA : A Tool for Utilizing Sejong Corpus based on Relational Database," *Journal of KIISE : Computing Practices and Letters*, Vol. 16, No. 11, pp. 1046-1050, 2010. (in Korean)
[30] Y. Lee and J. Lee, "Korean Dependency Parsing Using Online Learning," *Proc. of the KIISE Korea Computer Congress 2010*, Vol. 37, No. 1C, pp. 299-304, 2010. (in Korean)
[31] H. Kim, J. Lee, S. Lee, Y. Han, and J. Cha, "Efficient Parsing for Head final Language," Changwon National University, 2013. (in Korean)
[32] nearbydelta. KoalaNLP [Online]. Available: https://github.com/nearbydelta/KoalaNLP
[33] H. Park, "A Study on the Meaning Differentiation of Negatives in Korean Language," *Journal of CheongRam Korean Language Education*, Vol. 42, pp. 523-551, 2010. (in Korean)
[34] G. Lim, "On the characteristic of the word formation and co-occurrence of Korean degree adverb 'neomu'," *URIMALGEUL : The Korean Language and Literature*, pp. 77-100, 2004. (in Korean)
[35] H. Ko, "The Relations between Meaning and Co-occurrence Characteristic of the Adverb of Degree 'Neomu'," *The Journal of Language & Literature*, Vol. 46, pp. 121-139, 2011. (in Korean)
[36] D. Kingma and J. Ba, "Adam: A method for stochastic optimization," arXiv preprint arXiv:1412.6980, 2014.

최 민 성
2016년~현재 군산대학교 소프트웨어융합공학과 재학중. 관심분야는 자연어 처리, 인공지능

온 병 원
2007년 미국 펜실베이니아주립대학교의 컴퓨터공학과 박사, 캐나다 브리티시컬럼비아 대학교 박사후연구원, 2010년 미국 일리노이대학교 ADSC센터 선임연구원, 서울대학교 차세대융합기술연구원 연구교수. 현재 군산대학교 소프트웨어융합공학과 부교수. 관심분야는 데이터 마이닝, 정보검색, 빅데이터, 인공지능