https://doi.org/10.24303/lakdoi.2017.25.4.193

소셜 미디어 텍스트의 미분석어 처리를 위한 전처리기 및 사전확장 연구*

최성용 · 신동혁 · 남지순**
(한국외국어대)

Choi, Seong-Yong, Shin, Dong-Hyok & Nam, Jeesun. (2017). A methodology for building linguistic resources that recognize unanalyzed sequences in social media texts. *The Linguistic Association of Korea Journal, 25*(4), 193-226. This study aims to analyze linguistic problems with unanalyzed tokens of Social Media (SM) texts and to propose methodologies for dealing with them effectively. Recently, with SM users on the rise, the need for analyzing such texts has significantly increased. However, the unanalyzed tokens severally hamper the overall performance of processing SM textual data. This study proposes two methodologies: 1) a normalizing process with a preprocessing module named Preprocessing Grammar Table (PGT) to correct frequent unanalyzed sequences such as orthographic errors and space errors; 2) a lexicon-based method utilizing DECO dictionary and Local Grammar Graph (LGG). By applying PGT and an enhanced DECO dictionary to SM texts, preprocessing performance considerably improves with 87% of the unanalyzed tokens removed, which reveals the significance of the research.

주제어(Key Words): 전처리(preprocessing), 소셜 미디어(social media), 미분석어(unanalyzed tokens), 전처리 문법 테이블(PGT), 데코사전(DECO dictionary), 부분 문법 그래프(local grammar graph).

* 본 연구는 한국외국어대학교 2017년도 교내 학술연구지원에 의해 수행되었음. SNS 텍스트에서 관찰되는 언어학적 양상에 대한 연구세미나에서 한국외국어대학교 디코라연구센터(http://dicora.hufs.ac.kr)의 한재호, 황창회, 유광훈 연구원의 분석 및 토론이 큰 도움이 되었음을 밝히며 이 자리를 빌어 감사의 마음을 전한다. 이 논문에 대한 익명의 세 분 심사위원께도 감사의 말씀을 드린다.

** 제1저자: 최성용, 교신저자: 남지순

1. 머리말

본 연구는 현대의 소셜 미디어(social media) 텍스트에서 높은 비중으로 출현하는 미분석어(unanalyzed token)의 유형을 언어학적으로 분석하고 이 유형들을 효과적으로 처리하기 위한 방법론을 제안하는 것을 목적으로 수행되었다. 소셜 미디어는 통상적으로 온라인상에서 이루어지는 다양한 커뮤니케이션의 총합을 말한다. 현재 전 세계는 소셜 미디어를 이용한 활발한 소통이 이루어지고 있다. 각종 온라인 블로그, 온라인 포럼, 신문기사 댓글, 그리고 사용자 후기글 리뷰 등의 다양한 형식을 통해 사용자 자신의 의사표현이 활발히 이루어지고 있다. 이러한 개인적 의사표현의 중요성을 인지하여 이를 전산적으로 분석하고자 하는 다양한 연구들이 진행되기 시작하였고 이는 오피니언 마이닝(opinion mining) 또는 감성분석(sentiment analysis)이라 명명되었다(Liu 2012). 현대의 이러한 연구에 있어 각 문장의 토큰 인식과 사전 적용을 통한 형태소 분석 단계는 가장 중요한 첫 단계가 되는데, 글자 수의 제한이나 부주의 또는 의도적인 띄어쓰기 오류나 철자법 파괴 등의 현상이 빈번하게 나타나면서 그동안의 어떠한 종류의 자연언어 텍스트의 경우보다도 형태소 분석 결과의 오류를 발생시킨다. 즉, 소셜 미디어에서 수집되는 텍스트는 정제된 텍스트와는 달리, 띄어쓰기나 오탈자 그리고 신조어 등이 빈번하게 등장함으로써 컴퓨터가 자동으로 분석하기에 훨씬 어렵고 까다롭게 된 것이다. 한국어 SNS 코퍼스의 예를 몇 가지 보면 (1)과 같다.

(1) a. 눈썹문신도아프다고하던데 팔뚝은어떨지말잇못.
b. 여기는 예전부터 유명햇자낰.
c. 색깔도 너무 곱고, 전체적인 박시한 핏이 너무 마음에 듭니댱.
d. 판매대수가 절반인데 수익률 최고라는 건 전세계 앱등이들이 호갱이란거지ㅋ.

위의 (1)은 실제 관찰되는 문장들로서, 모두 기존 사전만으로 처리가 쉽지 않은 형태들이다. 한국어를 모국어로 하는 사람이 그 의미를 이해하는 데에는 크게 무리가 없으나, 문제는 컴퓨터로 이를 자동으로 처리할 때이다. (1a)은 전체적으로 띄어쓰기가 거의 지켜지지 않았으며, 특히 문장 종결부분의 '말잇못'이라는 표현은 '말을 잇지 못하다'를 줄여서 사용되는 형태이다. (1b)에서 '햇자낰'는 '했잖아'의 각 음절이 모두 변형되어 사용된 형태이다. 실제로 SNS 텍스트에서는 발음을 고려하면서 의도적으로 철자법을 지키지 않는 경우가 많다. 또한 이모티콘들을 특히 문미에 많이 사용하면서 'ㅋ' 혹은 'ㅎ'가 무종성 단어 뒤에 연이어 실현되는 경우 종성 위치에 결합한 형태로 실현되는 경우가 빈번하다. (1c)의 '박시한 핏'은 한국어 사전에는 아직 등재되지 않을 가능성이 있는 외래어 전사 표기이다. 동사 '듭니댱'의 경우는 '듭니다'에서 활용어미가 변형되어 실현된 형태이다. (1d)에서는 '앱등이', '호갱' 등의 신조어가 사용된 경우로, 각각 '애플 사용자', '고객'을 비하하는 표현이다. 이와 같이 기존 사전

을 통해 인식되지 않을 신조어의 출현 비중이 높기 때문에 이에 대한 처리가 동반되지 않으면 이들 단어가 포함된 부분에서 정보의 손실이 일어날 수 있다.

이렇게 SNS 텍스트에서는 비표준어적 표현이 끊임없이 생산되기 때문에 이를 자동으로 올바르게 분석해 내기 위해서는 유형들에 대한 보다 체계적인 접근이 중요해졌다. 현재 컴퓨터 프로그램에서 분석되지 못하고 남아있는 어절을 '미분석어절', '미분석어', '미분석 토큰' 등으로 명명하는데, 본 연구에서는 이들을 '미분석어'로 총칭하고자 한다. 이러한 미분석어는 한국어 자연어처리 및 오피니언 마이닝 등의 응용분야에서 분석결과에 왜곡된 결과를 가져올 수 있기 때문에 이에 대한 해결 방안을 마련하는 것이 시급하다.

본 연구에서는 SNS 텍스트에서 고빈도로 관찰되는 미분석어 유형을 분석하여 이들에 대한 처리 방안을 제안하고자 한다. 실제 SNS 텍스트에서 미분석되는 타입을 상향식(bottom-up)으로 분석하여 어떠한 유형으로 범주화가 가능한지, 또한 이러한 현상에 어떤 규칙적 특징이 관찰되는지, 그리고 이들을 어떠한 방식으로 보다 효과적으로 통제할 수 있는지를 논의할 것이다. 현존하는 전처리기들은 주로 일련의 학습 데이터를 이용하여 기계학습(machine learning)을 수행한 후 이를 토대로 새로운 데이터가 입력되면 전처리 작업을 통계적으로 휴리스틱하게 예측하도록 설계된다. 이를 위해서는 기계학습에 사용될 수 있는 학습 데이터의 질과 그 규모가 절대적으로 중요한 열쇠가 되는데, 정규화된 텍스트가 아닌 SNS 텍스트와 같은 비정형 문서를 올바르게 전처리할 수 있는 학습 데이터의 규모는 아직 미흡한 상태이다.

이런 점에서 본 연구에서는 SNS 텍스트에 나타난 문제들의 유형을 언어학적으로 분석하여 이를 변환하기 위한 형식화를 제안하는 것을 목적으로 하였다. 이 연구는 기존의 연구들과 몇 가지 차별성을 갖는다. 첫째, 언어학적 유형 분석에 입각한 변환 방식이므로 지속적으로 효과적인 보완과 확장이 가능하다는 점이다. 둘째로는 연구를 위한 응용 프로그램 유형에 관계없이 다양한 유형의 SNS 텍스트 분석에서 전처리를 위한 언어자원(language resources)으로 재활용될 수 있다는 점이다. 셋째는 재현율(recall)보다는 정확률(precision)에 더 중심을 둠으로써, 이 언어자원이 적용되어 가공된 코퍼스는 수정된 정도가 불완전하기는 하지만 기계학습 모듈에 의해 가공된 데이터에 비해 그 정확도와 신뢰도가 현저하게 높기 때문에, 곧바로 기계학습을 위한 학습 데이터와 같은 언어자원으로 투입될 수 있다는 점이다. 더불어, 부트스트랩(bootstrap) 방식을 이용하여 지속적으로 재적용함으로써 효과적으로 신뢰성 있는 자원을 보완 · 확장시킬 수 있다.

본 연구는 다음과 같이 진행된다. 2장에서는 미분석어에 대한 기존의 관련 연구를 검토하고, 3장에서 실제 미분석어의 유형을 추출하기 위한 코퍼스와 형태소 분석기 등에 대해 논의할 것이다. 4장에서는 미분석어들을 효과적으로 처리하기 위한 언어분석 플랫폼으로 DecoTex(Yoo & Nam 2017)와 유니텍스(Paumier 2003)을 소개할 것이다. 5장에서는 발생한 미분석어의 유형에 따라 이러한 플랫폼에서 실제 미분석어 처리를 위한 문법과 사전을 구축

하는 과정에 대해 소개하고 6장에서는 현재 구축된 언어자원의 성능을 실험할 것이다. 7장에서는 본 연구의 의의와 향후 연구 방향에 대해 논의하기로 한다.

2. 관련 연구 검토 및 문제의 제기

언어처리 연구에서 '미분석어(unanalyzed word)' 문제는 언어학 분야보다는 자연어처리 분야에서 더 논의되어 왔다. 그러나 이 분야에서도 이와 관련된 논문 수의 추이를 보면, 1990년대까지 활발히 진행되다가 2000년대부터는 연구자들의 관심과 중요성에 대한 인식이 감소되었다. 실제로 현재 자연언어처리 분야에서는 이미 해결된 주제로 인식되거나, 또는 중점적인 연구 주제가 아닌 부차적인 주제의 하나로 간주되는 경향을 보이고 있다.

지금까지 자연언어처리 분야에서 이루어진 연구는 주로 미분석어가 나타났을 때 이를 인식하고 추정하는 모델을 구축하는 것에 집중되어 있었다. 예를 들어 언어지식 기반 미분석어 인식 방법(양장모 외 1996, 차정원 외 1997), 주변문맥 기반 미분석어 인식방법(Weishedel et al. 1993, Nagata 1996), 또는 명사추출 기반 미분석어 인식 방법(이도길 외 2003), 전문분석 기반 미분석어 인식 방법(김선호 외 2002), 웹문서 기반 미분석어 인식 방법(박소영 2008) 등이 제시되었다. 이러한 연구들의 공통점은 특정 기준을 구축하여 미분석어의 정보를 통계적으로 '추정'하는 데에 있다. 그러나 이러한 추정을 위한 기준이 되는 언어지식이 어떻게 확보되는가에 따라 성능이 크게 좌우될 수 있다(박봉래 외 1996).

또 다른 연구로 이세희·김학수(2009)에서는 신조어, 줄임말과 같은 철자 오류들을 포함하는 텍스트 교정 모델을 제시하였다. 입력문장에 전처리 과정을 수행할 때 지정해 놓은 학습 시스템을 통해 교정하는 방식이다. 이 연구에서는 신조어와 줄임말에 대한 철자 오류들을 올바른 방식으로 교정하기 위한 모델을 제시하였지만, 이러한 교정 모델이 감성코퍼스를 분석하기 위한 전처리 시스템에서 고려되어야할 사항은 따로 언급되지 않았으며, 연구의 대상 또한 단순 철자오류를 교정하기 위함이어서 미분석어 처리를 목표로 하는 본 연구와의 차이점을 보인다. 남길임(2016)에서는 상품평 텍스트에서 나타난 감성표현을 추출하기 위해 상품평 어패럴 코퍼스를 구축하는 과정에서 전처리에 대한 문제를 언급하였다. 하지만 300여 개의 작은 단위의 오류-수정 쌍을 수집한 후 전처리를 수행하였기 때문에 실질적으로 활용성이 높다고 보기 어려운 한계점이 있다.

이상과 같은 기존의 연구들은 최근 들어 끊임없는 신조어의 생성과 더불어 문장 전체에 대한 형태 · 어휘적 변형이 이루어지고 있는 SNS 텍스트 처리 문제 이전에 제안된 것이 대부분이어서 한계가 있다. 가령 정형화된 텍스트의 모음인 21세기 세종 코퍼스와 현대의 비정형 텍스트 유형인 트위터 또는 리뷰글 등의 SNS 텍스트를 모은 MUSE[1] 코퍼스를 현존하는 형태소 분석기를 적용하여 분석해 보면 다음 그림 1과 같은 미분석어의 분포를 확인할 수 있다.

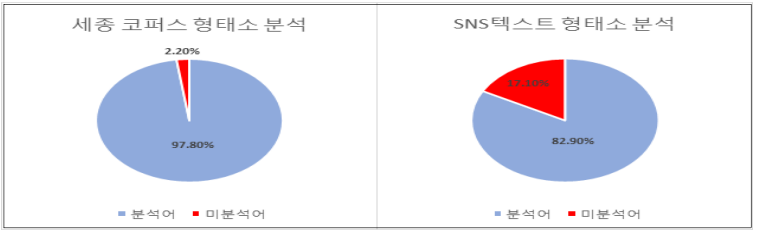

그림 1. 세종 코퍼스와 SNS 텍스트에서 나타난 미분석어의 비율 비교

위의 분석 결과는 국립국어원에서 제공하는 지능형 형태소 분석기를 사용하여 추출한 미분석어의 비율이다. 정제된 문장들로 이루어진 세종 코퍼스에서 임의로 구어체와 문어체 문서를 혼합하여 형태소분석을 실행한 결과, 전체 코퍼스의 7,776가지의 타입(type)에서 약 2.2.%에 해당하는 175개 타입만이 미분석어로 추출되었다. 반면 SNS 텍스트를 실험하기 위해 MUSE 코퍼스에서 정치면 신문기사의 댓글 문서를 추출하여 형태소분석을 수행하였다. 그 결과 총 5,894가지의 타입이 추출되었는데, 이때 17.1%에 해당하는 1,011가지의 타입이 미분석어로 나타났다. SNS 텍스트의 경우, 세종 코퍼스 대비 미분석율이 무려 7.8배에 이르는 것을 확인할 수 있다. 결국 이는 SNS 텍스트의 비정규적인 특징이 텍스트 분석에 적지 않은 걸림돌이 된다는 것을 보여준다.

이러한 이유로 초기의 언어처리 연구에서 조명받은 후, 현재에 와서는 상대적으로 특별한 관심의 대상이 되지 못하였던 '전처리(preprocessing)' 연구의 필요성이 다시 재기되고 있다. 기존의 예측 모델로는 처리되기 어려운 다양한 현상들이 최근 들어 빠른 속도록 생성되었기 때문이며 현재의 기계학습 모델로 이를 처리하기에는 여전히 양질의 대규모의 학습데이터가 제공되어야 하기 때문이다. 본 연구는 이러한 필요성에 대한 인식으로부터 시작되었다. 그리고 실제 대규모 SNS 텍스트에서 미분석어를 추출한 후 이들에 대한 유형을 분류하여 이를 정규화하기 위한 방법론에 대해서 논의하였다. 이러한 연구가 현재 통계적으로 예측하는 기계처리 방법으로는 수행되기 어렵기 때문에 실제 언어학 전공자들에 의해 정교한 변환문법과 언어자원이 구축되어야 할 것으로 보인다.

3. SNS 텍스트 형태소분석을 통한 미분석어 추출

3.1. 미분석어 추출을 위한 SNS 코퍼스와 분석 플랫폼

이 장에서는 SNS 텍스트에 나타나는 미분석어 유형을 추출하기 위하여 어떠한 코퍼스로

1) http://dicora.hufs.ac.kr를 참조할 것.

부터 어떠한 분석도구를 사용하여 이 작업을 수행하는지 논의한다. 본 연구에서는 한국외국어대학교 디지털언어지식콘텐츠연구센터(DICORA)에서 구축한 '문장층위 감성주석코퍼스(Sentence-level Sentiment-Annotated Corpora: SESAC)[2]'를 미분석어 추출을 위한 대상코퍼스로 사용하였다. SESAC은 모두 22가지의 도메인으로 구성되어 있는데, 이중 18가지 도메인은 온라인 신문기사의 댓글, 상품구매 후기, 맛집이나 영화, 성형수술 후기 리뷰글 등 다양한 온라인 후기글로 구성되어 있고, 나머지 4개 도메인은 트위터 문서를 수집한 텍스트로 구성되어 있다. 각 도메인당 10만~20만 어절씩 총 약 250만 어절의 규모이다.

미분석어를 추출하기 위해서는 위 코퍼스에 대한 토크나이징(tokenizing) 과정이 요구된다. 한국어의 경우 더 정확하게는 사전에 기반을 둔 형태소분석이 수행되어야 한다. 이때 형태소분석 과정에서 띄어쓰기 오류나 철자법 오류, 사전에 등재되지 않은 단어 등이 코퍼스에 출현할 때, 형태소 분석기에 이들에 대한 일정한 보완적 장치를 갖추지 않으면 올바른 형태소분석에 실패하게 되고 미분석어가 남게 된다. 그러므로 준비된 SNS 코퍼스로부터 미분석어를 수집하기 위해서는 형태소 분석기를 이용하여 코퍼스를 분석하는 작업이 우선 수행되어야 한다. 본 연구에서는 형태소분석을 위해 한국어 전자사전 DECO(Dictionnaire Electronique du COreen: Korean Electronic Dictionary, Nam 2015)에 기반을 둔 유니텍스(Unitex: Paumier 2003)[3] 시스템을 사용하였다. 유니텍스는 프랑스 파리 이스트 대학(University of Paris-Est)에서 구현된 다국어 사전 구축 및 코퍼스 분석 플랫폼으로서, 한국어를 포함한 다른 언어 텍스트가 유니코드로 저장되어 처리될 수 있는 다국어 호환성을 가진 것이 특징이다. 유니텍스는 유한 그래프 문법인 LGG(Local Grammar Graph: Gross 1997, 1999) 형식으로 다양한 어휘, 구문 패턴 등을 기술한 후, 이를 내부적으로 유한 트랜스듀서(Finite-State Transducer: FST)로 변환하는 기능을 갖추고 있어, 본 연구에서 미분석어 처리를 위한 언어자원 구축에도 중요한 플랫폼으로 기능하게 된다.

3.2. 미분석어 추출과 처리 플랫폼, 처리 방식

SESAC 코퍼스를 DECO사전을 이용하여 유니텍스 플랫폼에서 형태소분석을 수행한 결과, 전체 250만 어절로부터 약 17만 어절의 미분석어가 추출되었다. 이는 전체의 7.1% 정도를 차지한다. 이들 중 대다수의 경우는 한국어처리용 사전에 등재할 언어형태가 아니라 형태소 분석의 이전 단계인 '전처리' 모듈에서 다루어져야 하는 형태들임을 확인할 수 있다. 그런데 이와 같이 전처리 단계에서 처리되어야 할 형태들은 단어 층위가 아닌 음소 또는 음절, 형태소 층위인 경우가 많다. 예를 들면 (2)와 같이 단어 전체의 문제뿐 아니라 단어 내부의 일부 요소에 변이가 일어나는 표현들이 나타난다.

---

2) 여기에서도 http://dicora.hufs.ac.kr을 참조할 것.

3) 유니텍스 프로그램은 공개 플랫폼으로 http://unitexgramlab.org에서 다운로드 받을 수 있다.

(2) a. 열시미
b. 완죤
c. 좋앗죵
d. 갈걸루 (알았는데)

처음 것은 끝 2음절의 치환이 필요하고(즉 '열심히'), 두 번째 것은 끝음절의 치환이 필요하다(즉 '완전'). 세 번째 것도 끝 2음절의 치환이 요구되고(즉 '좋았죠'), 마지막 것은 끝 2음절의 치환과 함께 띄어쓰기가 이루어져야 한다(즉 '갈 것으로'). 이러한 변환이 이루어지지 않으면 현재의 형태소 분석기용 사전으로 분석되지 못하고 미분석어로 남게 된다. 그런데 이러한 음절 또는 음소 단위의 치환이 요구된다고 하더라도 (2a)과 (2b)의 경우는 어휘소 차원의 치환이라 곧바로 1:1 치환 방식으로 처리하는 것이 가능하다. 반면 (2c)과 (2d)의 경우는 양상이 조금 복잡해진다. 이들이 동일 유형의 문법 구성에 반복적으로 출현하는 문법소 차원의 문제이기 때문이다. 즉 (2c)과 (2d)에서 관찰되는 유사한 유형들을 살펴보면 다음 (3)과 같이 생산적인 유형임을 볼 수 있다.

(3) a. 먹엇죵, 웃엇죵, 죽엇죵
b. 먹을걸루, 잡을걸루, 막을걸루 (알았는데)

위에서 보듯이 (2c)과 (3a)에서 보인 구성은 '(동사/형용사)+었+죠'의 활용형이기 때문에 '동사/형용사' 부분에 다양한 변이가 일어나게 되며, (2d)과 (3b)의 구성은 '(동사/형용사)+(으)+ㄹ+것+으로' 연쇄가 축약된 형태이기 때문에, 이와 같이 다품사가 결합된 준말 형태의 연쇄를 올바르게 분리하지 못하면 현재 품사별 개별단어 중심의 사전으로는 이러한 형태들의 처리가 어렵게 된다. 문제는 (3)과 같은 형태들에 대한 효율적인 치환을 위해서는 이와 같이 한 토큰 내부에 들어있는 일정 성분이 동사 또는 형용사인가를 기술하거나 또는 확인할 수 있는 장치가 마련되어야 하는데 이는 형태소분석용 사전을 참조할 수 있는 단계에서 비로소 가능하다. 때문에, 그 이전의 전처리 단계에서 처리하기 어렵다는 순환적 딜레마에 빠지게 되는 것이다. 즉 (3)의 형태들이 다음 (4)와 같이 정규화된다면 이들에 대한 사전 적용을 통해 (5)와 같은 형태소분석이 가능해진다.

(4) a. (먹다)+었+죠
b. (먹다)+으+ㄹ+것+으로
(5) a. 먹다/Verb+었/PastEomi+죠/EndingEomi
b. 먹다/Verb+을/DeterminativeEomi+것/DependentNoun+으로/Josa

그런데 (3)의 형태들은 왜곡된 형태 때문에 사전 검색에 실패하므로 이들에 나타난 '먹'과 같은 음절이 동사라는 것을 확인할 수가 없다. 따라서 (3)의 연쇄를 치환하기 위한 변환 정규식을 다음과 같이 설정하는 것은 불가능하다.

(6) a. <Verb>+엇죵 ⇒ <Verb>+었죠
b. <Verb>+을걸루 ⇒ <Verb>+을+것+으로[4]

위와 같은 정규화가 불가능하다면 (3)과 같은 유형들을 치환하기 위한 정규식은 다음 (7)과 같이 되어야 한다.

(7) a. (먹|웃|죽|...)엇죵 ⇒ (먹|웃|죽|...)었죠[5]
b. (먹|잡|막|...)을걸루 ⇒ (먹|잡|막|...)을 것으로

이 때문에 (3)과 같은 형태들을 사전 적용 전에 바로잡기 위한 정규화 작업이 까다로워지는 것이다. 이 문제에 대해서는 다음 4장과 5장에서 다시 언급될 것이다.

4. 미분석어 처리를 위한 두 단계 플랫폼

4.1. DecoTex와 PGT 전처리 문법 테이블

본 연구에서는 우선 DecoTex 플랫폼(Yoo & Nam 2017)의 '전처리 문법 테이블(Preprocessing Grammar Table: PGT)'을 통해 이러한 미분석어 문제를 접근할 것이다. DecoTex 플랫폼에서 지원하는 PGT는 코퍼스의 잘못된 표현을 올바른 형태로 수정하기 위한 변환테이블로서 1:1 대응이나 정규표현(regular expression) 방식(파이선 언어에서 사용하는 정규식 형식)을 사용한다. DecoTex 플랫폼은 한국외국어대학교 DICORA에서 구축한 DECO 전자사전과 LGG 문법 적용문, 유니텍스 생성문 등의 언어자원을 활용하기 위한 전산 모듈로서, 코퍼스 수집부터 전처리, 감성분석 등 여러 모듈을 제공한다. 이 플랫폼에서 제공하는 전처리 모듈에서는 정규식을 테이블 형식의 파일에 저장한 후 이 파일을 호출하여 주어진 코퍼스를 분석하도록 프로그래밍되어 있다. 이러한 전처리 작업은 형태소분석 단계 이전에 적용되므로 사전 적용의 인식률을 높이는 효과를 가져온다.

---

4) 형용사(<Adjective>)는 제외한 경우임.
5) 이 예에서 '잡앗죵'이나 '슬펏죵'과 같이, 양성모음 어간이라 '었'대신 '았'이 실현되어야 하는 서술어나 무종성 어간이라 'ㅆ'이 수반되어야 하는 유형은 고려하지 않았음.

이 모듈에서는 처리하고자 하는 원시 코퍼스와 구축된 PGT 파일을 별도로 호출하게 되어 있으며, {Start Preprocessing} 기능을 이용하여 전처리를 수행한다. 이 작업이 수행되면 {Download Preprocessing Text} 기능을 이용하여 그 결과 문서를 저장할 수 있다. 다음은 PGT를 구성한 예를 보인다.

	A	B
1	before	after
2	맛과서비스는	맛과 서비스는
3	마넌	만 원
4	머라하나	뭐라 하나
5	늬들은	너희들은
6	할(라|려)(고|는|면)	하려₩2

그림 2. PGT 구성 예

위의 PGT 테이블은 기본적으로 'XXX.CSV' 파일 형식으로 작성된다. CSV 파일은 내용을 구분하기 위해 ','(쉼표)를 이용하는데 마이크로소프트 엑셀이나 한글과컴퓨터 한셀 등과 같은 스프레시트 프로그램을 이용하면 각 열마다 자동으로 ','를 입력하여 CSV 파일로 저장 가능하기 때문에 PGT를 구성하기 편리하다. 위의 테이블에서 A열의 'before' 부분에는 수정하려는 연쇄를 직접 또는 일련의 정규식으로 표현하고, B열의 after 부분에는 수정 후 변환하려는 형태를 입력하여 데이터 셋을 구성한다. DecoTex 전처리 모듈에서 사용하는 PGT 파일에서는 첫 번째 줄은 인식하지 않도록 프로그래밍되어 있으므로, 첫째열의 'before, after'에는 원하는 제목을 입력하여 구성할 수 있다. 둘째열의 '맛과서비스는'는 1:1 형식으로 수정 전 값과 수정 후 값이 직접 입력되어 있는 예를 보인다. 이렇게 다섯째 열까지 모두 1:1 방식으로 치환되도록 구성되었다. 마지막 여섯째 열은 정규식 문법을 사용한 경우이다. 즉 이것은 '할(라|려)(고|는|면)'을 '하려\2'로 치환하라는 명령으로 다음 (8)과 같은 입력 스트링들이 모두 (9)로 치환되도록 한다. '\2'는 치환 전 구성의 두 번째 괄호 '(고|는|면)'을 그대로 호출하는 정규식 표현이다.

(8) 할라고/할라는/할라면/할려고/할려는/할려면
(9) 하려고/하려는/하려면

정규식은 프로그래밍 언어에 따라 약간의 차이가 있는데, 앞서 언급한 바와 같이 DecoTex는 파이선 프로그래밍 언어와 호환되는 정규식을 인식하도록 되어 있으므로 여기서 사용하는 문법을 따라야 한다. 본 연구에서는 PGT를 구성하는 데에 가장 기초적인 정규식만을 이용하여 기술하고자 하였기 때문에 다른 프로그래밍 언어로 작성한 프로그램에서도 쉽게 호환이 가능하다. 다음의 표 1은 기본적으로 본 연구에서 사용된 정규표현의 연산자 및 문법을 설명

한 것이다.

표 1. DecoTex의 PGT에서 사용된 정규식 연산자 및 문법의 예

번호	문법	설명	예시	예시 의미
1	( )	집합 연산자. 묶어진 집합 문자를 '\1~\n'형식으로 받을 수 있음	(넘)(좋다) → \1_\2	넘좋다 → 넘_좋다 (두 집합을 띄어쓰기 위함)
2	[ ]	범주 표현(한글의 경우 하나의 음절만 받음)	[이그저]	'이' 또는 '그' 또는 '저' 표상
3	+	최소 하나 이상 반복	[가나]+	'가' 또는 '나' 1번 이상 등장 (예: 가가나나가, 가나가나)
4	|	또는 (OR) 표현, 문자열의 여러 유형을 한 번에 표현할 때	(넘|너무)	'넘' 또는 '너무'가 나타난 문자열 인식
5	^	문장의 시작점 표시	^오늘	'오늘'로 시작하는 문자열만 표상
6	$	문장의 끝점 표시(^과 반대 의미)	행복$	'행복'으로 끝나는 문자열만 표상
7	\	이스케이프(Escape) 문자 - 정규식에서 사용하는 특수문자의 용법을 해제하고 고유의 용법으로 표현하고자 할 때 (자판의 ₩)	\+	정규식 연산자에서는 1회 이상 출현된 문자열을 의미하지만 \를 사용하면 단순 부호 '+'로 인식
8	\s	공백(Space)이 있음을 표현	\s너무\s	'너무' 앞뒤로 공백이 있는 경우만 표상
9	0-9	'0'부터 '9'까지 숫자 표현	[0-9]+	0,1,2,3,4,5,6,7,8,9의 모든 숫자로 이루어진 구성 (순서/횟수 제한없이)
10	가-힣	'가'부터 '힣'까지 한글 표현	[가-힣]+	모든 한글 음절로 이루어진 단어(음절수와 순서 관계없이)

4.2. 유니텍스와 DECO 사전 · LGG 그래프문법

미분석어의 일부 유형은 위에서 언급한 것처럼 전처리 단계에서 1:1 또는 정규식 형식으로 코퍼스 자체를 변환시켜야 하는 형태들도 있지만, 사전적 정보를 보강하여 형태소분석 단계에서 처리되는 것이 더 효과적인 어절들도 존재한다. 따라서 이러한 형태들에 대한 처리를 위해 본 연구에서는 DECO 한국어 전자사전(Nam 2015)과 부분 문법 그래프(LGG: Local-Grammar Graph)에 기반을 두는 유니텍스(Unitex) 플랫폼을 사용한다.

DECO사전은 기계 가독형 사전(MRD: Machine-Readable Dictionary)으로 어휘 표제어에 대해서 활용, 형태, 품사, 통사 및 의미 정보를 DecoTagset에 기반하여 제공하는 대규모 언어자원이다. 현재 DECO사전은 총 27만개의 표제어로 구성되어 있으며, 지속적으로 신조어에 대한 추가 관리가 이루어지고 있다(Version 3.0/2017)[6]. DECO사전은 프랑스 전산언어학자 모리스 그로스(Maurice Gross)의 어휘문법 및 부분문법 이론(Gross 1997,

6) http://dicora.hufs.ac.kr 참조.

1999)을 바탕으로 세바스티앙 포미에(Sebastian Paumier)에 의해 개발된 자연언어처리 플랫폼인 유니텍스와 호환 가능하게 구현되어 있다. 유니텍스 플랫폼에서 DECO사전을 컴파일하고 이를 코퍼스 처리에 사용할 수 있는데, 이 과정에서 사전 기반 부분문법을 구현할 수 있다. 부분문법은 유한상태문법(Finite-State Grammar)의 형식을 가지는데 유니텍스 플랫폼에서 사용자가 직접 방향성 그래프 방식으로 표상하는 것이 가능하다. 이렇게 구축되는 LGG는 '유한 상태 오토마타(Finite-State Automata: FSA)' 및 '유한상태 트랜스듀서(Finite-State Transducer: FST)'로 자동 변환되어 코퍼스에서 분석할 수 있는 문법 형태로 전환된다. 통사 규칙의 형식으로는 일반화하기 어려운 개별 어휘적 문법 현상을 방향성 그래프로 효과적으로 표상하는 장점을 가진다(남지순 2013).

코퍼스의 한 구문을 DECO사전에 기반하여 분석한 결과의 예를 보면 (10)과 같다. '요새 나오는 아이돌들'을 분석한 결과이다.

(10) {요새,요새.DS+ZDZ} {나오,나오다.VS+ZVZ}{ㄴ,ㄴ.EV+DT+PAS} {아이돌,아이돌.NS+ZNW}{들,들.JN+SP+PLU}

이를 기반으로 일정 문법적 단위를 LGG로 표상하여 처리하면 아래 그림과 같은 그래프형식으로 구현된다. 다음은 '명사'와 '조사'가 연이은 형태를 인식하기 위한 유한오토마타 형식을 보인다.

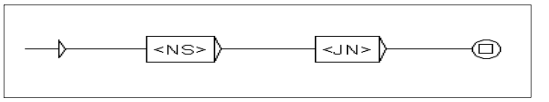

그림 3. LGG 표현 예

5. 미분석어 유형별 처리를 위한 전처리기 및 사전확장 연구

앞서 언급한 바와 같이 이 장에서는 미분석어 처리의 문제점을 해결하기 위해 미분석어의 유형을 두 범주로 나눈다. SNS 텍스트에 나타나는 문제의 유형을 PGT로 변환하는 부분과 DECO사전 및 LGG로 처리하는 부분으로 나누어 설명한다. PGT로 해결할 수 있는 유형은 대체로 단순오타 및 띄어쓰기 문제, 철자법 파괴, 어휘소 변형 형태 등으로, 이들은 사전에 등재되어야 할 형태로 판단되지 않으므로 텍스트 자체를 변환하는 PGT를 적용한다.

반면, 의존명사나 활용어미 등 품사 정보에 의존적인 패턴을 가지고 변형이 일어나는 경우는 전처리 방식이 용이하지 않다. 또한 발생빈도가 증가하는 신조어나 복합구성 유형은 사

전적 처리가 바람직하므로 이들은 LGG로 기술하거나 사전 표제어 항목에 직접 추가하여 형태소 분석 단계에서 처리하도록 하였다. 이에 해당하는 유형은 활용어미 및 의존명사 등의 변형 패턴, 특수문자 및 이모티콘 형태, 그리고 신조어 및 외래어 등이다.

5.1. DecoTex 플랫폼에서 PGT를 통한 미분석어 처리

5.1.1. '띄어쓰기 오류 · 생략 유형' 처리와 DecoPGT 테이블 문법

띄어쓰기 문제는 SNS 텍스트에서 가장 빈번하게 발생하는 문제 중 하나이다. 한국어 맞춤법에서 붙여 쓰기도 허용되는 경우가 많아 다양한 형태가 지속적으로 나타날 수 있으며 일반 문어 코퍼스와 다르게 여러 사람이 작성하는 SNS 코퍼스에서는 더욱 변칙적인 띄어쓰기 오류 유형 및 생략 유형이 나타난다. SNS 코퍼스에서 발생한 띄어쓰기 문제의 유형별 본 연구의 처리방식은 다음과 같다.

● 유형 1. <관형절 + 비단위성 의존명사> 내포 미분석어 유형

현재 빈도가 가장 높은 띄어쓰기 오류 연쇄의 언어학적 구조는 <관형절 + 비단위성 의존명사>를 포함한 유형과 연관되어 있다. 이러한 경우는 주로 한 글자로 구성된 의존명사인 경우가 많으며 몇 가지 예를 보면 (11)과 같다.

(11) a. 까불까불대는것도 좋고 흥 많은 남자 겁 많은것도 귀엽지
b. 맑고 투명함을 다끌어다 모은듯하다
c. 박보검 종교발언 정신나간여자들은 좋다고 그럴수있지 감싸고 날뛰네
d. 지창욱이 그런데서 노는게 걍시룸
e. 우엥 당장 내일 명동갈때 해야대는데 어찌해야할지모르겠다.

단음절로 된 의존명사 유형은 왼편에서 수식하는 관형절과 결합한 형태로 실현되는 경우가 많다. 이는 SNS 텍스트뿐 아니라 일반적으로 한국어 띄어쓰기 오류에서도 많이 나타나는 유형이다. 이를 PGT로 처리하면 이들 비단위성 의존명사 앞의 모든 한글을 띄어주는 형식으로 변환할 수 있을 것이다. 이를 테면 '([가-힣]+)것 → \1_것' 이런 식으로 입력하여 띄어주는 것이 가능하다. 그러나 이러할 경우 몇 가지 문제가 발생한다. 사전처리 이전 단계의 전처리 정규식으로는 의존명사 앞의 문법적 환경을 제어해줄 수 없기 때문에 모든 한글 조합이 오도록 한 것인데, 이 경우 '이것저것'이라는 단어가 온다면 '이 것저 것' 이러한 식으로 오교정이 일어난다. 또한 여기서 'ㄴ, ㄹ' 뒤에 의존명사가 실현될 때, 이들 형태소가 관형형 어미임을 정의하는 것이 필요하다. 따라서 이런 경우 PGT에 의한 변환보다는 문법적 정보를 이용할 수 있는 사전 차원의 LGG를 사용하는 것이 더 효율적이다. 이들 유형의 경우는 '어찌해야할

지모르겠다'와 같은 경우 '어찌해야'와 '할지', '모르겠다'로 변환하는 PGT를 구성한 후 '할지'와 같은 결합형은 뒤의 LGG에서 처리되도록 하였다. LGG에 대해서는 5.2.1절에서 후술된다.

● 유형 2. <관형절 + 명사> 내포 미분석어 유형

이 유형은 앞서 언급한 유형 1과 마찬가지로 관형절이 오고 그 뒤에 명사가 수반되는 유형인데, 유형 1과 다른 점은 유형 1에서는 비단위성 의존명사의 목록이 어느 정도 한정되어 있다는 점이지만, 여기서는 명사 전체이기 때문에 그 목록을 한정지어서 설명할 수 없다는 점이다. 이에 대한 예를 보면 (12)와 같다.

(12) a. 딴남자아이 때문에 세상 우울해하고 있는 양이.
b. 컵을 바꿔치기 했던 게 생각남 미친애녀석들.
c. 이런속담표현은 한국어, 영어 둘 다 있다.

(12)의 경우를 보면 알 수 있듯이, 관형절 뒤에 오는 명사의 유형이 다양하게 나타날 수 있기 때문에 이들 중 출현빈도가 높을 것으로 판단되는 유형은 복합어 사전의 표제어로 등재되는 것이 바람직하다. 반면 일회성 오류 혹은 사전 등재의 대상으로 간주하기 어렵다고 판단되는 명사구 구성은 PGT를 통해 띄어쓰기 문제를 바로잡는 것이 필요하다.[7] PGT로 변환하는 경우는 주어진 토큰에 대한 오교정이 되지 않도록 앞뒤에 제약을 두는 것이 필요하다. 표 2는 이에 대한 예이다.

표 2. 〈관형절 + 명사〉 띄어쓰기 관련 PGT 구성 및 설명

수정 대상	수정 후 형태	설명
(\s|^)딴남자아이($|\.|\!|\?|\s)	\1딴 남자아이\2	제약을 둔 "딴 남자아이" 띄어쓰기
(\s|^)미친애녀석들($|\.|\!|\?|\s)	\1미친 애녀석들\2	제약을 둔 "미친 애녀석들" 띄어쓰기
([가-힣]+)런속담표현	\1런 속담표현	확장 고려 "이런 속담표현은" 띄어쓰기

위의 표에서 의도하지 않은 어절을 변경하지 않도록 앞뒤에 공백 및 마침 표현에 대한 정규표현으로 제약을 두었다. '(\s|^)'는 공백(\s) 또는 문장 시작(^)이 앞에 위치할 수 있다는 정규 표현식이고, '($|\.|\!|\?|\s)'는 문장 끝($), 마침표(\.), 느낌표(\!), 물음표(\?), 공백(\s) 등이 뒤에 올 수 있다는 제약을 정규 표현식으로 구성한 것이다. '([가-힣]+)런애녀석들'의 경우, '( )'는 집합, '[ ]'는 범주를 표현하는 정규 표현식인데 범주 표현 내에 '가-힣'을

7) 언어학 연구에서 논의되는 합성명사와 복합명사구의 정의를 따라 주어진 토큰을 사전에 등재하는 유무를 고려하는 것은 본 연구에서 제기하는 문제의 해결에 핵심적 논의가 되지 않으므로 여기서는 배제하기로 한다.

입력한 것은 한글 전체를 범주로 표현한 것이며 '+'는 한 번 이상 반복될 수 있다는 것을 표현한 연산자이다. 그리고 이를 묶어서 집합으로 표현하며, 수정 후의 형태에서는 '\1'(첫번째 집합)으로 호출하게 한다. 그래서 '([가-힣]+)런애녀석들'는 '이런애녀석들, 그런애녀석들, 이런애녀석들을, 이런애녀석들의' 등 다양한 형태를 수정할 수 있다.

● 유형 3. <명사 + 비단위성 의존명사> 내포 미분석어 유형

앞서 유형 1이 비단위성 의존명사 앞에 관형절이 오는 경우라면, 이 경우는 의존명사가 명사를 수반하는 경우이다. 몇 가지 예를 보면 (13)과 같다.

(13) a. 저날에 별 일 없으면 전주 놀러갈까 고민중인데.
b. 합작 공개되서 홍보겸 올려요!
c. 오늘 저녁때 별일없으면 미생시간 맞춰서 1국 볼거에요.
d. 개통시 안내문 내용이 좀더 자세히 나왔으면 좋겠어요.

이 경우도 결합 목록이 상대적으로 제한된 경우 가능한 한 복합구성 사전 표제어로 등재되는 방식을 취하고, 사전에 일일이 등재하는 것이 적합하지 않다고 판단되는 유형은 PGT에서 변환하는 방식을 취한다. 다만 이들에 대한 문법적 정보 없이 음절 차원의 기술만이 가능하므로 분명히 과분석의 위험이 배제된 경우가 아니라면 1:1 방식의 PGT 구성이 요구된다. 표 3은 이에 대한 예이다.

표 3. 〈명사 + 비단위성 의존명사〉 띄어쓰기 관련 PGT 구성 및 설명

수정 대상	수정 후 형태	설명
(\s|^)고민중	\1고민 중	확장 고려 "고민 중인데" 띄어쓰기
(\s|^)홍보겸	\1홍보 겸	확장 고려 "홍보 겸" 띄어쓰기

● 유형 4. <수사 + 단위성 의존명사> 내포 미분석어 유형

해당 유형은 수사 표현의 수식을 필요로 하는 단위성 의존명사가 출현하는 연쇄이다. 몇 가지 예를 보면 (14)와 같다.

(14) a. 티켓 두장이나 있는데ㅡ ㅜㅜ
b. 전지현 화교가 아니라고 여러번 부인을 했나보네
c. 오늘은 워너원의 걸리버 막내 관린이의 열일곱번째 생일입니다

이들에 대한 PGT를 구성할 때, '[0-9]+'를 포함시켜 '1장, 17번째' 등과 같이 아라비아 숫자

가 출현할 경우에 대한 정규식과 (한|두|.. 열여덟|열아홉|스무) 등과 같은 수관형사를 동반하는 경우를 고려하는 것이 필요하다(표 4 참조).

표 4. 〈수사 + 단위성 의존명사〉 띄어쓰기 관련 PGT 구성 및 설명

수정 대상	수정 후 형태	설명
(\s|^)(한|두|..|[0-9]+)장	\1\2 장	확장 고려 "두 장이나" 띄어쓰기
(\s|^)(한|두|..|[0-9]+)번	\1\2 번	확장 고려 "여러 번" 띄어쓰기
(\s|^)(한|두|..|[0-9]+)번째	\1\2 번째	확장 고려 "열일곱 번째" 띄어쓰기

만일 위 표 4에서 '장', '번' 등의 단음절 앞부분에 대해서 '([가-힣]+)장'과 같이 PGT를 구성한다면 다수의 오류를 발생시킬 것이다. 따라서 앞부분에는 정확한 수사표현으로 제약을 두고 뒤에는 제약을 가하지 않음으로써 조사의 확장성을 고려하는 방식이 필요하다.

● 유형 5. <명사 + 명사> 내포 미분석어 유형

유형 5는 명사와 명사가 연결되는 형태이다. 명사 연쇄의 경우 '단일민족', '고등학교'와 같이 복합명사로 인정되어 단일어로 사용되는 경우가 있다. 이러한 특성을 반영하여 명사와 명사 연쇄는 가능한 경우 복합구성 사전 표제어로 등재하는 방식이 바람직하다. 다만 그 실현빈도가 상대적으로 낮다고 판단되는 형태는 PGT를 구성하여 변환하도록 한다. 몇 가지 예를 보면 (15)와 같다.

(15) a. 이민호드라마소식이라니 너무 좋으다ㅜㅜ
b. 170930 명동팬싸인회 다녀왔어요><
c. Gpro 가격대비성능비가 좋아지고 있다.
d. 요즘 스크린활용도가 높아졌어요.

(15a)의 경우는 '이민호, 드라마, 소식'이라는 3단어의 연쇄로 이루진 것으로 그 전체가 사전에 등재되어야 하는 복합구성으로 처리하기 어렵다. 반면 (15d)를 보면 '스크린'과 '활용도'라는 두 단어의 연쇄이지만, 최근 IT 제품의 관심 증가로 한 단어처럼 사용되는 사례가 빈번하여 이들을 사전에 등재하는 것이 효과적으로 보인다. 따라서 이 경우에도 (15a)와 같은 연쇄는 1:1 방식의 PGT를 통해 세 단위로 변환되며, (15b)의 경우도 '명동'과 '팬싸인회'를 분리하는 PGT를 구성한다.

● 유형 6. <명사 + 동사/형용사> 내포 미분석어 유형

유형 6은 명사에 용언, 즉 동사나 형용사가 연속하는 경우이다. 해당 유형은 DECO 복합

어 사전에 등재하는 방식과 PGT를 활용하여 텍스트를 변환하는 형식을 병행하여 사용한다. 다음의 예 (16)을 보자.

(16) a. 아이폰과 갤럭시 사이에는 가격차이없어요.
b. 비와이얘기한 적 없어요.
c. 한국판 에어포스원 비행기주인 길라임... 국뽕맞는다.

위에서 (16a)의 '가격차이없어요'는 PGT를 통해 '가격차이'와 '없어요'로 변환되는 것이 바람직하며, (16b)의 경우도 '비와이얘기하다'가 사전에 등재되기 어려운 유형이므로 '비와이'와 '얘기한'이 PGT를 통해 분할된다.

● 유형 7. <부사 + 동사/형용사> 내포 미분석어 유형

유형 7은 주로 '안, 못' 등의 부정부사와 용언이 결합한 형태인데, 많은 경우 DECO 사전에 표제어로 추가 등재하는 방식을 취하였다. 다음의 예 (17)을 보자.

(17) a. 못난 애들이 ㅠㅠ
b. 난 말 안했는데 송중기 얼굴 있는 포장판에 줌 ㅎㅎ
c. 마케팅 좐나못하네 ㅋㅋ

(17)의 예들에 나타난 구성은 부정부사인 '안'과 '못'이 결합한 형태들로서, DECO사전에 '못나다', '안하다' 형태로 등재된다. 그러나 (17c)의 경우를 보면, '좐나'라는 형태가 결합되어 있어 이 경우는 '좐나못하네'를 '좐나 못하네'로 변환하는 PGT를 구축한다.

● 유형 8. <동사/형용사 + 동사/형용사> 내포 미분석어 유형

이 유형은 동사나 형용사에 연결어미 '-고-'나 '-게-' 등이 결합하고 다시 동사나 형용사가 수반되는 유형이다. 특히 '본용언 + 보조용언' 형태의 경우 개별 어휘이지만 붙여 쓰는 것이 허용되므로 이러한 복합구성의 가능성을 예측하거나 목록화하지 못하는 경우 분석의 어려움이 나타난다. 이들의 경우도 복합구성으로서 사전에서 처리되는 것이 가능한 형태들은 직접 사전에 등재하되, 그렇지 않은 경우는 PGT를 통해 텍스트를 수정하는 절충적 방식이 바람직하다. 이들 중 몇 가지 예를 보면 (18)과 같다.

(18) a. 역대급 가을 신제품 제가 한 번 먹어보겠습니다.
b. 송중기님의 팬이시라 선물로 보내고싶으시다구.
c. 눈 뜨는 지창욱 평범한 행동을 평범하지 않게만들면 어떻게 하나요 ㅜㅜ

(18a)처럼 '-어보다'의 형태는 명확한 기준을 제시하기 어렵다. 일례로 국립국어원의 표준국어대사전에서는 '물어보다'의 경우는 합성어로 등재되어 있으나, '먹어보다'는 합성어로 등재되어 있지 않다. 이런 경우 분석의 오류가 나타나게 된다. (18b)와 (18c)는 일반적으로 복합동사로 분류되지 않는 유형들이다. 본 연구에서는 (18a)와 같은 유형은 DECO사전에 추가적으로 등재하되 (18b)와 (18c)와 같은 연결구성형은 PGT를 통해 변환하는 방식을 취한다.

● 유형 9. 파생접미사에 의한 복합구성

일부 단어에서는 명사구나 어근에 특정 접미사가 결합하여 새로운 토큰이 구성되는 경우를 볼 수 있다. 다음의 예 (19)를 보자.

(19) a. 하는 짓이 참 애플스럽다.
b. 공고출신답다 니새끼 뇌피셜이지?

(19a)는 외래어로 된 고유명사에 '-스럽다'가 결합하여 생성된 형태이고, (19b)는 명사구 '공고출신'에 '답다'가 결합한 복합구성이다. 그 어느 것도 사전에 등재되어 있을 가능성이 낮은데, (19a)의 경우 '스럽다'가 단독으로 단어를 구성하지 못하므로 '애플스럽다'를 신조어의 유형으로 사전에 등재하는 것이 바람직하다. (19b)에서도 '답다'만으로 단어가 구성되지 못하나, 이 경우는 이미 선행하는 성분이 일련의 명사구를 구성하고 그 조합 목록의 생산성이 매우 높기 때문에 본 연구에서는 우선적으로 '공고'와 '출신답다'를 PGT로 분리하는 방식을 채택하였다.

5.1.2. '철자법 오류 · 파괴 유형' 처리와 DecoPGT 테이블문법

이상에서 띄어쓰기 문제로 나타난 미분석어를 살펴보았다면, 하나의 단일 토큰 내에서 철자법이 변형되어 실현된 미분석어들을 관찰할 수 있다. 철자법 오류 및 파괴 형태는 주어진 토큰 내의 일부 음절이 변형되어 나타나는 현상으로서 SNS 텍스트에서는 의도하지 않은 오류 현상뿐 아니라, 사용자가 의도적으로 형태를 변형하여 사용하는 경우도 빈번하게 나타난다.

● 유형 10. 용언어간 · 활용어미가 변형된 유형

활용어미와 관계된 변형 형태에 대해서는 세 가지로 나눌 수 있다.

▪ 종결어미 변형 유형

(20) a. 했어염 (<했어요)

b. 그랬당 (<그랬다)
c. 거에요 (<거예요)

- 선어말어미 변형 유형

(21) a. 그랫었어요 (<그랬었어요)
b. 오새요 (<오세요)
c. 사랑이었잔아 (<사랑이었잖아)

- 어간, 어미 모두 변형된 유형

(22) a. 머거 (<먹어)
b. 미쳐따 (<미쳤다)
c. 조켓다 (<좋겠다)

먼저 (20a, b)의 '했어염, 그랬당' 등의 예는 '-요, -다' 등에서 마지막 종결어미만 변형된 경우이다. 현대 SNS 텍스트에서 의도적으로 이러한 어투의 글쓰기가 증가하고 있어 그들은 이미 SNS 텍스트의 특성을 보인다고 판단되므로, 전처리에서 치환하는 것보다는 별도의 사전 형식으로 이들을 인식하는 것이 바람직하다고 보인다. 반면, (20c)의 '거에요'는 단순히 철자 오류로 볼 수 있기 때문에 전처리로 변환하는 것이 효과적인 방법이라고 판단하였다. (21)의 유형은 선어말어미의 변형으로 인한 경우이다. (21a)은 '그러하다'의 어근에 '-었었-'이 결합되어야 하는데, '-엇었-'으로 표기되어 있는 경우이고, (21b)은 '오세요'에서 '-세-'의 오타인 '-새-', 그리고 (21c)에서는 '-잖-'을 '잔'으로 나타낸 것이다. 이들은 앞서 (20)과 달리 사전에 수록할만한 유의미한 변형으로 판단되지 않으므로 PGT를 이용하여 변환시키도록 한다. 마지막 (22)의 예는 어간과 어미가 결합할 때, 어미뿐만 아니라 어간까지 변화를 일으키는 유형들이다. 이러한 예들은 앞서 (20)의 경우처럼 DECO 활용어미 사전에서 제어하는 것이 용이하지 않다. 이들은 특정 동사 또는 형용사 어휘소와 결합하여 변형된 형태들이므로 전처리 단계에서 PGT를 통해 변환되는 것이 불가피하다.

이밖에도 어미 활용형과 관련하여, 철자를 잘못 입력하거나 자판을 잘못 사용해서 실현된 것으로 보이는 유형들이 나타난다. 이러한 형태들도 모두 전처리 단계에서 PGT를 통해 변환된다. 예를 들면 다음 (23)과 같다.

(23) 철자 입력 오류 유형
듀ㅓ쓰(<뒀), 미쳐ㅑ쓰(<미쳤어), 끝이없는ㄴ(<끝이 없는), 귀여유ㅓ(<귀여워), 엿ㅅ어요(<였어요), 가곴ㅍ어(<가고 싶어), 햄ㅅ지요(<했지요),

● 유형 11. 명사 및 부사형의 변형

철자법 오류 및 파괴의 또 다른 유형은 명사 또는 부사 범주의 어휘소들과 관련하여 나타난다. 이들의 유형은 상대적으로 많지 않은 편인데, 크게 발음에 따른 형태 변형, 키보드 입력 오류에 따른 형태 변형으로 나눌 수 있다.

- 발음의 영향으로 변형된 유형

(24) a. 표기법 혼란

데박(<대박), 테그(<태그), 모댈(<모델), 내이버(<네이버)

b. 발음그대로 표기

촤령(<촬영), 마니(<많이), 그노무(<그놈의), 지으니가(<지은이가)

c. 축약 표현

베치씨(<비에이치씨), 유튭(<유튜브), 전젼(<전지현), 왜케(<왜 이렇게), 뱅기(<비행기), 왜냠(<왜냐하면), 글고(<그리고), 천랸(<천리안)

d. 발음 변형

쥬금(<죽음), 증말(<정말), 뽕아리(<병아리), 슨생님들(<선생님들), 완죤(<완전), 구냥(<그냥), 덴당(<젠장), 데이뚜(<데이트)

- 키보드 입력 오류 유형

(25) a. 자모음 분리 형태

ㄴㅓ(<너), 아ㅇㅣ유(<아이유), 구ㅣ걸이(<귀걸이)

b. 주변 키로 잘못 입력된 형태

사럼(<사람), 최댜한(<최대한), 마앾류(<마약류)

c. 이해 불가 유형

ㄱㄱㄱㅂㄱㄱㄱ, ㅅㅂㅁㅇㄹㅎㅁㅇㄴㄹ

(24)의 경우는 언중들의 발음 습관이 철자에 영향을 준 유형으로 4가지로 분류된다. (24)의 예들이 어떠한 의도를 가지고 사용된 예라면, (25)은 단순한 키보드 오타로 판단된다. 위의 유형들은 전처리 단계에서 PGT를 통해 변환되는 것이 바람직하다.

5.2. 유니텍스 플랫폼에서 DECO사전과 LGG를 통한 미분석어 처리

5.2.1. DECO사전과 LGG 그래프문법에 의한 처리

이 장에서는 텍스트를 변환하는 PGT 구성 방식과는 달리, 사전에 수록된 문법정보를 활용하는 것이 효과적인 형태들에 대해 DECO사전의 어휘소와 문법소를 LGG방식으로 처리

하는 방법에 대해 논의한다.

● 유형 1. 변형된 종결형 활용어미의 처리
다음의 예 (26)에서 밑줄 친 부분을 보자.

(26) a. 저는 오늘 너무나도 바삭하고 고소한 돈까스를 먹었어용.
b. 가지 마셈.
c. 쓰레기예욤.

앞서 언급한 바와 같이 이 부분을 PGT를 이용한 전처리 단계에서 텍스트를 변형하는 방식으로 처리하지 않고 DECO사전의 문법소 LGG에서 이들 형태의 인식 자체가 가능한 방식으로 처리하고자 하는 이유는 두 가지이다. 첫째는 이들은 SNS 텍스트에서 빈번하게 발생하는 특징적 언어학적 현상의 하나로서, 일회성으로 배제하지 않고 주목할 필요가 있는 유형들로 판단된다는 점이다. 둘째는 이들은 거의 대부분의 용언에 결합 가능한 활용어미들이므로 PGT에서 단순 음절 기반으로 변환하는 것은 한계가 있거나 또는 과생성되는 위험이 있어 '용언'이라는 사전 정보를 기술할 수 있는 사전의 문법소에서 처리하는 것이 더 효율적이고 정확하기 때문이다. 이들 유형을 분석하면 종결어미 '요'가 변형된 형태, 종결어미 '다'가 변형된 형태, 종결어미 '세+요'가 변형된 형태, 종결어미 '지+요'가 변형된 형태로 나눌 수 있다. 가장 빈번히 출현하는 종결어미 '요'의 변형 형태들을 인식하기 위한 LGG는 그림 4와 같다.

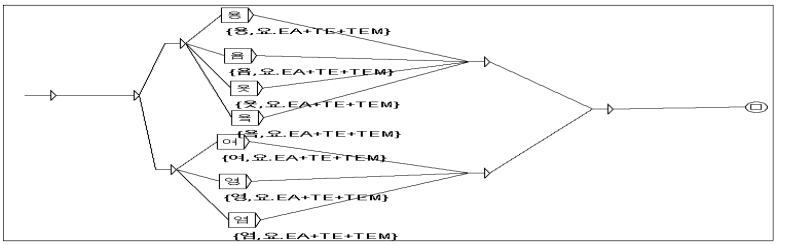

그림 4. 종결어미 '요'의 변형 형태를 인식하기 위한 LGG

위의 LGG를 현재 DECO 활용형 사전에 추가적으로 적용함으로써, 형태소분석 단계에서 SNS 텍스트에 나타난 '-용, 욤, 욧, 여, 영, 염' 등의 변이형들을 모두 인식할 수 있으며, 이들의 본래 기본 형태소가 '요'라는 정보를 태그값으로 할당할 수 있게 된다. 종결형 '-지요'의 변형된 유형은 다음의 예 (27)과 같다.

(27) a. 무교동 낙지 맛나죵ㅎㅎ. (→맛나지요)
b. 우왕 ! 맛있쪼 맛있쪼 ㅋㅋ. (→맛있지요)

다음 그래프는 위와 같은 변형 유형을 인식하기 위한 LGG이다. 이 LGG는 용언의 활용어미 '-지요' 대신 실현되는 '-죵, 징, 찡' 등의 종결어미 형태를 인식하기 위해 사용된다.

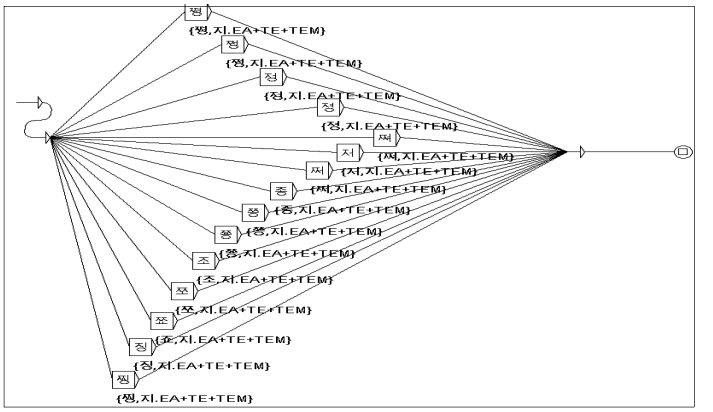

그림 5. 종결어미 '지+요'의 변형 형태를 인식하기 위한 LGG

이와 같은 변이 형태들을 LGG에서 기술함으로써 이와 관련된 현재의 미분석어들이 올바르게 처리될 수 있다.

● 유형 2. 명사 뒤에 실현되는 의존명사 또는 통사적 접미사 부류에 대한 처리

명사 뒤에서 '년씩'이나 '천일동안'과 같이 선행 명사와 결합하여 복합명사를 이루는 단어의 경우, 그 결합 생산성이 높다고 판단되는 유형의 경우는 LGG를 통해 기술하였다. 다음 LGG는 명사와 결합하여 실현될 수 있는 일련의 의존명사 또는 통사적 접미사(또는 학자에 따라 '조사'로 분류) 유형을 인식할 수 있도록 구축된 그래프문법을 보인다.

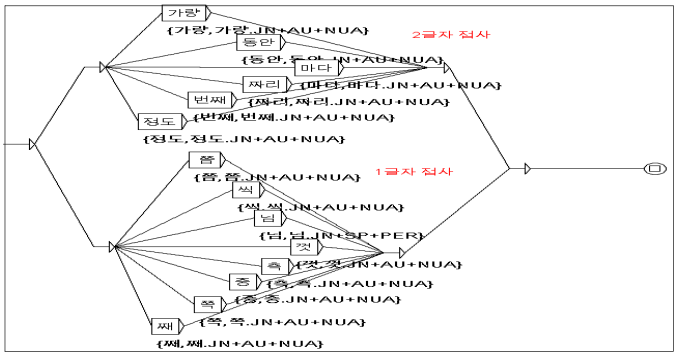

그림 6. 명사 이후에 결합된 접사성 단어를 인식하는 LGG

이 그래프는 '명사(NS)+'가량/동안/정도'+조사(JN)'와 같은 연쇄를 분석하기 위해 명사 뒤의 일련의 의존명사 유형의 성분들을 표상한 LGG로서, DECO사전에서 명사 표제어에 조사가 결합하기 전에 연결되도록 구조화되었다. 이 그래프의 연결을 통해 실제 코퍼스에 나타난 미분석어 '여름철동안에는'과 같은 연쇄에 대한 올바른 분석이 가능해진다.

● 유형 3. 의존명사 {것}과 관계된 변이형 구문의 처리

전술하였던 <관형절 + 비단위성 의존명사> 유형 내에서도 가장 빈도가 높은 형태가 의존명사 '것'과 관계된 유형이다.

(28) a. 되는거 찾아보자.
b. 부족한건 없는데.
c. 내가 먹을게 없다.
d. 내가 먹을걸 주웠다.
e. 내가 먹을걸로 생각한다.

위의 형태들을 PGT의 정규식으로 처리한다면 다음과 같은 형태가 가능할 것이다.

(29) a. ([가-힣]+)거 → \1_것
b. ([가-힣]+)건 → \1_것은
c. ([가-힣]+)게 → \1_것이
d. ([가-힣]+)걸 → \1_것을
e. ([가-힣]+)걸로 → \1_것으로

그러나 앞서 언급한 바와 같이 '것'이라는 의존명사에 대한 정보 없이 이러한 음절 정보만으로 이를 제어하는 것은 한계가 있다. 가령 (29a)에서 '증거, 수거, 선거, 과거' 등 '거'로 끝나는 단어의 경우, '증것, 수것, 선 것, 과것'과 같이 잘못 변환되는 문제가 발생한다. (29b)에서도 '수건, 가족건강' 등과 같은 단어들이 맞지 않게 변환되는 결과가 생성된다. 더 어려운 문제는 (29c)와 (29d)이다. 다음의 예 (30)을 보자.

(30) a. 내가 사과를 먹을게.
b. 내가 사과를 먹을걸.

위의 (30)에 출현한 '먹을게'와 '먹을걸'은 앞서 (29c), (29d)과 형태가 같지만 그 문법적 기능이 다르다. 즉, (29c), (29d)의 '게'와 '걸'은 의존명사 '것'과 조사가 결합된 형태를 정규화하지만 (30a), (30b)의 경우는 '-(으)ㄹ게' '-(으)ㄹ걸'의 종결어미로 사용된 형태이다. 이에

따라 모든 '용언+(으)ㄹ게' '용언+(으)ㄹ걸' 유형을 획일적으로 의존명사 '것'의 결합형으로 변환하여 분석하는 것은 위험하다. 그러므로 두 가지 분석 가능성을 모두 사전 LGG에 기술한 후, 주어진 문맥에 의해 올바르게 선택하도록 하는 추가적인 문법 장치가 요구된다. 본 연구에서는 이러한 중의적 유형을 각각의 LGG로 표상하였다. 가령 의존명사 '것'과 관련된 LGG의 예는 그림 7과 같다.

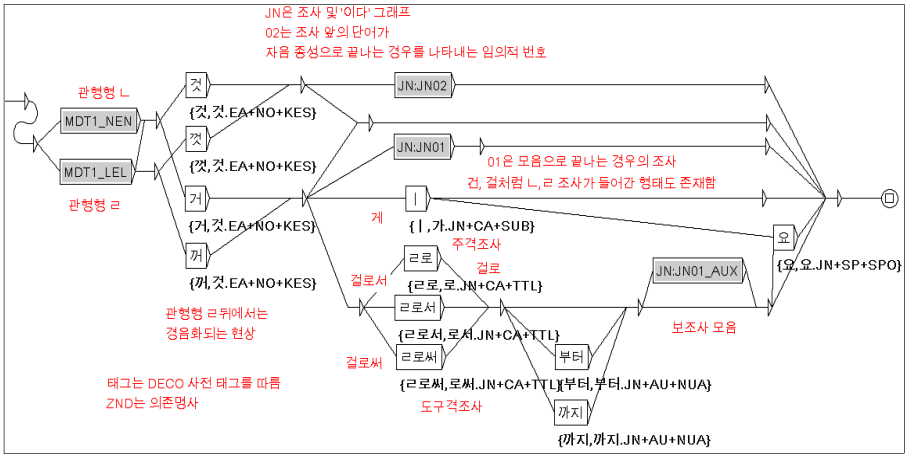

그림 7. 의존명사 '것'과 관련된 LGG

● 유형 4. 그 외 관형절을 수반하는 단음절 의존명사 관련 구문의 처리

앞서 의존명사 '것'처럼 해결하기 어려운 경우가, 관형절 뒤에 결합하는 단음절 의존명사의 처리이다. 이 경우도 의존명사 '것'과 마찬가지로 결합형 구문의 분석을 위한 사전 LGG를 구축함으로써 해결할 수 있다. 이에 대한 그래프가 다음 그림 8에 제시되어 있다.

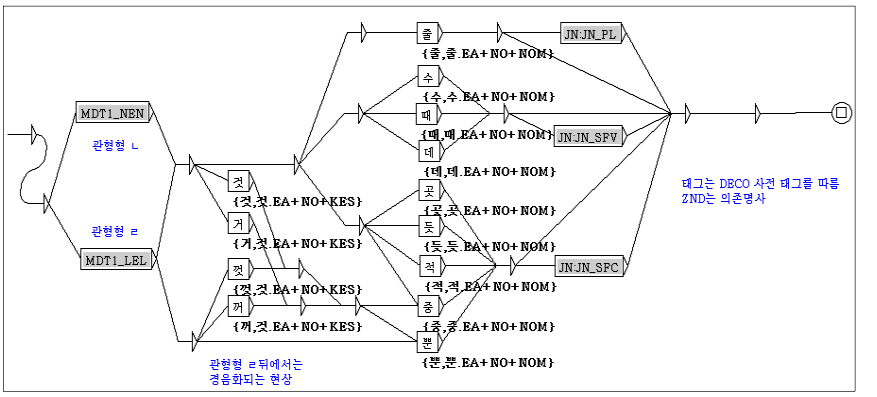

그림 8. 관형절 뒤에 오는 의존명사와 관련된 LGG

그림 8의 LGG는 '줄, 수, 때, 데, 곳, 듯' 등의 단음절 의존명사들이 결합하는 복합구성을 기술한다. 즉 관형절과 결합하는 의존명사 포함 형태들로서, 뒤에 JN_PL(종성 'ㄹ' 뒤에 결합할 수 있는 복합조사 구성), JN_SFV(종성이 없는 명사와 결합할 수 있는 복합조사 구성), JN_SFC(종성 'ㄹ'을 제외한 모든 종성을 가진 명사와 결합할 수 있는 복합조사 구성)를 통해서 의존명사와 결합하는 조사 그래프가 연이어 결합할 수 있게 구조화하였다. 위와 같은 LGG를 통해서 관형절과 띄어쓰기가 안된 단음절 의존명사의 선행요소와 후행요소를 모두 인식하는 문법이 제공된다.

● 유형 5. 특수문자 및 이모티콘 구문의 처리

SNS 텍스트에서는 이모티콘을 결합하여 자신의 감정을 표출하는 특성이 빈번히 나타난다. 한국어에서는 주로 'ㅎ'나 'ㅋ'를 이용하여 웃음소리를 표현하거나, 'ㅜ, ㅠ' 등을 통해 울음을 형상화하여 글쓴이의 마음을 표상한다.[8] 또한 '♡'과 같은 그림문자나 'ㄱ'과 같이 상대적으로 빈도가 높지 않은 이모티콘 문자열도 관찰된다. 이러한 특수문자들은 긍정 또는 부정의 극성을 가질 수 있으므로, 전처리 단계에서 PGT를 이용해 소거하거나 치환시키지 않고 이 자체를 인식하여 일정 의미정보를 태깅할 수 있도록 특수 사전적 처리를 하는 것이 바람직하다. 실제 SNS 텍스트에서 나온 예는 (31)과 같다.

(31) a. 웅성우 '활활 (Burn it up)' 직캠 by 도라삐 @ 인천 스카이페스티벌
b. 투명 동그리 1장 + 포카 2장 = 1500원
c. 수상축하# 최고의스타# 황치열 매일듣는노래##.

위의 예에 나타난 부호들처럼 감성과 직접 관련이 없는 유형들이 존재하는가 하면, 다음 (32)와 같이 일정 감정을 표현하는 부호들이 사용된 예들이 존재한다.

(32) a. 1화 등장부터 남다른 제하씨 ~♡♡.
b. 거기가 본점이었나 인기많거든요 ㅋㅋ ㄱㄱ.
c. 제주 워너블 ㅎㅎㅎㅎㅎ
d. ㅅㅂ 미쳤나봐 미친시발
e. 미친 제주 、、、、 ㅜㅜㅜㅜㅜㅜㅜㅜㅜㅜㅜㅜㅜㅠ.

8) 그런데 이러한 자모음이 나타날 경우 다음과 같은 문제가 나타나기도 한다.
(i) 자음 이모티콘 오류 유형: a. 재밋엇닼ㅋ / b. 그랳ㅎㅎㅎ
(ii) 모음 이모티콘 오류 유형: a. 그랬으뮤ㅠㅠ / b. 완전감도우ㅜ
즉 (i)처럼 무종성 단어 뒤에 ㅋ이나 ㅎ과 같은 자음이 연이을 때 종성 위치에 삽입되거나 (ii)처럼 유종성 단어 뒤에 ㅠ나 ㅜ와 같은 모음이 연이을 때 새로운 음절을 구성하게 되는 경우로서, 이들은 본 연구에서는 DecoTex 플랫폼에서 별도의 모듈을 통해 올바르게 분리되도록 정규화되었다.

(32a)는 긍정적 감정을 '하트 ♡'로 표현하고 있고, (32b)에서는 '인기 많다'는 내용에 웃음소리를 나타내는 'ㅋㅋ'와 '가자 가자(go go)'를 한글 자음으로 표상한 'ㄱㄱ'이 나타나 긍정의 극성을 추가한다. (32c)은 웃음소리의 'ㅎ'가 연속된 형태로 긍정을 표시하고 있고, 반대로 (32d)에서는 욕을 표현하는 '시발'의 초성만을 표시한 'ㅅㅂ'이라는 한글 자음이 부정적인 감정을 표현하고 있다. (32e)에서도 울음을 표시하는 'ㅜ'와 'ㅠ'를 섞어서 부정적 감정을 표현한다.

이러한 특수문자들은 오피니언 마이닝 응용분야에서 실제 분석에서 중요한 키워드가 될 수 있으므로, 사전에 극성을 지니는 특수문자를 미리 지정해 놓으면 텍스트 분석에 있어 높은 정확도를 기대할 수 있다. 특수문자와 이모티콘들은 LGG를 통해 극성 정보를 기술하여 이를 통해 올바른 정보가 부착될 수 있도록 하였다.

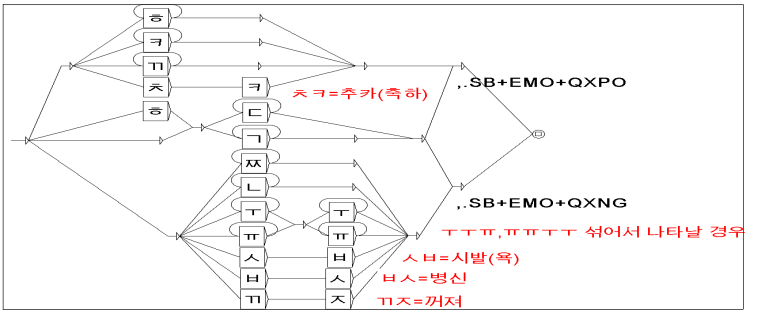

그림 9. 이모티콘에 대한 그래프 문법

예를 들어, 그림 9의 LGG는 'SB+EMO+QXPO'가 기호(SB) 범주에서 이모티콘(EMO), QXPO(긍정)라는 사전적 분류 정보를 표현하는 유한상태 트랜스듀서(FST)를 보인다. 마찬가지로 아래의 QXNG는 부정을 의미한다. 이는 이모티콘 형식으로 긍정 또는 부정을 표현할 수 있는 형태들의 사전적 정보가 된다.

5.2.2. DECO사전의 LEX 리스트 방식에 의한 처리

이 장에서는 SNS 상에서 추출된 신조어 혹은 외래어 표현을 DECO사전의 표제어로 확장 등재하는 방법을 논의한다. SNS의 특성상 새로운 표현이 지속적으로 발생하기 때문에 빈번히 등장하는 표현을 지속적으로 추가하는 것이 필요하다. 유니텍스 플랫폼에 기반을 둔 DECO사전에서는 미분석 토큰들을 확인하고 이들을 사전 목록으로 작성한 뒤에 다시 적용할 수 있는 부트스트랩 방식을 사용하고 있어서 SNS 텍스트의 빠른 변화를 감지하고 반영할 수 있다. 여기에서는 유니텍스 플랫폼에서 미분석 토큰들의 유형을 확인하고 이를 토대로 어휘 표제어를 작성하고 확장하였다.

● 유형 6. '외래어 · 신조어' 등 명사 표제어의 확장

신조어의 대부분이 명사 범주로 실현되므로 본 연구에서 SNS 텍스트 분석을 통해 1,160개의 명사 목록을 새로 추가하였다. 크게 외래어로 구성된 개체명, 외래어와 한글 및 숫자가 결합된 개체명, 개체명 변형, 측량 표현 등으로 유형을 나누어 볼 수 있다. 먼저 새로 나오는 개체명의 경우가 많은데, 대부분 외래어이거나 외래어와 한글, 숫자 혼합으로 이루어져 있다.

(33) a. 외래어(전사표기 및 로마자)
아이패드, 아이폰, LG디스플레이, 기가헤르츠
b. 외래어+한글
DMB기능, 배터리깡패, 스마트폰용, 스크린활용도, 이어폰음질, 짝퉁맥북
c. 외래어+숫자
아이패드3, 아이폰7, 옵뷰2
d. 외래어+한글+숫자
G8기능, Gpro7전용

위와 같이 대부분 유형은 외래어가 포함된 개체명인데, 도메인에 따라서 '쎄븐(아이폰7), 갤(갤럭시)' 등으로 축약해서 나타나는 경우가 있다. 이러한 축약 형태 역시 리스트에 포함되어 사전에 등재되었다. 한편, 일부 개체명을 변형하여 조롱이나 비난의 뜻을 표현하는 경우도 나타난다. 다음의 예 (34)를 보자.

(34) 헬레기전자(LG), 옴레기(옴니아), 삼엽충(삼성 사용자), 앱등이(애플 사용자), 네이년(네이버), 엘지빠순이(LG 사용자), 샘송빠돌이(삼성 사용자), 삼까앱등(삼성을 까는 애플 사용자)

위에서 나온 예들과 같이 상대 제품을 사용하는 사용자들이나 제품, 회사를 비난하기 위한 용어들은 감성분석에서 중요한 키워드로 작용할 수 있는 부분이기 때문에 이들 목록도 별도의 표제어로 처리되었다. 이외에도 한 글자 단어이기 때문에 붙여서 쓰기 쉬운 형태이거나 SNS에서만 사용하는 신조어들의 예도 볼 수 있다.

(35) a. SNS 신조어
개쓰렉, 병신타령, 용팔이
b. 축약 형태
개취(개인취향), 남팬(남자팬)

예를 들면 (35)는 현재 SNS 텍스트에서 빈번히 출현하는 형태들인데, 사전에 이런 유형의 어느 수준까지 표제어로 확장해야 하는가에 대한 논의는 쉽게 합의되기 어려우나 처리의 효율성을 위해서는 이들을 특수 사전에 수록하는 형식이 바람직하다고 판단되어 별도의 목록을 구성하였다.

● 유형 7. 그 외 품사의 어휘 목록의 확장

우선 형용사에서는 다음과 같은 어휘들이 추가되었다. 이들은 단일 형용사라기보다는 다른 단어와 결합하여 실현된 합성어나 구 형식의 표현도 포함된다. 형용사의 예는 (36)과 같다.

(36) 가격차이없다. 개아깝다, 어쩔수없다. 고려없다, 고장없다, 무리없다, 성능좋다, 애플스럽다, 이상없다, 전혀없다, 중독성있다, 짝퉁답다, 책임감있다, 혁신없다

한편 동사의 예는 (37)과 같다.

(37) 갈아타보다, 괜찮아보이다, 되도않다, 고급티내다, 용서못하다, 정붙이다, 욕처먹다, 국뽕맞다

특히 외래어가 결합되어 생성되는 동사들도 빈번하다.

(38) 재부팅되다, 튜닝되다, 프로그래밍하다, 에이에스받다, 캐리해주다

부사의 예는 (39)와 같다.

(39) 기냥, 긍께, 거기서거기, 이빠이, 그래봤자, 오질나게, 카아악, 카카, 크, 히융, 훠얼씬

대부분 표준어에 어긋나거나 의도적으로 오탈자를 내는 경우들, 그리고 과장된 표현 등으로 이루어져 있다. 현재 이들은 일반적으로 사전에서 인식되지 않는 유형들로서, DECO사전에서는 특수사전 모듈을 통해 이들을 수록하는 방식을 취하였다.

지금까지 5장에서는 미분석어에 대한 전체적인 유형과 이에 대한 해결 방법을 모색하였다. 크게 두 가지 방향으로 처리 방법을 분류하였는데, 우선 전처리단계에서 텍스트 자체를 수정하는 것이 바람직한 유형은 PGT의 정규식을 통해 변환하고, 그 외 문법 정보를 활용하여 사전에 기술함으로써 상세 정보와 함께 분석되도록 하는 방식이 효과적인 유형은 DECO사전과 LGG 구축을 통해 해결하였다.

6. DecoPGT의 적용 실험 및 성능 평가

6.1. DecoPGT 성능 실험 절차

이 장에서는 본 연구에서 구축한 DecoPGT의 성능을 평가하기 위해 기존의 전처리 관련 프로그램들과의 비교 실험을 수행하였다. 실험을 위해 SNS 텍스트에서 특히 IT, 성형, 게임, 생활 도메인을 중심으로 무작위로 100문장을 추출하여 샘플을 구성하였다. 실험도구로 DecoPGT 외에 네이버 맞춤법 검사기[9], 다음 맞춤법 검사기[10], 부산대 인공지능연구소 맞춤법/문법 검사기[11]를 사용하여 결과를 비교하였다. 다음은 본 연구의 실험 절차를 도식화 한 것이다. 다른 맞춤법 교정기의 경우도 같은 방법으로 성능을 확인하였다.

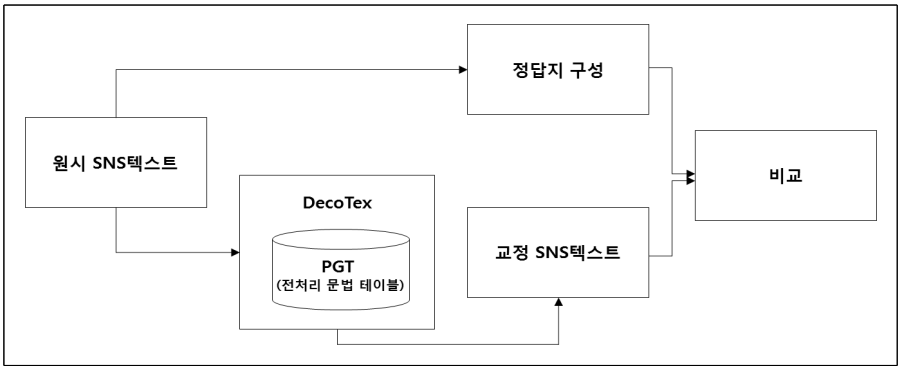

그림 10. DecoPGT 실험 절차

위 그림에서 보듯이 실험 절차는 맞춤법 교정기에 원시 SNS 텍스트를 입력하여 교정된 결과를 얻고 이를 정답지와 비교하여 정확률, 재현율, F-score 등을 구하는 것으로 성능을 확인한다. 이때 정답지는 원시 SNS 텍스트에 대하여 사람이 직접 확인하여 구성한다. 그리고 각 시스템별 성능을 평가한 뒤 비교하여 SNS 텍스트에 대해 본 연구에서 구축한 PGT의 성능이 어느 정도 의미를 보이는지를 확인하였다.

6.2. SNS 텍스트 실험 결과 분석

실험 내용은 PGT를 적용한 DecoTex와 더불어 추가적으로 세 가지 맞춤법 검사기에 대

9) https://www.naver.com/에서 '네이버 맞춤법 검사기' 검색 결과에 제공. 특정 링크를 제공하지 않으며, 500자 이하 검사 가능.
10) http://dic.daum.net/grammar_checker.do 제공. 1000자 이하 단위로 검사 가능.
11) http://speller.cs.pusan.ac.kr/제공. 검사 단위는 제한하지 않음.

한 성능을 확인하는 것이다. 샘플로 사용한 100문장의 경우 문제가 되는 미분석어 유형이 총 270 어절이었으며, 이를 바탕으로 정답지를 만들고 각각의 맞춤법 검사기의 결과와 비교하여 정확히 수정하였는지 확인하였다. 실험 결과는 다음의 표 5와 같다.

표 5. 성능 실험 결과

	변환대상	변환부분	정답부분	Precision	Recall	F-score
네이버 맞춤법검사기	270	191	115	60.20%	42.59%	49.89%
다음 맞춤법검사기	270	208	146	70.19%	54.07%	61.08%
부산대 맞춤법검사기	270	288	169	58.68%	62.59%	60.57%
DecoTex (PGT)	270	128	127	99.21%	47.03%	63.81%

표 5는 각각의 맞춤법 검사기에 100문장을 입력하여 변환된 어절과 정답지가 일치하는지 확인하고 각각의 수치에 따라 Precision(정확률), Recall(재현율), F-score를 구한 것이다. 이 때, 'ㅋㅋㅋ', 'ㅎㅎㅎ' 등의 이모티콘 표현 형태는 각 검사기마다 처리 방식이 달라서 정답 판단에서 제외하였다. 검사를 거친 결과물의 경우, 네이버와 다음 맞춤법 검사기는 하나의 수정 결과를 제공하기 때문에 그대로 사용하여 비교하였다. 부산대 인공지능연구소의 맞춤법/문법 검사기의 경우는 문제가 되는 부분을 대치어로 고를 수 있도록 하는데 1순위로 지정된 대치어를 기준으로 정답을 판단하였으며, 대치어 없이 '?' 등으로 치환되거나 의미가 있는데 삭제되는 부분은 틀린 것으로 간주하였다.

전반적으로 결과에 나타난 특징을 살펴보면, DecoTex의 경우는 정확하게 지정된 표현만을 수정하기 때문에 변환 부분은 적지만, 수정한 대상이 거의 정답임을 알 수 있다. 단 하나의 어절에서만 잘못된 수정 결과를 출력하였는데, '몰겟네용'에서 '겟네용'부분만을 수정하고 '몰'을 '모르'로 수정하는 부분은 재현율의 한계로 발생한 오류였다. 반면 나머지 맞춤법 검사기들의 경우는 DecoTex와 비해 더 많은 부분을 변환하였으나, 틀린 부분의 비중이 상대적으로 매우 높다는 것을 확인하였다. 이를 통해 SNS 텍스트에 대한 고려가 미흡하다는 것을 확인할 수 있다.

본 연구에서 DecoTex를 통해 구축된 PGT의 경우 휴리스틱한 방식을 배제하였기 때문에 정확율이 매우 높을 것이라는 것은 충분히 예상되었으나 동일한 이유로 재현율의 비율이 타 검사기에 비해 현저히 낮을 것으로 우려되었다. 그러나 네이버 검사기보다도 높은 재현율을 보였으며, 전체적으로 예상보다 그렇게 낮지 않은 재현율을 나타낸 것이 중요한 의의를 보인다고 판단된다. 궁극적으로 전체 성능의 지표가 되는 F-Score에서는 타 검사기 3가지보다 가장 높은 성능을 보였음을 확인할 수 있다.

다음은 타 맞춤법 검사기에서 잘못 수정된 예의 일부를 보인다.

표 6. 수정 오류 형태

대상	네이버 맞춤법검사기	다음 맞춤법검사기	부산대 맞춤법검사기
마꿀이유를(바꿀 이유를)	*메꿀 이유를	*마 꿀 이유를	*마꿀이 유를
괜찮은듭(괜찮은 듯)	*괜찮은 등	괜찮은 듯	*?
덕질할때는(덕질할 때는)	*덕진 할 때는	덕질 할 때는	*팬 활동할 때는
갤노7	*갤로 7	갤노 7	*갤로 7
시러짐(싫어짐)	*스러짐	*싫어 짐	싫어짐
가잔아여(가잖아요)	*가잔 아예	*가 잔아 여	가잖아요
같에영(같아요)	*같에요	*같아 영	*같아 영
거애요(거예요)	*거래요	거예요	거예요
마니(많이)	많이	많이	*머니
곳가서(곳 가서)	*곧 가서	곳 가서	곳 가서
개호구로	*개표구로	*개호 구로	*간호 구로
댓는데(됐는데)	*댔는데	*댓는데	*댔는데
많태여(많데요)	*많 때여	*많태여	*많태여
잘되따고(잘됐다고)	*잘되 따로	*잘되다고	*잘 됐다고
조아용(좋아요)	*조아영	*조아요	좋아요
깔앗는대(깔았는데)	*깔아는 대	깔았는데	*깐 대
부렀어(버렸어)	*부렀어	*불었어	버렸어
업시염(없어요)	*업에 염	*업시요	*없이 염
팀운은(팀 운은)	*팀원은	*팀원은	팀 운은
공먼연금까지(공무원 연금까지)	*공만 연금까지	*공 먼 연금까지	공무원 연금까지
넘시러(너무 싫어)	*넘기러	*넘시러	너무 싫어

위의 표 6에서 보듯이, 음절은 올바르게 변형하여도 최종적으로 띄어쓰기가 잘못된 '시러짐-->*싫어 짐' 같은 경우는 틀린 것으로 간주하였고 '덕질할때는-->*팬 활동할 때는'처럼 과하게 의미로만 표현한 경우도 틀린 것으로 판단하였다. 그리고 수정되지 않은 것에도 비교를 위해 오류표시(*)를 해두었다. 표 7의 대상들은 SNS 텍스트에서 자주 발생하는 띄어쓰기 문제나 철자법 오류 형태들이 대부분이며, 이러한 비정형 텍스트의 경우 '정확하게' 수정하지 않으면 오히려 본래 문장을 더 훼손시킬 수 있다는 점을 확인할 수 있다. 즉, SNS 텍스트 교정에서는 오류율이 높은 재현율 위주의 전처리보다는 정확률이 중요하며, PGT를 구성하여 문제 형태를 정확하게 교정하는 것이 유의미하다는 것을 확인할 수 있다.

7. 결론

지금까지 SNS 텍스트의 형태소 분석과 관련하여, 미분석어가 중요한 문제가 됨을 밝히고 이러한 문제가 자연어 처리 전반에 걸림돌이 되고 있음을 논의하였다. 문제의 심각성에 비해 최근 전처리 관련 연구는 자연어 처리 분야에서도 많이 논의되지 않는 문제라는 점을 인식하여 본 연구가 시작되었다.

본 연구에서는 텍스트 자체에서 수정이나 변환이 이루어져야 할 미분석어 유형을 PGT 정규식 방식으로 처리하는 단계와, 해당 어절의 문법정보를 고려하여 사전 층위에서 기술되거나 처리되어야 할 미분석어 유형을 DECO사전과 LGG 방식으로 기술하는 단계로 이원화하는 방법론을 제시하였다.

그리고 실험을 통해 채택한 방법론의 유용성을 확인하는 과정을 거쳤다. 실험은 DecoPGT를 적용한 전처리 텍스트를 네이버, 다음, 부산대 인공지능연구소의 맞춤법 검사기에 적용하여 문제가 되는 형태를 얼마나 교정하는지 비교 · 확인하는 방식으로 진행하였다. 그 결과 PGT를 적용한 텍스트는 정확률(99.21%), 재현율(47.03%), F-score(63.81%)로 매우 높은 정확률을 보여주었고, 전체적인 F-score에서 네이버(49.89%), 다음(61.08%), 부산대 맞춤법 검사기(60.57%) 이상의 성능을 보인다는 것을 확인하였다. 이를 통해 SNS 텍스트를 분석하는 데에 본 연구에서 제안하는 PGT 방법론의 성능이 유효함을 입증하였다.

추가적으로 이와 같은 DecoPGT를 기반으로 하는 본 연구의 전체 성능을 확인하기 위해 무작위로 추출한 800문장 규모의 SNS 코퍼스에 형태소분석을 수행하여 궁극적으로 미분석어 처리 성능이 얼마나 개선되었는지 최종 실험하였다. 효과적인 검증을 위해 기존의 형태소 분석기인 글잡이[12], 꼬꼬마[13] 등 분석기에서도 동일한 DecoPGT를 적용하여 텍스트를 개선한 후 그 결과를 비교하였다. 다음의 그림 11을 보자.

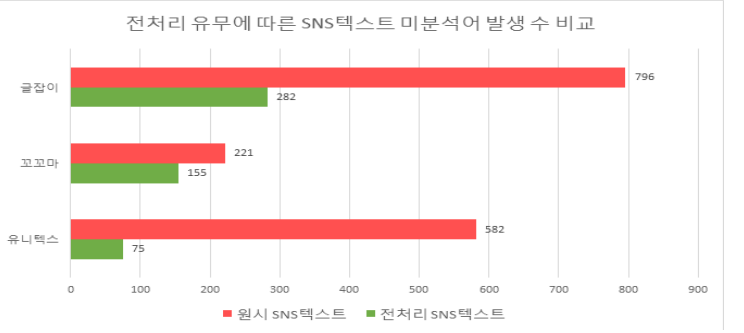

그림 11. 전처리 유무에 따른 SNS 텍스트 미분석어 발생 수 비교

12) https://ithub.korean.go.kr/국립국어원 언어정보나눔터에서 제공.
13) http://kkma.snu.ac.kr/ 서울대 IDS 연구실에서 자바 라이브러리로 제공.

위 그림 11에서 PGT를 적용하여 개선한 텍스트를 입력한 경우, 글잡이나 꼬꼬마와 같은 형태소 분석기에서도 성능이 향상된 것을 볼 수 있다. 글잡이의 경우 미분석어 발생율이 35% 수준으로 감소하였고(282/796), 꼬꼬마 분석기의 경우 70% 수준으로 감소한 것을 볼 수 있다(155/221). 본 연구에서는 미분석어 비율이 13% 수준으로 상대적으로 우수한 성능을 보이고 있다(75/582). 또한 이와 같이 감소한 비율에서뿐 아니라 최종적으로 발생한 미분석어의 개수에 있어서도 글잡이 282개, 꼬꼬마 155개에 비해 본 연구에서는 75개로 가장 적은 수로 나타난 것을 확인할 수 있다. 이는 단지 PGT의 적용뿐 아니라 형태소분석을 위한 DECO사전과 LGG 확장이 지속적으로 이루어질 수 있는 본 시스템의 차별적인 특징에 힘입은 것으로 보인다.

추후 연구에서 현 단계의 재현율을 보다 효과적으로 향상시킬 수 있는 방법론에 대한 연구가 현재 계속 진행 중이며, 이는 후속 연구에서 보다 본격적으로 소개될 수 있을 것으로 기대된다.

참고문헌

김선호, 윤준태, 송만석. (2002). 한국어 문서 처리를 위한 동적 생성 로컬 사전 기반 미등록어 분석. *정보과학회논문지: 소프트웨어 및 응용, 29*(6), 407-416.

남길임. (2016). 상품평 텍스트에 나타난 감성표현 연구 -감성분석과 국어학 연구의 접점. *언어과학연구, 78*, 101-123.

남지순. (2010). *Korean Electronic Dictionary DECO, DICORA-TR-2010-02.* 한국외국어대학교 디코라연구센터.

남지순. (2013). 모리스 그로스의 언어처리 모델과 전산학적 적용의 이해. *인문언어, 15*(1), 125-151.

박봉래, 황영숙, 임해창. (1998). 용례 분석에 기반한 미등록어의 인식. *정보과학회논문지, 25*(2), 397-407.

박소영. (2008). 웹문서에서의 출현빈도를 이용한 한국어 미등록어 사전 자동 구축. *한국컴퓨터정보학회논문지, 13*(3), 27-33.

박영준. (1994). *현대국어의 국어사적 연구.* 국학자료원.

배주채. (2017). 교체의 개념과 조건. *국어학, 81*, 295-324.

양장모, 김민정, 권혁철. (1996). 언어정보를 이용한 한국어 미등록어 추정. *한국정보과학회 봄 학술발표논문집, 23*(1), 957-960.

이도길, 이상주, 임해창. (2003). 명사 출현 특성을 이용한 효율적인 한국어 명사 추출 방법. *정보과학회논문지: 소프트웨어 및 응용, 30*(2), 173-183.

이세희, 김학수. (2009). 음절 통계를 이용한 경량화된 철자 오류 교정 모델. *한국정보과학회 학술발표논문집, 36*(1), 84-85.

차정원, 이원일, 이근배, 이종혁. (1997). 형태소 패턴 사전을 이용한 일반화된 미등록어 처리. *정보과학회 인공지능연구회 춘계학술대회 논문집*, 37-42.

Gross, M. (1997). The construction of local grammars. In E. Roche & Y. Schabes (Eds.), *Finite-State Language Processing* (pp. 329-354). MA: The MIT Press.

Gross, M. (1999). A bootstraph method for constructing local grammars. In *Proceedings of the Symposium Comtemporary Mathematics*, 229-250. University of Belgrad.

Lee, S. (1995). A Korean part-of-speech tagging system with handling unknown words. In *Proceedings of International Conference on Computer Processing Pacific Rim Symposium,* 89-94.

Liu, B. (2012). *Sentiment analysis and opinion mining*. Morgan and Claypool Publishers.

Mikheev, A. (1996). Unsupervised learning of word-category guessing rules. In *Proceedings of 34th ACL*, 327-334.

Nagata, M. (1996). Automatic extraction of new words from Japanese texts using generalized forward-backward search. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, 48-59.

Nam, J. (2015). *Korean Electronic Dictionary DECO TR-2015-02,* DICORA. Seoul, Hankuk University of Foreign Studies.

Park, B., & Rim, H. (1995). A Korean corpus refining system based on automatic analysis of corpus. In *Proceedings of Natural Language Processing Pacific Rim Symposium*, 89-94.

Paumier, S. (2003). *De la reconnaissance de formes linguistiques a l'analyse syntaxique.* Unpublished doctoral Dissertation, Univ. of PEMLV, France.

Weichedel, R., Meteer, M., Schwartz, R., Ramshaw, L., & Palmucci, J. (1993). Coping with ambiguity and unknown words through probabilistic models. C*omputational Linguistics*, *19*(2), 359-382.

Yoo, G., & Nam, J. (2017). DecoTex users' mannual DICORA-TR-2017-12. Version V01-2017MAY. Hankuk University of Foreign Studies.

최성용
17035 경기도 용인시 처인구 외대로 81
한국외국어대학교 인문대학 언어인지과학과
전화: (031)330-4349
이메일: csy@hufs.ac.kr

신동혁
17035 경기도 용인시 처인구 외대로 81
한국외국어대학교 인문대학 언어인지과학과
전화: (031)330-4349
이메일: sdh876@hufs.ac.kr

남지순
17035 경기도 용인시 처인구 외대로 81
한국외국어대학교 인문대학 언어인지과학과
전화: (031)330-4349
이메일: namjs@hufs.ac.kr

Received on October 30, 2017
Revised version received on December 13, 2017
Accepted on December 31, 2017