임베딩 교체에 따른 구어체 텍스트 탐지 모델 성능 비교

Performance Comparison of Spoken Language Detection Model with Embedding Replacement

요 약

딥러닝 기반 욕설 탐지 모델은 한국어 텍스트에서 빈번하게 나타나는 오탈자 및 띄어쓰기 오류로 인해 정확도 향상에 많은 제약이 있다. 특히, 구어체를 사용할 경우 학습 데이터 생성을 위한 형태소 분석 과정에서 단어의 의미 파악을 방해하는 형태소가 빈번하게 추출되는 문제점이 있으며, 이는 욕설 탐지 모델의 정확도를 떨어뜨리는 가장 큰 요인이 된다. 본 논문에서는 이러한 한국어 구어체 텍스트의 문제점을 극복하기 위해, 임베딩(embedding)에 따른 탐지 모델을 설계 및 구현하고, 이를 기반으로 욕설 탐지 정확도를 비교하고자 한다. 탐지에는 Word2Vec, fastText, SKT-KoBERT, KoELECTRA의 총 네 가지 임베딩 모델을 사용하였으며, 실험을 통해 각 임베딩 기반 욕설 탐지 모델 성능을 비교 및 평가한다. 실험 결과, 사용 문자 단위에 따른 실험은 Word2Vec과 fastText 모두 90% 이상의 정확도를 보였고, 중의성 판단 여부에 따른 실험에서는 SKT-KoBERT가 fastText에 비해 월등히 높은 성능을 보이는 것으로 나타났다. 마지막으로, 사전 학습 방법에 따른 실험 또한 SKT-KoBERT가 KoELECTRA에 비해 높은 성능을 보이는 것으로 나타났다. 본 논문의 실험 결과를 통해, 다양한 구어체 기반 딥러닝 서비스에 보다 효과적인 임베딩 기술을 적용할 수 있을 것으로 사료된다.

1. 서론

일반적으로, 문어체는 띄어쓰기, 맞춤법 등 한국어 표준 어법을 잘 따르는 반면 구어체는 한국어 표준 어법에 어긋나는 경우가 많다. 이러한 구어체를 특정 단어 탐지 딥러닝 모델 학습에 사용할 경우, 학습 데이터 정제 과정에 많은 어려움이 발생한다. 특히, 딥러닝 모델 학습에 사용할 한국어는 형태소 단위까지 구분하는 전처리 과정이 필요하다. 그러나, 오탈자 및 띄어쓰기 오류가 빈번한 구어체가 이러한 형태소 분석을 거칠 경우 의미가 불분명한 데이터가 빈번하게 생성된다. 구어체를 학습 데이터로 사용하게 되면, 특정 단어나 문맥을 탐지, 분류하는 딥러닝 모델의 정확도가 크게 저하된다. 이 문제는 대부분의 학습 데이터가 구어체인 욕설 탐지 모델에도 해당된다.

욕설 탐지란 문장에서 욕설의 정의에 부합하는 텍스트 유무를 탐지하는 것으로, 최근 사회적 이슈가 되고 있는 사이버 언어 폭력에 의해 그 필요성이 크게 증가하는 추세이다. 이러한 사이버 언

어 폭력을 방지하기 위해 포탈, 게임, 인터넷 방송 등에서는 자체 개발한 댓글 및 채팅 욕설 탐지 기능을 제공하고 있다[1]. 초기 욕설 탐지 시스템은 욕설 사전 기반 서비스를 제공했기 때문에 사전에 없거나 우회적인 욕설을 제대로 탐지하지 못한다. 이를 해결하기 위해 딥러닝 기반 욕설 탐지 기법이 등장하였으나, 대부분의 온라인 서비스에서 생성되는 구어체 텍스트때문에 탐지 정확도 향상에 어려움을 겪고 있다. 최근 SNS, 채팅, 커뮤니티 사이트 등에서 대량 생성되고 있는 한국어 텍스트는 대표적인 구어체의 예시로, 이에 대한 욕설 탐지는 매우 중요한 연구 주제이다.

본 논문에서는 구어체 텍스트의 문제점을 해결하기 위해, 임베딩 모델에 따른 세 가지 한국어 욕설 분류 모델을 제시한다. 이때, 모델을 세 가지로 구분한 이유는 (1) 사용 문자 단위, (2) 단어 중의성 판단 여부, (3) 사전 학습 방법 중 어떤 요소가 구어체 탐지 성능에 가장 큰 영향을 끼치는지를 비교하기 위함이다. 실험에는 Twitch 채팅, 유튜브 자막, 디시인사이드 일부 갤러리의 글 제목을 크롤링하여 구축한 한국어 데이터를 사용한다. 첫 번째 실험은 사용 문자 단위가 탐지 결과에 어떤 영향을 끼치는지를 비교하기 위한 실험으로, Word2Vec과 fastText를 비교 및 평가한다. 실험에서는 임베딩 단계와 탐지 단계를 포함한 전체 모델 학습에 80%의 데이터를 사용하고, 사용하지 않은 20%로 테스트를 진행한다. 이때, Word2Vec의 경우 새로운 데이터에 대한 적응력이 없기 때문에 임베딩 모델 학습에 데이터 전체를 사용한다. 두 번째 실험은 문장 내에서 단어의 중의성 판단 여부를 비교하기 위한 실험으로, fastText와 SKT-KoBERT를 비교한다. 두 모델은 새로운 데이터에 대해 적응 가능하므로 80%의 데이터만 임베딩 학습에 사용한다. 또한, 첫 번째 실험과 마찬가지로 임베딩과 탐지 단계의 학습에는 80%의 데이터를 사용하며, 나머지 데이터 20%로 테스트를 진행한다. 세 번째 실험은 사전 학습 방법에 따른 탐지 정확도 평가를 위한 실험으로, SKT-KoBERT와 KoELECTRA 모델을 비교한다. 임베딩 모델 및 전체 모델 학습과 테스트에 사용된 데이터의 비율은 두 번째 실험과 동일하게 진행한다. 실험 결과, 사용 문자 단위가 다른 첫 번째 실험에서는 문자 단위와는 상관없이 두 모델 모두 93% 이상의 탐지 정확도를 보였으며, 단어 중의성 판단 여부가 다른 두 번째 실험에서는 SKT-KoBERT가 fastText에 비해 9% 이상 높은 정확도를 보였다. 사전 학습 방법에 따른 세 번째 실험에서는 SKT-KoBERT가 KoELECTRA에 비해 4%

이상 높은 정확도를 보였다.

본 논문의 공헌은 다음과 같다.

(1) 구어체 텍스트의 특징과 문제점을 제시하고, 이로 인한 딥러닝 모델의 탐지 정확도 향상의 한계점을 설명한다.

(2) 구어체 텍스트 대한 딥러닝 모델의 탐지 정확도 향상 방안을 제시하기 위해, 조건이 다른 세 가지 욕설 탐지 모델을 설계한다.

(3) 실제 수집한 데이터를 통해 각 모델의 실험을 진행하고, 탐지 정확도 향상을 위해 어떤 요소가 큰 영향을 미치는지를 분석 및 평가한다.

본 논문의 구성은 다음과 같다. 제 2절은 본 연구의 배경이 되는 임베딩 모델과 욕설 탐지에 대해 설명한다. 제 3절은 구어체 텍스트의 특징과 문제점을 분석하고, 이로 인한 탐지 정확도 향상 한계에 대해 설명한다. 제 4절에서는 실험을 위해 수집된 데이터의 예시와 데이터 레이블링의 방법 및 기준에 대해 설명하고, 임베딩에 따른 한국어 욕설 탐지 모델을 설계한다. 제 5절은 각 설계에 대해 실험 및 비교 평가를 진행한다. 마지막으로, 제 6절에서 실험 결과를 바탕으로 결론을 내린다.

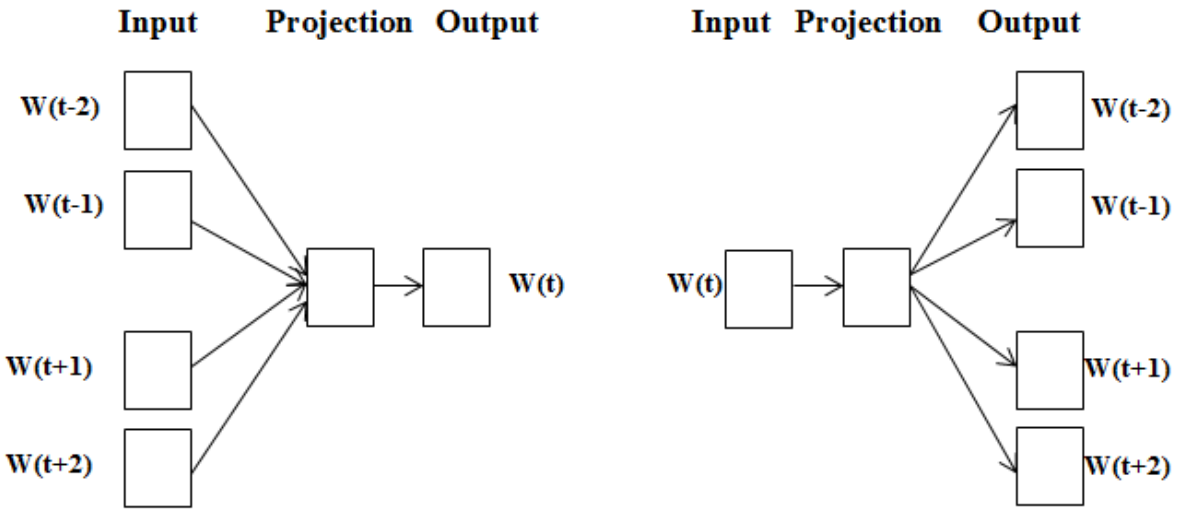
2. 관련 연구

2.1 임베딩 모델

딥러닝 기술의 등장으로 인해, 문장의 특징이나 문맥, 혹은 특정 단어의 유무를 추출하기 위한 많은 연구들이 이루어졌다. 특히, 텍스트 데이터의 학습과 분류를 위해 텍스트를 숫자 데이터로 변환하는 임베딩 기법들이 활발하게 연구되었다. 초기 단어 임베딩에 많이 사용된 기법은 Word2Vec으로, 단어를 벡터화하는 알고리즘이다. Word2Vec은 중심 단어와 비슷한 주변 단어들을 맞추거나, 주변 단어들을 통해 중심 단어를 더 잘 맞추기 위해 가중치 행렬을 업데이트하며 단어 벡터를 학습한다. 이러한 Word2Vec에는 그림 1과 같이 중심 단어로 주변 단어를 예측하는 Skip-Gram[2]과 주변 단어로 중심 단어를 예측하는 CBOW(continuous bag of words)[2]의 두 가지 모델

이 있다. 그러나, 이 두 모델은 소프트맥스를 사용하기 때문에 학습 데이터가 커질수록 계산량이 폭증한다는 단점이 있다. 이들의 계산량 감소를 위해, Word2Vec에서 소프트맥스 연산 시 전체 단어를 대상으로 하지 않고 확률적으로 일부 단어만 샘플링하여 계산하는 네거티브 샘플링(negative sampling)을 도입[3]하였다. 이때 한 단어 w_i가 네거티브 샘플로 뽑힐 확률 $P(w_i)$는 식 (1)과 같이 정의된다. 여기에서, $f(w_i)$는 (단어 w_i의 빈도/전체 단어 수)를 의미한다.

$$P(w_i) = \frac{f(w_i)^{3/4}}{\sum_{j=0}^{n} f(w_i)^{3/4}} \tag{1}$$



(a) CBOW 모델 (b) Skip-gram 모델

그림 1. Word2Vec의 대표 모델.

이러한 Word2Vec은 주변 단어로 중심 단어를 예측하는 특성 때문에 주어진 단어 집합에 포함되지 않은 단어에 대해서는 예측 정확도가 떨어진다. 이를 해결하기 위해, Word2Vec을 확장한 fastText[4]가 등장했다. fastText는 단어를 구성하는 보조 단어를 벡터로 변환하고, 이 벡터의 합으로 필요한 단어 벡터를 표현하는 기법이다. 특히, 데이터의 기본 단위가 단어인 Word2Vec과 달리 학습 데이터의 각 단어를 그림 2와 같은 문자 단위(n-gram)로 구분해서 사용한다.

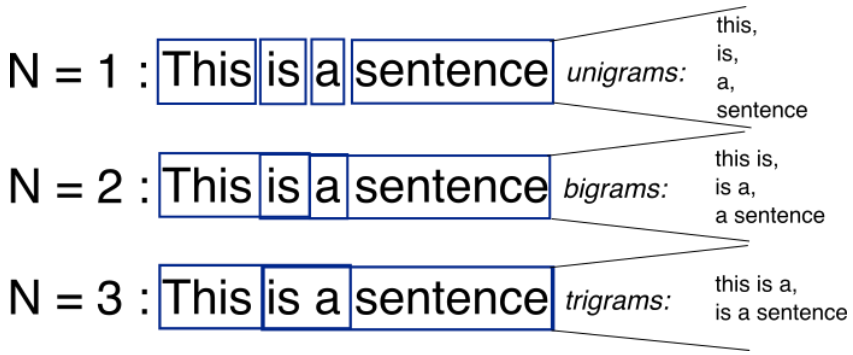


그림 2. n-gram 예시.

Word2Vec과 fastText는 단어의 중의성에 취약하다. 한 단어는 문맥에 따라 다른 의미로 사용될 수 있는데, 두 모델은 이를 고려하지 않고 각 단어를 고정된 벡터로 표현한다. 이는 동일한 단어가 많은 의미를 내포할 수 있는 구어체 의미 분석의 정확도 하락을 야기할 수 있다. 이러한 문제를 해결하기 위해, BERT[5], ELECTRA[6] 등과 같이 대상 언어를 사전 학습하고 이를 목표에 맞게 파인튜닝(fine-tuning)하여 모델의 성능을 높이는 연구가 진행 중이다.

BERT는 단어의 중의성 판단이 가능하고 사전 학습된 언어 모델의 재사용이 가능하다는 장점이 있다. BERT는 단어 임베딩을 위해 자주 등장하면서 가장 긴 길이의 단어 조각(word piece)을 토큰(token)이라는 처리 단위로 사용한다. 임베딩 과정에서는 등장 빈도가 높은 단어 조각 자체가 토큰이 되고, 등장 빈도가 낮은 단어들은 새로운 토큰으로 분할된다. 이러한 단어 조각 임베딩 방식은 모든 언어에 적용할 수 있으며, 모델 성능 향상 효과도 얻을 수 있다. 임베딩된 토큰은 그림 3과 같이 빈칸 채우기 문제와 유사한 MLM(Masked Language Model)을 통해 사전 학습된다. 그러나, MLM을 사용하면 각 입력 문장 당 토큰의 15%만을 학습하므로 전체 학습 비용이 매우 커진다는 단점이 있다.

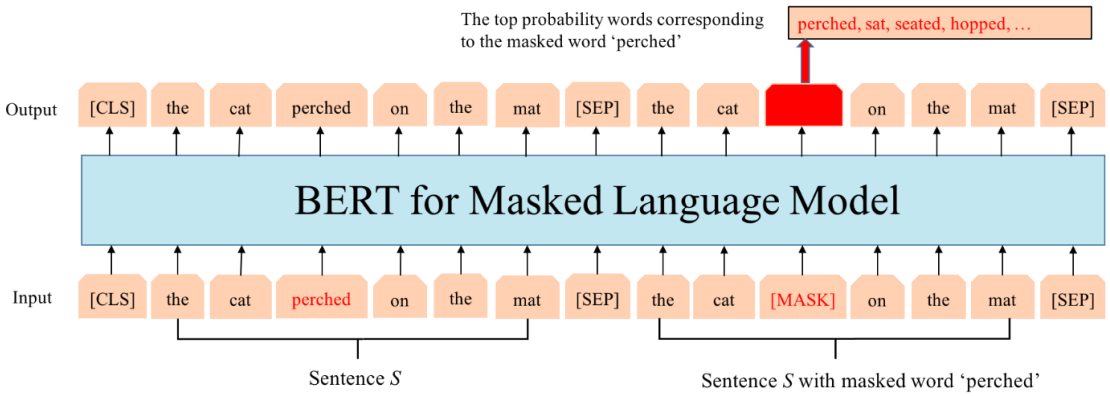


그림 3. BERT의 MLM 기반 사전 학습 과정.

BERT의 학습 시간 단축을 위해, ELECTRA에서는 MLM의 구조를 경량화한 RTD(Replaced Token Detection) 사전 학습 모델을 제시한다. 그림 4는 RDT의 구조를 나타낸다[6]. RDT는 GAN(Generative Adversarial Network)와 유사한 방식으로, 입력 토큰 일부와 유사한 가짜 토큰을 생성하고, 각 토큰이 실제 입력 문장에 포함된 진짜 토큰인지, 생성된 가짜 토큰인지를 구분하는 이진 분류 모델이다. RDT를 이용하면 모든 토큰에 대한 빠른 학습이 가능하므로 기존 BERT에 비해 하드웨어 사용량을 크게 감소시킬 수 있다.

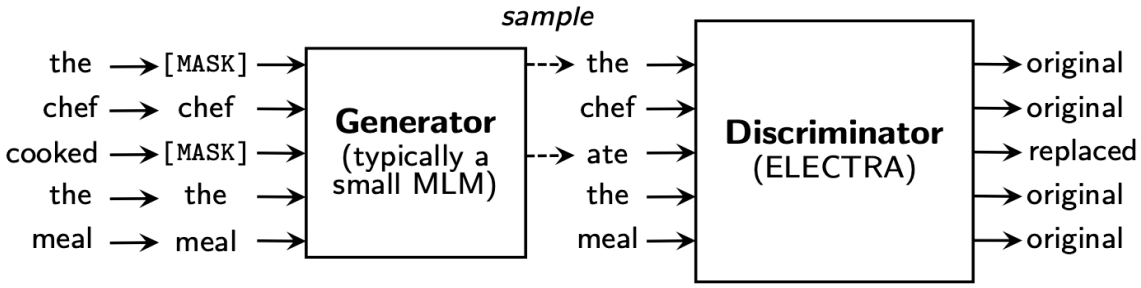


그림 4. Replaced Token Detection 구조.

2.2 욕설 탐지

욕설 탐지는 문장 내 각 단어들의 특징을 파악한 후, 욕설의 유무를 판단하는 과정을 거친다. 그러나, 한국어는 다른 언어에 비해 높은 중의성, 복합 표현 등으로 인해 욕설의 특징을 정확히 추출하는 것이 쉽지 않다. 이를 해결하기 위해, [7]에서는 한글 단어를 자소 분리하여 우회적 욕설의 진화 과정과 그 특징을 추출하는 방안을 제시하였다. 또한, [8]에서는 n-gram을 이용하여 자질을 생성한 후, 카이 제곱 통계량을 기반으로 일정치 이상의 자질만 사용함으로써 욕설 탐지의 문제점

을 극복하고자 하였다. [9]에서는 한글 자소 분리와 형태소 분석을 통해 학습 데이터를 전처리한 뒤, 컨볼루션(convolution) 계층을 이용하여 문장 내 욕설을 탐지하였다. [10]에서는 불용어를 제거한 단어를 추출하고 이를 형태소 분석하여 양방향 장단기 메모리 신경망(bidirectional LSTM Networks)을 이용하여 욕설을 탐지하였다.

3. 임베딩에 따른 한국어 욕설 탐지 모델 설계

본 절에서는 구어체의 문제점을 분석하고, 욕설 탐지의 정확도 향상을 위한 세 가지 탐지 모델을 설계한다. 욕설은 인터넷 댓글, 게시판, 채팅 등에서 쉽게 볼 수 있는 대표적인 구어체로, 많은 형태의 변형이 존재한다. 최근 온라인 서비스들에서는 이러한 욕설을 차단하여 사이버 언어 폭력을 방지하고자 한다. 그러나, 이러한 노력에도 불구하고 욕설을 입력하기 위한 다양한 우회 형태들이 등장하고 있어 실제 차단 효과는 미미하다. 본 논문에서는 이러한 욕설 탐지 시스템의 정확도 향상을 위해 구어체가 가진 문제점을 분석하고, 실험을 통해 한국어 구어체에 적합한 임베딩 방식을 제시하고자 한다.

3.1 구어체 텍스트의 문제점

구어체 텍스트는 비표준어로 이루어진 텍스트이며 오탈자 및 띄어쓰기 오류가 빈번하다. 이는 형태소 분석시 의미 파악을 방해하는 원인이 되며, 딥러닝 모델 탐지 정확도를 하락시킨다. 이러한 비표준 형태소 때문에 불용어 제거를 진행하지만, 이는 원본 텍스트의 의미를 유실시킨다. 예를 들어, 데이터 정제과정에서 'ㄴ'을 불용어라 판단하고 제거한다면,'년아'를 우회적인 방법으로 사용하기 위한 'ㄴ아'는 'ㄴ'이 제거되면서 본래의 뜻을 상실하게 된다.

이를 해결하기 위해 불용어 제거가 아닌 자소 분리를 적용할 수 있지만, 이 또한 다른 문제점을 야기한다. 예를 들어, 특정 단어 사용 금지를 우회하기 위해 입력하려는 단어의 음절 사이에 여러 개의 자소를 추가한다면 텍스트의 특징을 추출할 수 없게 되어 탐지 정확도가 하락한다. 결과적으로, 이러한 구어체 텍스트는 데이터 정제 과정에서 데이터 제거 및 변형을 최소화해야 하며, 임베딩 시 비표준 형태소의 값이 본래의 의미를 가진 표준 형태소의 값과 유사해야 한다.

3.2 구어체 데이터 구축

인터넷 상에서 욕설이 포함된 한국어 구어체 텍스트는 한국어 사용자가 주로 활동하는 인터넷 방송의 채팅이나 각종 커뮤니티의 게시글과 댓글에서 흔히 볼 수 있다. 본 논문에서 사용한 데이터는 총 93,461개의 텍스트이며, 그 예는 그림 5와 같다. 데이터는 인터넷 방송 플랫폼인 Twitch 채팅, 유튜브 자막, 디시인사이드의 인터넷 방송 갤러리와 야구 갤러리의 글 제목을 크롤링하여 구축했다. 그림 5의 예시와 같이, 수집 데이터의 대부분은 오탈자 및 띄어쓰기 오류가 있는 구어체 텍스트이다.

Text	True	False
너겟ㄴ아 왜 정지노	1	0
아직 저거 0.1%의 힘밖에 안쓴건데	0	1
내가 라이브를 보고있어	0	1
시ㅂ 렐빡이들 포스코 간부엿엇네	1	0
옥희말듣고 스갤 폐쇄시키려는 새끼들	1	0

그림 5. 수집 데이터 예시.

그림 6은 수집한 텍스트를 구성하고 있는 25,347개의 형태소 출현 횟수 중 상위 30개를 나타낸 것이다. 그림 6에 보이는 'ㄴ', '1', '?', ',', 'ㅁ' 등은 문장의 의미 파악을 방해하는 구어체 텍스트 형태소 예시이다. 그림 6을 보면 구어체가 단어의 의미를 파악하기 힘든 형태소를 갖는다는 것을 알 수 있다.

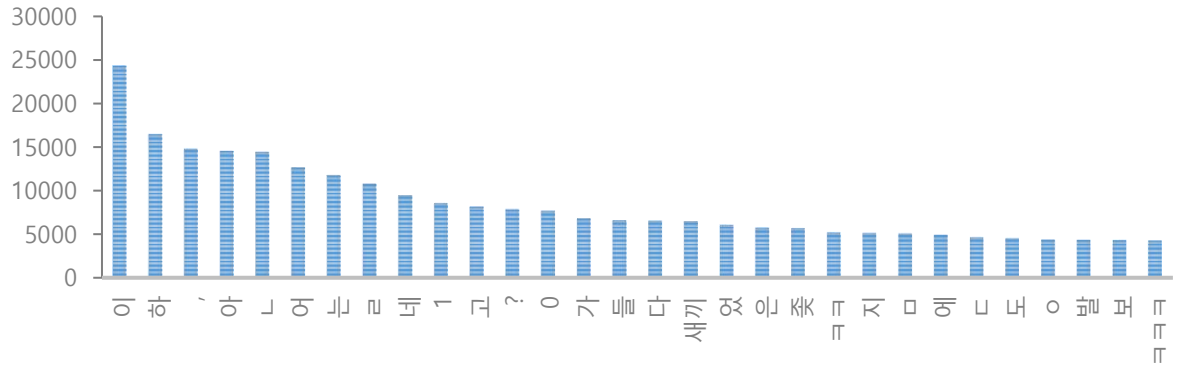


그림 6. 형태소 빈도 Top 30.

그림 7은 레이블링 초기에 사용한 욕설 사전을 일부 발췌한 것이다. 총 908개의 욕설로 이루어진 초기 욕설 사전은 그림 7과 같이 원래 욕설과 유사한 형태를 갖는다. 하지만 이는 다른 언어에 비해 중의적 표현과 복합 표현이 많은 한국어의 특징 때문에 신생 욕설이나 새로운 우회 욕설에 취약하다. 이를 해결하기 위해, fastText로 수집 데이터를 학습시켜 사전 내의 욕설과 코사인 유사도가 높은 3개의 단어를 욕설 사전에 추가하고, 이를 기반으로 레이블링을 진행한다.

개새끼	십새끼	상놈
개새꺄	십색끼	상노무
개새리	십색히	상놈
개객끼	십쉐리	상것

그림 7. 초기 사용 욕설 사전 예시.

욕설 사전 기반 레이블링은 구어체의 특성상 오류가 잦고, 사전에 없는 우회적인 욕설 및 신생 욕설에 대한 레이블링이 불가능하다. 따라서, 정확도를 높이기 위해 추가적인 정제 과정이 필요하다. 본 논문에서 진행한 정제는 다음 세 가지 기준을 가진다.

기준 1. 사용 목적을 알 수 없거나 뜻을 알 수 없는 단어인 경우, 사전 기반 레이블링 결과를 따른다.

기준 2. 단어 단독은 욕설이 아니지만 다른 단어와 조합하여 사용할 시 욕설의 정의에 부합하면 “욕설 문장“으로 취급한다.

기준 3. 단어 단독은 욕설이지만 의미상 욕설의 정의에 부합하지 않으면 “욕설이 아닌 문장”으로 취급한다.

기준 1의 예로는 ‘야붕이’이라는 단어가 있다. 이 단어는 대부분 욕설과 함께 쓰인 단어이지만 사용 목적과 그 뜻을 제대로 파악할 수 없는 단어이다. 이러한 경우 사전 기반 레이블링 결과를 따른다. 기준 2의 예로는 ‘고딩’이라는 단어와 ‘ㅗㅜㅑ’라는 단어이다. ‘고딩’은 고등학생을 뜻하는 단어이며 ‘ㅗㅜㅑ’는 남사스러운 것을 보았을 때 하는 일종의 감탄사이다. 이 두 개의 단어를 단독으로 보았을 때는 욕설이 아니나, 같이 사용하면 욕설의 정의에 부합한다고 판단하여 욕설 문장으로 레이블링한다. 기준 3의 예로는 “미친ㅋㅋㅋㅋㅋㅋ”과 같은 문장이다. 이 문장에서 ‘미친’이라

는 단어는 단독으로 보았을 때 욕설이지만, 예시의 문장은 그 의도가 욕설의 정의에 부합하지 않기 때문에 욕설이 아닌 문장이라고 레이블링한다.

3.3 욕설 탐지 모델 설계

제 3.1절에서 언급한 구어체의 문제점을 해결하기 위해, 본 논문에서는 임베딩 단계를 세 가지로 구분하여 탐지 모델을 설계한다. 첫 번째는 사용 문자 단위에 따른 설계이다. 이는 임베딩에 사용한 문자 단위가 구어체 탐지 결과에 얼만큼 영향을 끼치는지를 파악하기 위한 모델이다. 이 모델에서는 사용 문자 단위가 다른 Word2Vec과 fastText를 임베딩 모델로 선정하여 비교한다. 두 번째는 단어 중의성 판단 여부에 따른 탐지 결과를 비교하기 위한 모델이다. 이 경우, 중의성 판단 여부가 다른 따른 임베딩 모델이 필요하므로 fastText와 BERT를 한국어에 맞게 개선한 SKT-KoBERT[11]를 사용한다. 세 번째는 사전 학습 방법 따른 탐지 정확도를 비교하기 위한 모델이다. 이를 위해, BERT 인코더를 그대로 사용하지만 사전 학습 방법이 다른 SKT-KoBERT와 한국어에 맞게 개선된 ELECTRA인 KoELECTRA[12]를 사용한다.

그림 8은 사용 문자 단위에 따른 욕설 탐지 모델의 구조도이다. 그림을 보면, 두 개의 첫 번째 모델은 임베딩 모델을 통해 생성된 단어 벡터를 입력으로 하여 욕설을 탐지하는 FCN(F-C-N)구조로 되어 있다. 이때, Word2Vec이 새로운 데이터에 대한 적응력이 없기 때문에, Word2Vec과 fastText 모델의 사전 학습에는 전체 데이터가 사용된다. 학습 옵션은 최소 등장 횟수(min_count) 1, 윈도우 사이즈(window) 2, 차원(size) 50, 학습 반복(iter) 100으로 구성되며, Skip-Gram(sg) 기법을 통해 학습한다. 이러한 임베딩을 거친 욕설 탐지 모델은 6개의 완전 연결 계층과 4개의 1차원 컨볼루션 계층(Conv1d)으로 구성되며, 옵티마이저로는 Adam을 사용한다.

첫 번째 모델의 주요 학습 옵션과 동작 순서는 다음과 같다. 먼저, 각 문장당 8개의 형태소를 임베딩하여 400차원의 입력 데이터를 생성하고, 이를 400개의 유닛을 갖는 완전 연결 계층으로 입력한다. 다음으로, 완전 연결 계층의 결과는 배치 정규화를 거쳐 Relu 기반 Conv1d(필터 400개, 커널 5개)에 입력된다. 이후 Conv1d의 출력은 Sigmoid 기반 Conv1d(필터 400개, 커널 5개)의 입력이 된다. 이때, 출력 매트릭스 중 대표값을 찾기 위한 MaxPooling, 과적합을 막기 위한

Dropout, 완전 연결 계층 입력을 위한 Flatten 과정을 추가하였다. 각 과정을 거친 데이터는 각각 500, 250, 125, 50개의 유닛으로 구성된 Relu 기반 완전 연결 계층에서 순차적 학습되며, 최종적으로 2개의 유닛으로 구성된 Sigmoid 기반 완전 연결 계층으로 문장 내의 욕설 유무를 판별한다

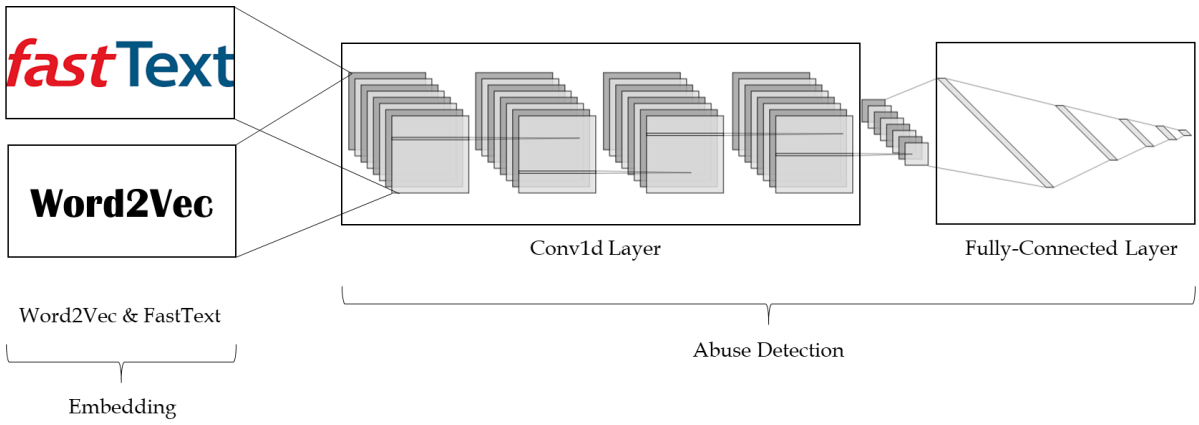


그림 8. Word2Vec 및 fastText 기반 욕설 탐지 모델.

중의성 판단 여부에 따른 두 번째 모델은 사전 학습된 fastText 기반 탐지 모델과, 파인튜닝된 SKT-KoBERT 기반 탐지 모델로 구분하여 설계한다. fastText 기반 탐지 모델의 사전 학습 옵션은 차원이 96으로 구성되는 것 외에는 첫 번째 모델과 동일하다. 그러나, 첫 번째 모델과는 다르게, 중의성 판단 여부 모델에서는 문장의 8개 형태소를 768차원의 입력 데이터로 생성해 사용한다. 이후, 이를 활성화 함수로 Sigmoid 사용하고 입력 유닛 768개, 출력 유닛 2개의 하나의 완전 연결 계층의 입력으로 사용한다. fastText와 비교할 SKT-KoBERT는 Google의 BERT 기반 다국어 모델의 한국어 성능 제약을 해결하기 위해 SKT에서 한국어 위키 500만건과 한국어 뉴스 2,000만건을 사용하여 학습시킨 임베딩 모델이다. 그림 9에서 보듯이, SKT-KoBERT는 기존 BERT를 활용하므로, fastText와 마찬가지로 입력 유닛 768개, 출력 유닛 2개의 완전 연결 계층으로 구성되며, 활성화 함수는 Sigmoid를 사용한다. fastText와 SKT-KoBERT 모델 모두 옵티마이저는 Adam을 사용한다.

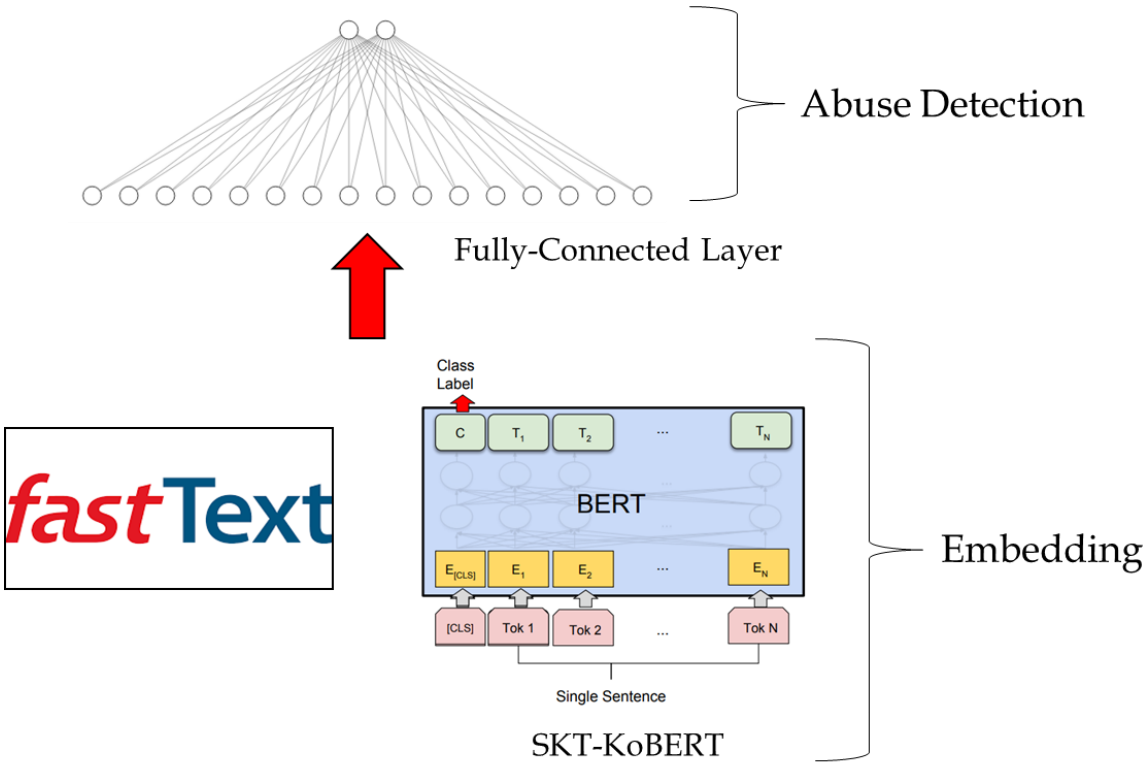


그림 9. fastText 및 SKT-KoBERT 기반 욕설 탐지 모델.

사전 학습 방법에 따른 세 번째 모델은 파인튜닝된 SKT-KoBERT 기반 탐지 모델과 파인튜닝된 KoELECTRA 기반 탐지 모델로 구분하여 설계한다. KoELECTRA는 한국어에 대해 성능의 한계가 있는 기존 ELECTRA를 14GB의 한국어 텍스트(9,600만 문장, 2.6B 토큰)로 학습시킨 임베딩 모델이다. 여기에서, SKT-KoBERT 기반 탐지 모델의 구조는 두 번째 모델과 동일하다. KoELECTRA 기반 탐지 모델의 구조는 그림 10과 같다. ELECTRA 역시 BERT 인코더와 같은 구조이므로, 입력 유닛 768개, 출력 유닛 2개의 완전 연결 계층으로 구성되며, 활성화 함수는 Sigmoid를 사용한다. SKT-KoBERT와 KoELECTRA 모델 모두 옵티마이저는 Adam을 사용한다.

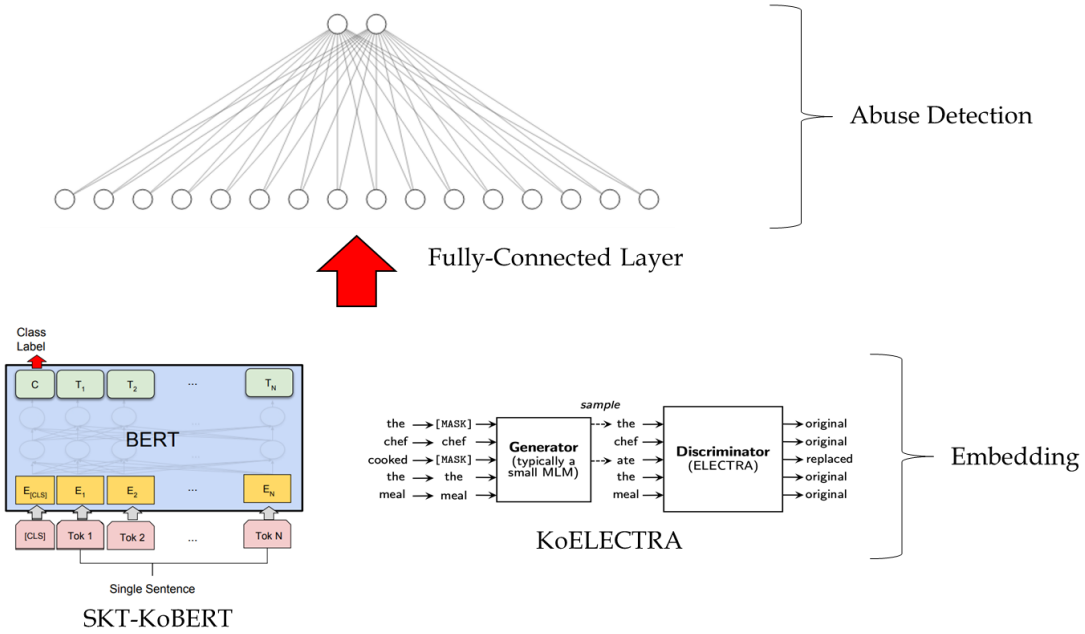


그림 10. SKT-KoBERT 및 KoELECTRA 기반 욕설 탐지 모델.

4. 실험 및 평가

실험에서는 93,461개의 텍스트 중 80%를 트레이닝 데이터로, 20%를 테스트 데이터로 설정한다. 첫 번째 모델은 전체 텍스트가 임베딩 학습에 관여한 결과를 비교 및 평가하고, 두 번째, 세 번째 실험에서는 80% 텍스트가 임베딩 학습에 관여한 결과를 비교 및 평가한다.

표 1은 Word2Vec과 fastText로 임베딩한 데이터를 학습한 첫 번째 탐지 모델의 테스트 결과이다. 표 1을 보면, 임베딩 모델과 상관없이 4개 항목(Recall, Precision, F1, Accuracy)에서 약 90% 이상의 성능을 보임을 알 수 있다. 이는 단어의 구분 단위와는 관계없이 모든 텍스트가 임베딩 모델 학습에 관여하는 경우 높은 정확도로 문장 내의 욕설을 탐지할 수 있음을 의미한다.

표 1. Word2Vec과 fastText의 성능 비교.

	Word2Vec	**fastText**
Recall	0.8935	0.8984
Precision	0.9359	0.9321
F1	0.9142	0.9150
Accuracy	0.9281	0.9263

표 2는 두 번째 탐지 모델인 fastText 및 SKT-KoBERT 기반 욕설 탐지 테스트 결과이다. 표를 보면, fastText 기반 탐지 모델이 Precision을 제외한 3개 항목에서 SKT-KoBERT 기반 탐지 모델에 비해 낮은 성능을 보임을 알 수 있다. 이는, fastText와 같이 단어 중의성 파악이 어려운 임베딩 모델의 경우, 사전 학습에서 접하지 못한 새로운 텍스트의 분류 성능이 중의성 파악이 가능한 BERT 기반 임베딩 모델에 비해 크게 떨어진다고 볼 수 있다.

표 2. fastText와 SKT-KoBERT의 성능 비교.

	fastText	**SKT-KoBERT**
Recall	0.8541	0.9366
Precision	0.8363	0.9670
F1	0.8451	0.9516
Accuracy	0.8639	0.9585

표 3은 세 번째 탐지 모델인 SKT-KoBERT 및 KoELECTRA 기반 욕설 탐지 결과이다. 표를 보면, SKT-KoBERT 기반 탐지 모델의 점수가 모든 항목에서 KoELECTRA 기반 탐지 모델에 비해 높은 성능을 보임을 확인할 수 있다. 이는 예외가 많은 구어체 탐지에서는 사전 학습 방법으로 MLM을 사용한 임베딩 모델이 RTD를 사용한 것보다 좀 더 높은 성능을 보임을 의미한다.

표 3. SKT-KoBERT와 KoELECTRA의 성능 비교.

	SKT-KoBERT	KoELECTRA
Recall	0.9366	0.8588
Precision	0.9670	0.9356
F1	0.9516	0.8955
Accuracy	0.9585	0.9129

5. 결론

본 논문에서는 다양한 임베딩 모델 기반 한국어 욕설 탐지 모델을 설계하고, 각 모델의 성능을 비교, 평가하였다. 구어체 텍스트에 대한 임베딩 모델별 탐지 정확도를 비교하기 위해, 사용 문자 단위, 중의성 판단 여부, 사전 학습 방법에 따라서 세 가지 탐지 모델을 설계하고, 실험을 통해 각 모델 탐지 성능을 비교 및 분석하였다. 실험 결과, 첫 번째 모델은 사용 문자 단위와 상관없이 Word2Vec, fastText 모두 약 90% 이상의 탐지 정확도를 보였다. 그러나, 중의성 판단 여부와 사전 학습 방법에 따른 두번째, 세 번째 모델에서는 SKT-KoBERT가 fastTexT와 KoELECTRA에 비해 높은 탐지 성능을 보임을 확인하였다. 이는 단어 중의성 파악이 가능하고 사전 학습 방법으로 MLM을 사용한 SKT-KoBERT 모델이 구어체 텍스트를 대상으로 하는 탐지 환경에 좀 더 효과적임을 의미한다. 본 논문에서 제시한 결과를 통해 향후 욕설 탐지뿐만 아니라 구어체 기반 자연어 분석, 탐지 시에 좀 더 효율적인 임베딩 모델을 적용할 수 있을 것으로 사료된다.

참고 문헌

[1] https://www.sedaily.com/NewsView/1VQS6U8894.

[2] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient Estimation of Word Representations in Vector Space," In *Proc. of the Int'l Conf. on Learning Representations Workshop Track*, pp. 1301-3781, Sep. 2013.

[3] T. Mikolov, I. Sutskever, K. Chen, G. Corrado, and J. Dean, "Distributed Representations of Words and Phrases and their Compositionality," In *Proc. of the Int'l Conf. on Neural Information Processing Systems*, pp. 3111-3119, Dec. 2013.

[4] P. Bojanowski, E. Grave, A. Joulin, and T. Mikolov, "Enriching Word Vectors with Subword Information," *Transactions of the Association for Computational Linguistics*, Vol. 5, pp. 135-146, July 2016.

[5] J. Devlin, M. Chang, K. Lee, and K. Toutanova, “BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding,” In *Proc. of the Conf. of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, Vol. 1, pp. 4171-4186 , June 2019.

[6] K. Clark, Minh-Thang Luong, Quoc V. Le, Christopher D. Manning, “ELECTRA: Pre-training Text Encoders as Discriminators Rather Than Generators”, https://arxiv.org/abs/2003.10555, Mar. 2020.

[7] 윤태진, 조환규, “한글 자소정렬을 이용한 온라인 욕설 필터링 시스템,” *한국정보과학회 학술발표논문집*, Vol. 36, No. 2C, pp. 194-198, Nov. 2019.

[8] 박교현, 이지형, “SVM을 이용한 온라인게임 비속어 필터링 시스템,” *한국정보과학회 학술발표논문집*, Vol. 33, No. 2C, pp. 260-263, Oct. 2016.

[9] 박성희, 김휘강, 우지영, “딥러닝을 사용한 온라인 게임에서의 욕설탐지”, *한국컴퓨터정보학회 학술발표논문집*, Vol. 27, No. 2, pp. 13-14, July 2019.

[10] 나인섭, 이신우, 이재학, 고진광, “양방향 장단기 메모리 신경만을 이용한 욕설 검출”, *한국빅데이터학회지*, 제4권, 제2호, pp. 35-45, Dec. 2019.

[11] https://github.com/SKTBrain/KoBERT.

[12] https://github.com/monologg/KoELECTRA.