컴퓨터교육학회 논문지 2024년 제27권 제3호
https://doi.org/10.32431/kace.2024.27.3.010

상품 카테고리 자동분류를 위한 BERT-분류기 아키텍처 연구*

Research on a BERT-Classifier Architecture for Automatic Product Category Classification

김도훈† · 임희석††

Do-Hoon Kim† · Heui-Seok Lim††

요 약

본 연구는 생활 속 존재하는 다양한 상품들의 명칭을 BERT를 통해 임베딩 벡터화한 다음 이를 기반으로 상품 카테고리 예측을 수행하는 아키텍처에 대한 연구이다. 아키텍처의 성능은 상품 명칭으로부터 임베딩 추출을 수행하는 BERT 모델과, 추출된 임베딩으로 카테고리 예측을 수행하는 분류기에 의해 결정된다. 따라서 본 연구는 우선 상품 명칭 분류에 적합한 BERT 모델을 선정하고, 선정된 BERT 모델에 다양한 분류기를 적용하여 가장 높은 성능을 달성하는 BERT-분류기 조합을 찾고자 하였다. 최초 적합한 BERT 모델 선정에는 단순한 CNN 분류기를 사용하였으며 이를 baseline으로 다른 분류기와 성능을 비교하였다. 아키텍처의 성능은 카테고리 정답에 대한 precision, recall, f1 score, accuracy로 정량화하여 평가하였다. 실험 결과 BERT 측면에서는, Sentence BERT 모델이 비교 대상인 일반 BERT 모델보다 적합함을 확인하였다. 그리고 분류기 측면에서는, Sentence BERT와 CNN으로 구성된 baseline 대비하여 Residual Block이 추가 적용된 분류기가 더 높은 성능을 보였다. 본 연구에 사용된 Sentence BERT 모델의 경우 한국어 데이터가 학습되지 않은 단순 모델로, 향후 추가적 연구를 통해 다양한 한국어 데이터를 학습시켜 Domain Adaptation을 수행할 경우 추가적 성능 향상이 기대된다.

주제어: 문장 분류, 문장 유사도, Sentence BERT, CNN, ResNet, Transformer, Classification

ABSTRACT

This research focuses on an architecture that vectorizes the names of various products found in daily life using BERT, followed by predicting product categories based on these embeddings. The architecture's performance is determined by the BERT model, which extracts embeddings from product names, and the classifier that predicts categories from these embeddings. Consequently, this research initially aimed to identify a BERT model suitable for classifying product names and then find the most efficient combination of BERT model and classifier by applying various classifiers to the chosen BERT model. A simple CNN classifier was employed for the initial selection of a suitable BERT model, serving as a baseline for performance comparison with other classifiers. The architecture's effectiveness was quantified using precision, recall, f1 score, and accuracy for category predictions. Experimental results showed that the Sentence BERT model was more suitable for this task than a conventional BERT model. Additionally, classifiers enhanced with Residual Blocks demonstrated superior performance compared to the baseline combination of Sentence BERT and CNN. The Sentence BERT model used in this study, not trained on Korean data, suggests that further improvements could be achieved through Domain Adaptation by training with diverse Korean datasets.

Keywords: Sentence classification, Sentence similarity, Sentence BERT, CNN, ResNet, Transformer, Classification

†정 회 원: 고려대학교 컴퓨터정보통신대학원 석사과정
††정 회 원: 고려대학교 컴퓨터학과 교수 (교신저자)

논문투고: 2024년 02월 29일, 심사완료: 2024년 04월 09일, 게재확정: 2024년 04월 10일

* 본 논문은 2024년 한국컴퓨터교육학회 동계학술대회 (KCEC 2024 Winter) 에서 우수 논문으로 선정된 "상품 명칭 분류 작업에서의 BERT 성능 비교 : Sentence-BERT와 RoBERTa 기반의 CNN 분류기 성능 분석" 논문을 확장한 것임.

1. 서 론

트랜스포머 기반의 BERT가 등장한 이후로 다양한 알고리즘이 추가된 파생 모델들이 탄생하였다. 이들 모델은 맥락이 고려된 임베딩을 추출하는 작업에 특화되어 있다. 그리고 이를 기반으로 다양한 자연어 유관 작업을 효율적으로 처리하는 방법이 활발하게 연구되고 있다[1]. 여러 연구 주제 중에서도 '문장 분류'는 교육, 마케팅, 재난 안전, 연구 동향 파악, 지식 데이터베이스 구축, 추천 등 다양한 분야에서 확장적 활용이 가능한 연구 주제라고 볼 수 있다.

본 연구는 여러 한국어 문장 분류 작업 중에서도 우리가 일상에서 마주하는 상품의 명칭을 식별 및 분류하는 작업에 대해 다루고 있다. 보다 구체적으로는 상품들의 명칭을 BERT를 통해 임베딩 벡터화한 다음 이를 기반으로 상품 카테고리 예측을 수행하는 아키텍처에 대한 연구이다.

일상생활 속 상품의 명칭을 효율적으로 식별 분류하는 연구는 장기적으로는 사물에 대한 인식이 필요한 모든 영역, 사람의 생활을 보조하는 비서인공지능 영역 등 다양한 일상영역에 기여할 수 있다. 또한 산업적으로는 e-commerce나 유통산업에서 관리하는 8만~10만개에 달하는 운영상품의 기준정보 관리 측면에 기여할 수도 있다는 점에서 의미가 있다[2].

본 연구가 다루는 데이터는 우리가 실제 세계에서 마주하는 상품의 명칭을 대상으로 한다. 상품 명칭의 식별 및 분류 작업은 일견 단순해 보일 수 있으나, 실질적으로 많은 실무상 어려움이 따른다. 상품의 명칭은 특정한 패턴이 극단적으로 반복되거나, 혹은 전혀 일정한 패턴이 없는 경우도 존재하는 불균형 데이터이기 때문이다. 또한 충분히 문맥을 담을 만큼 긴 텍스트를 형성하기보다는, 필요한 정보만 담겨있는 15~20자 내외의 짧은 문장으로 구성되어있다.

예를 들어, 가전제품의 모델 명칭, 패션상품의 명칭은 모델옵션, 색상, 사이즈 등 파생형에 따라 종류가 매우 다양하여 동일한 패턴의 반복이 무수히 발생한다. 반면 농산물 등 식료품의 경우 기본적으로 데이터의 표본 수 자체가 적으며, 품종 또한 다양하지 않아 일정한 패턴이 없는 경우가 많다. 이와 더불어 연간 수많은 상품들이 생성과 소멸을 반복하기 때문에 모델이 전혀 보지 못한 데이터들이 지속적으로 등장할 가능성이 높다. 본 연구에서는 이러한 상품 명칭의 자연스러운 특성이 반영된 데이터 세트를 이용해 작업의 일반적인 수행 환경을 재현하였다.

아키텍처의 성능은 상품 명칭으로부터 임베딩 추출을 수행하는 BERT 모델과, 추출된 임베딩으로 카테고리 예측을 수행하는 분류기에 의해 결정된다. 따라서 본 연구는 우선 상품 명칭 분류에 적합한 BERT 모델을 선정하고, 선정된 BERT 모델에 다양한 분류기를 적용하여 가장 높은 성능을 달성하는 BERT-분류기 조합을 찾고자 하였다.

최초 적합한 BERT 모델 선정에는 단순한 CNN 분류기를 사용하였으며 이를 baseline으로 다른 분류기 적용 사례들과 precision, recall, f1 score, accuracy 등의 정량지표 성능을 비교하였다.

2. BERT기반 분류기의 기존 연구

2.1 Fine-tuning 접근

BERT는 기본적으로 Token, Segment, Position의 3종 입력 벡터가 트랜스포머 인코더 레이어를 통과하여 최종적으로 출력 벡터로 변환되는 구조를 취하고 있다. 이 때, 입력된 문장의 클래스는 출력 벡터의 '[CLS]' 토큰으로 식별 가능하다. 이 토큰이 입력 벡터의 종합적인 의미 전반을 담고 있기 때문이다.

기존 연구에서는 문장의 분류 작업에 대해 '[CLS]' 토큰 벡터를 소프트맥스 함수가 적용된 레이어를 통과시켜 각 클래스별 확률을 도출하는 방식으로 문장의 분류 작업을 수행한다. 이러한 방식은 '[CLS]' 토큰 하나에 의존하여 기초적인 수준의 분류 작업을 수행하며, 분류 성능의 개선 및 타 도메인에 대한 적응을 위해서는 기존 모델에 대한 추가학습 과정이 필연적이라는 특징이 있다. 이러한 특성으로 인해 BERT 모델 자체가 특정 작업에 특화되어 해당 도메인 내에서의 높은 성능을 달성할 수 있다는 장점이 있다[3].

2.2 Feature-based 접근

이 방식에서는 BERT 모델 자체를 추가학습 시키지 않으며, 일반화된 성능을 가진 BERT의 출력 벡터를 다른 여러 분류기 모델의 입력 벡터로 활용한다. 또한 '[CLS]' 토큰에만 의존하지 않고 트랜스포머 상위 레이어에서 추출된 벡터들을 연결(concatenation)하여 BiLSTM등 모델을 통과시켜 분류작업을 수행하는 등 다양한 벡터를 활용한다는 특징이 있다[3].

Feature-based 접근방식의 장점으로는 임베딩 추출을 수행하는 BERT 모듈과, 분류작업에 특화된 분류기 모듈을 다양하게 조합해볼 수 있다는 점이 있다. 가령 문맥의 특성을 기존 BERT모델보다 더 정확히 담을 수 있고, 문장 간의 유사도 비교에 특화된 Sentence BERT로 BERT 모듈을 구성하고, 그 위에 CNN, BiLSTM등의 다양한 분류기 모델을 활용하는 등 여러 조합에 대한 성능 분석이 가능하다[4, 5]. 또 다른 장점으로는, BERT 모델을 별도의 추가학습 없이 일반화된 모델 기반으로 분류 작업 수행이 가능하여 수반되는 비용이 적다는 점이 있다.

2.3 기존 연구의 한계

먼저 Fine-tuning 방식의 경우 전체 모델에 대한 추가학습이 수반되어 컴퓨팅 자원 소모 대비 효율이 좋지 못하다는 한계가 있다. 이러한 방식은 계약문서 작업 등, 고도의 정밀도를 요구하는 환경에서 유효할 수 있다. 그러나 인공지능 비서, 운전 보조 장치, 재난구조 등 다양한 환경에 대한 적응성이 중요한 환경에서는 이러한 방식이 비효율 적일 수 있다. 또한 본 연구에서 다루는 상품 명칭데이터의 경우 지속적으로 새로운 상품들이 등장하고, 다양한 분류를 가지고 있기 때문에 전체 모델에 대한 빈번한 추가학습이 비용 및 효익 관점에서 적합하지 않다고 볼 수 있다.

Feature-based 방식의 경우 일반화된 BERT모델과 특정 작업에 특화된 분류기 모델을 별도로 모듈화 한다. 따라서 분류기 성능 향상에 필요한 컴퓨팅 자원 소모를 효율화 할 수 있다는 장점이 있다. 이로 인해 다양한 도메인에 대하여 다방면의 빠른 활용이 가능하다. 또한 단일 '[CLS]' 토큰 벡터가 아닌 다양한 출력 벡터를 Feature로 활용하기 때문에 보다 다양한 형태의 엔지니어링이 가능하며 각 모듈을 따로 연구할 수 있다는 장점이 있다[4].

다만 현재까지의 연구들은 해당 연구가 다루고 있는 특정 데이터에 대해 Feature-based 모델이 어느 정도 성능을 보이는지 단일 모델만으로 실험을 수행할 따름이라는 한계가 있다. 본 연구에서는 분류 작업에 응용이 가능한 CNN, Transformer 구조 등 다양한 알고리즘으로 비교 실험을 수행한다는 점에서 기존 연구대비 차별화된다.

3. 실험 데이터 선정 및 전처리

3.1 데이터의 선정 및 출처

실험을 위한 데이터는 인터넷 쇼핑몰 상에서 수집한 식품/일상용품/전자제품/패션잡화 등 다양한 카테고리별 상품명 데이터와 한국지능정보사회진흥원 ai-hub의 상품분류용 데이터를 조합하여 원시 데이터 풀을 확보하였다. ai-hub상의 데이터는 롯데정보통신에서 구축한 공개용 데이터 '상품 이미지 (2020)'를 기반으로 본 연구에 필요한 상품 명칭과 대/중/소/상품명의 데이터만을 취사선택하였다. 이 중 패턴의 반복, 모델 명칭으로만 구성된 상품 명칭 등을 제외하고 최종적으로 53,830개 상품 명칭 데이터를 실험 대상으로 선정하였다. 53,830개의 데이터는 각 상품별로 Level 1, 2, 3 의 카테고리 라벨을 가지고 있으며, 각각 76개, 423개, 1,083개의 class가 존재한다. 본 연구에서는 기준이 되는 카테고리로 Level 1(대분류)을 사용하였고, 여러 가지 세부적인 결과의 해석에 Level 2(중분류)와 Level 3(소분류)을 보조적으로 활용하였다. Table 1.은 본 연구가 최종적으로 실험을 수행할 데이터 세트의 클래스 수와 데이터 예시를 나타낸 표이다. Figure 1.은 76개의 데이터 클래스별로 53,830개의 데이터가 어떻게 분포하고 있는지를 그래프로 나타낸 그림이다.

Table 1. Dataset Summary and Examples

Level	count(class)	Example
Lv1	76	'음료'
Lv2	423	'생수', '탄산음료'
Lv3	1,083	'일반생수', '콜라', '사이다'
SKU	53,830	'삼x수 500ml', '코x콜x 1.25L'

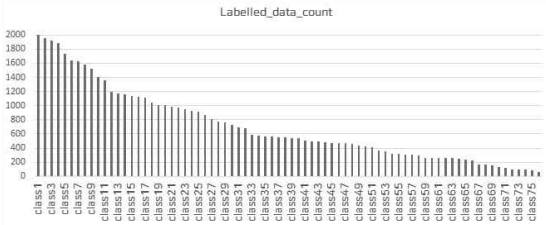

Figure 1. Class-wise data distribution

3.2 전처리

전처리 수행은 정규표현식을 이용하여 특수문자와

숫자만을 제거하였다. 특수문자와 숫자가 있을 경우 단량, 단위의 표기 등이 분류 성능에 직접적인 결과에 영향을 주기 때문이다. 가령, ‘500g’ 와 같은 빈번하게 발생하는 단량 정보의 경우 하나의 패턴으로 인식되어 성능에 영향을 주게 된다. 반면 영문은 제거하지 않았다. ‘USB’, ‘C타입’ 등 상품의 중요 정보를 파악함에 있어 중요한 정보들이 소실되어 임베딩의 품질이 낮아지기 때문이다. ‘Sam****’, ‘Ap***’ 과 같은 브랜드 정보 역시 소실을 방지하기 위해 전처리대상에서 제외하였다. Table 2.는 전처리 유형별 처리 방식에 대한 표이다. Examples는 문제가 된 사례이고, Case는 문제의 유형을 의미한다. Preprocessing은 최종적으로 어떤 전처리를 수행하였는지에 대한 내용이다.

Table 2. Cases and Examples by Preprocessing Type

Examples	Case	Pre-processing
‘[라탄] 직T 바스켓 1호 ※그레이색상’	Special Characters	Removal
‘딸기잼 500g’, ‘티라미수 500g’	Numbers	Removal
‘시아시아 팔렛 멀티탭 4구 USB_3m’	English	Preservation

4. BERT 모델 성능평가

본 장에서는 Sentence BERT[5] 모델 2종과 일반 BERT[3] 모델 3종을 간단한 CNN구조가 적용된 분류기와 조합해 성능을 비교한다. 이를 통해 상품 분류 작업에 더 적합한 성능을 가진 BERT 모델을 식별하고 이를 분류기 성능평가의 baseline으로 상정하고자 한다.

4.1 아키텍처 정의

상품 명칭 분류 작업을 위한 아키텍처는 2장에서 언급한 기존의 Feature-based 연구들을 참고하여 일반적인 구조로 설계하였다. 아키텍처는 (1) 상품 명칭의 전처리 및 임베딩 추출을 수행하는 BERT 임베딩 추출 모듈, (2) 추출된 임베딩을 Input으로 받아 분류기의 학습을 수행하는 모듈, (3) 학습된 분류기를 통해 테스트 데이터에 대한 prediction을 수행하는 모듈 총 3개로 구성된다.

본 장에서는 BERT 모듈에 Sentence BERT[5] 계열 2종과 일반 BERT[3] 계열 3종의 모델과 기본적인 CNN분류기를 조합해 본 연구의 작업에 적합한 BERT 모델을 선택하고 본 연구의 baseline으로 상정한다. 후술할 5장에서는 다양한 분류기를 baseline과 비교분석을 수행한다. 이 때 사용되는 분류기는 ResNet (32), ResNet (Shallow), Transformer의 3종을 사용한다. 아래 Figure 2.의 그림은 본 장에서 수행할 baseline 아키텍처를 도식화한 것이다.

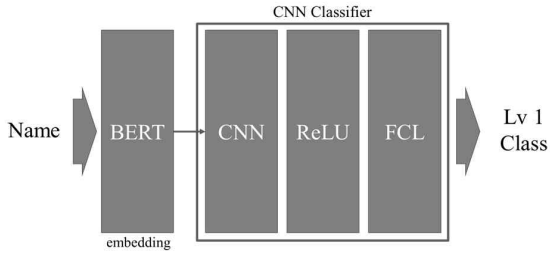

Figure 2. Example of task workflow with classifier (CNN)

4.2 임베딩 추출에 사용될 BERT 모델 정의

임베딩 추출을 위한 BERT 모델은 Sentence BERT[5] 모델과 토큰화 기반의 BERT[3] 일반 모델을 사용하였다. 먼저 Sentence BERT 모델은 이 모델의 파생형인 paraphrase-multilingual-mpnet-base-v2와 distiluse-base -multilingual-cased-v2 (Nils Reimers, Iryna Gurevych, 2019) 2종류를, 일반 BERT 모델로는 KcBERT(이준범, 2020), KlueBERT (upstage, 2021), KcELECTRA (이준범, 2021) 3종류를 사용하였다.

Sentence BERT 모델은 일반 BERT 모델의 문장 임베딩이 문장과 문장을 비교하는 작업에 적합하지 않음을 해결하기 위해 제시되었다. Sentence BERT 모델은 Siamese 네트워크와 Triplet 네트워크를 활용하여 서로 다른 문장들의 비교를 학습한 모델이다. Figure 3.은 바로 이러한 Sentence BERT 네트워크의 구성을 보여준다. 이를 통해 생산된 임베딩은 문장의 전체적인 의미구조를 담고 있으며 논문은 이를 “semantically meaningful” 이라는 표현으로 언급하고 있다[5].

반면 일반 BERT모델의 경우, 우선 문장을 토큰화한 다음, 이 토큰들을 기반으로 문장에 대한 임베딩을 생성하게 된다[3]. 결론적으로, Sentence BERT 모델이 생성하는 임베딩의 경우 문장 전체의 맥락을 담고 있는 반면[6] 일반 BERT 모델의 임베딩은 토큰화 된 개별데이터의 임베딩으로 볼 수 있다[7].

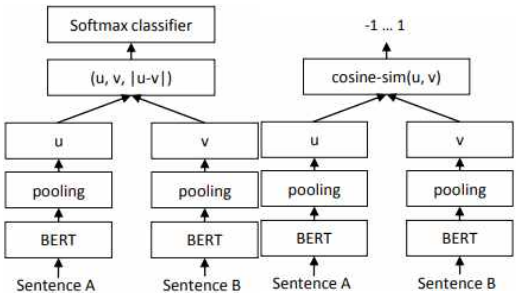

Figure 3. The Basic Structure of Sentence BERT[5]

4.3 CNN[8] 분류기(baseline) 정의 및 학습 진행

baseline으로 사용될 CNN 분류기는 1차원 CNN Layer, ReLU 활성함수, Fully connected layer 총 3개 요소로 구성된다[8]. 최대한 CNN 성능의 개입 없이 순수하게 BERT 임베딩의 결과만으로 Classification을 수행 하고자 CNN에 별도의 구조적 기법을 사용하지 않았다. 모델은 10 epoch으로 학습되었으며, 0.001의 Learning rate, Adam optimizer를 사용하였다. Table 3.은 CNN 분류기(baseline)의 레이어 구성을 설명한 표이다.

Table 3. The CNN architecture (in this experiment)

Layer	Channel	Kernel	Stride	Padding
Conv	128	5	1	2
ReLU	-	-	-	-
MaxPool	-	5	5	-
Linear	num_class	-	-	-

위의 구조에 따라 CNN 분류기(baseline) 학습 결과 아래 Table 4.의 결과와 같이 Sentence BERT 모델의 파생형인 paraphrase-multilingual-mpnet-base-v2이 가장 낮은 Loss를 달성하였다. Figure 4.는 이때의 Loss Curve를 그래프로 나타낸 그림이다. 본 연구의 train, test 데이터의 분할은 8:2의 비율로 진행하였으며, stratify 옵션을 통해 클래스별로 균일하게 추출하였다. 최종적으로 각 분류기 모델들은 43,064개 데이터로 학습하였다.

Table 4. Training results for different BERT models

Model	BERT	Loss
sentence-transformers/paraphrase-multilingual-mpnet-base-v2	S-BERT	0.3692
sentence-transformers/distiluse-base-	S-BERT	0.9839
multilingual-cased-v2		
klue/roberta-large	BERT	1.0126
beomi/KcELECTRA-base-v2022	BERT	0.8885
beomi/kcbert-large	BERT	0.6685

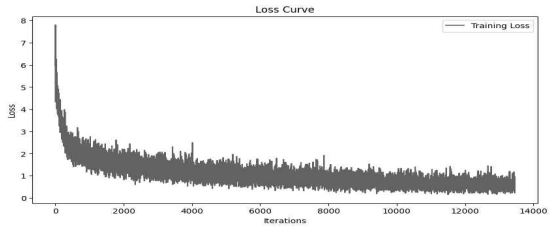

Figure 4. Loss Curve of CNN baseline case (epoch=10)

4.4 성능 측정 결과

4.4에서 학습을 진행한 CNN 분류기에 Sentence BERT와 일반 BERT를 조합하여 분류의 성능을 평가하였다. BERT모델은 'Hugging Face' 플랫폼에 오픈소스로 공개되어있는 Sentence BERT 모델과 한국어 기반의 일반 BERT 모델을 활용하였다. 우선 Sentence BERT 의 경우 paraphrase-multilingual-mpnet-base-v2, distiluse-base-multilingual-cased-v2 (Nils Reimers, Iryna Gurevych, 2019)를 이용하였다. 일반 BERT 모델은 KcBERT(이준범, 2020), KlueBERT(upstage, 2021), KcELECTRA (이준범, 2021)를 이용해 진행하였다. 결론적으로 총 5개의 분류기가 각각 10,766개의 데이터에 대해 prediction을 진행하였다[3, 5, 8, 9, 10].

성능의 평가는 데이터 상 가장 상위 카테고리 분류인 Level 1 카테고리를 prediction 하는 방식으로 수행되었다. 성능의 지표로는 precision, recall, f1 score, accuracy를 사용하였다. precision, recall, f1 score의 경우 macro average를 사용하였는데, 그 이유는 사용한 데이터의 class별 분포가 차이나는 경우가 많아 이 영향을 최소화하기 위함이다. 아래 Table 5.은 prediction의 수행 결과이다.

Table 5. Evaluation based on the Baseline Classifier

No.	BERT embedding	BERT	macro avg.			acc
			pre-cision	recall	f1 score	
(1)	paraphrase-multilingual-mpnet-base-v2	S-BERT	0.73	**0.65**	**0.65**	**0.70**
(2)	distiluse-base-multilingual-cased-v2	S-BERT	0.71	0.64	**0.65**	0.68
(3)	roberta-large	Klue RoBERTa	**0.74**	0.62	0.62	0.68

No.	BERT embedding	BERT	macro avg.			acc
			pre-cision	recall	f1 score	
(4)	KcELECTRA-base-v2022	ELECTRA	0.63	0.53	0.55	0.60
(5)	kcbert-large	BERT	0.65	0.51	0.54	0.58

4.5 적합한 BERT 모델의 선택

테스트 수행 결과 Table 5.에서 볼 수 있듯, Sentence BERT 2개 모델이 일반 BERT 모델보다 종합적인 측면에서 더 높은 성능을 보여주었다. 이는 Sentence BERT 모델이 생성한 임베딩이 문맥에 대한 정보를 실질적으로 더 깊게 담고 있기 때문으로 판단된다[2]. 반면 일반 BERT 모델의 경우 KcELECTRA, KcBERT모델보다 Klue-RoBERTa 모델이 더 좋은 성능을 보여주었다. Klue-RoBERTa모델의 경우 Table 6.에서 볼 수 있듯, 나머지 2개 모델보다 상품 명칭을 식별하는데 적합한 데이터로 pre- trained 되었기 때문에 일반 BERT 모델 중 가장 나은 성능을 보이는 것으로 판단된다[9, 10].

Table 6. Training data set and Number of parameters

Model	Data	Parameter
KorBERT	News Article, Encyclopedia 23GB	110M
KcBERT	News Article comment 15GB	109M
KlueBERT	Wiki, News, Petition 63GB	111M
KcELECTRA	Updated comment 17GB	124M

특이한 점은 precision 측면에 있어서는 Klue-RoBERTa 모델이 Sentence BERT 모델보다 더 좋은 성능을 보인다는 점이다. 이는 실험에 사용된 데이터의 특징에 그 원인이 있다.

본 연구에 사용된 데이터 세트는 온라인 쇼핑몰, 오프라인 쇼핑몰 등에서 수집한 다양한 '영업용 상품명칭' 이다. 따라서 일반적인 상품의 명칭보다 훨씬 더 구체적으로 상품에 대한 정보를 담고 있으며, 이로 인해 특정 브랜드, 특정 모델, 성별, 사이즈, 색상 등 정보가 반복되는 패턴의 데이터가 다수 존재한다.

Klue-RoBERTa모델은 상품 명칭을 토큰화 하여 그 결과를 바탕으로 임베딩을 수행하기 때문에, 이러한 반복 패턴 인식에는 좋은 성능을 보이지만, 패턴이 없거나 부족한 데이터를 마주했을 경우 Sentence BERT보다 성능이 떨어지는 모습을 보이는 것으로 판단된다[7].

다시 말해, Klue-RoBERTa모델은 불규칙 데이터에 대한 강건함이 Sentence BERT보다 낮다. 가령 Table 9.에서 볼 수 있듯, Klue- RoBERTa 모델 실험시 통조림류 상품이 있었던 'class 1' 의 경우, precision 0.95, recall 0.17이라는 극단적인 값을 보였는데, Table 7.에서와 같이 특정 브랜드를 패턴으로 인식해 해당 브랜드의 상품만을 통조림 상품으로 분류하는 모습을 보였다.

Table 7. Examples of high precision in the Klue-RoBERTa

Name	Class No.	Prediction
'OO(브랜드명) 살코기참치 g'	1	1
'OO(브랜드명) 김치찌개용참치g'	1	1
'OO(브랜드명) 키즈참치g'	1	1
'리오마레 올리브오일 참치 g'	1	0
'스타부르 고등어 토마토바질 g'	1	0
'식물성 참치 스리라차 스파이시 g'	1	6
....		

5. 다양한 분류기 추가 성능 평가

본 장에서는 4장에서 도출한 결과를 바탕으로, 다양한 분류기 구조를 추가적으로 실험하여 각 분류기별 성능평가를 수행한다. 구체적으로는, 상품 명칭 분류 작업에서 최적의 결과 값을 보인 Sentence BERT-CNN(baseline) 아키텍처를 baseline으로 상정하고, 분류기 모듈만을 대상으로 다양한 알고리즘을 교체 적용해보며 추가적 성능 향상이 있는지를 관찰하였다.

5.1 추가 분류기 정의 및 학습 진행

추가 분류기의 경우 ResNet (32), ResNet (Shallow), Transformer의 3종을 사용하였다. 이들 분류기에 feature-based 방식으로 사용 가능한 Sentence BERT의 문장 임베딩을[4] Input으로 하여 학습을 진행하였다.

본 장의 train, test 데이터의 분할은 위의 baseline 조건과 동일하게 8:2의 비율로 진행하였으며, stratify 옵션을 통해 클래스별로 균일하게 추출되도록 하였다. 최종적으로 각 분류기 모델들은 43,064개 데이터로 학습하였다. 따라서 모든 조건은 4장의 CNN 분류기(baseline)와 동일한 상태에서 학습을 수행하였다. 단, Transformer모델의 경우 낮은 epoch상태에서는 성능 평가가 무의미할 정도로 학습이 이루어지지 않아, 해당 case에 한하여 epoch=100 으로 학습을 수행하였다.

5.1.1 ResNet-34 분류기 정의 및 학습

ResNet은 깊은 신경망 구조에서도 기울기 소실(Gradient Vanishing)이 발생하지 않도록, 잔차 학습(Residual Learning) 방법을 도입한 모델이다[11]. 이러한 구조적 특성으로 인해, 분류 작업에 있어서 신경망을 깊게 가져가더라도 높은 계산 효율과 성능을 보인다.

본 연구에서는 He et al. (2016)이 제안한 구조를 바탕으로 본 연구의 작업에 적합하도록 ResNet-34를 구성하였다. ResNet 구조는 이미지 분류작업에서 높은 효율을 달성한 바 있으며, feature-based 텍스트 임베딩 기반의 작업에서 역시 이러한 효율을 보이는지 검토하기 위해 실험을 수행하였다. 학습은 CNN 분류기와 마찬가지로 10 epoch, 0.001의 Learning rate, Adam optimizer를 사용하였으며, 각 레이어의 구조는 Table 8.와 같으며, 학습 수행 결과 Loss Curve는 Figure 5.와 같다.

Table 8. The ResNet-34 architecture (in this experiment)

Layer	Channel	Kernel	Stride	Padding
Conv	128	5	1	2
MaxPool	-	5	5	-
Residual 3	128	5	1	2
Residual 4	128	5	2	2
Residual 6	256	5	2	2
Residual 3	512	5	2	2
AdaptiveAvgPool	-	-	-	-
Linear	num_class	-	-	-

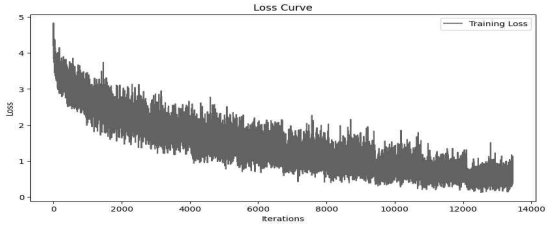

Figure 5. Loss Curve of ResNet-34 case (epoch=10)

5.1.2 ResNet-Shallow 분류기

본 연구에서는 기본적인 ResNet 구조 외에도, 얕은 수준의 ResNet 또한 실험을 진행하였다. 일반적으로 ResNet은 깊어질수록 효율을 발휘하는 모델이다. 그러나 깊은 네트워크의 경우, 깊이가 깊어질수록 과적합의 문제 또한 같이 발생하기 때문에, 데이터가 충분하지 않거나 불균형한 상황에서는 오히려 성능이 떨어질 수 있다.

따라서 본 연구에서는 깊이를 깊게 가져가지 않고 오직 Residual block 개념만을 적용한 ResNet-Shallow 구조 또한 실험을 수행하였다. CNN 분류기와 마찬가지로 10 epoch, 0.001의 Learning rate, Adam optimizer를 사용하였으며, 레이어의 구조는 Table 9.와 같으며, 학습 수행 결과 Loss Curve는 Figure 6.와 같다.

Table 9. The ResNet Shallow architecture

Layer	Channel	Kernel	Stride	Padding
Conv	128	5	1	2
MaxPool	-	5	5	-
Residual	128	5	1	2
Residual	128	5	2	2
Linear	num_class	-	-	-

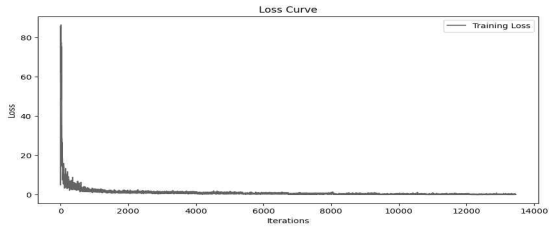

Figure 6. Loss Curve of ResNet-Shallow case (epoch=10)

5.1.3 Transformer 분류기

마지막으로 Transformer 분류기[12] 또한 성능 비교를 위해 실험을 수행하였다. head 수는 32, encoder layers는 2로 설정하였다. CNN 분류기와 마찬가지로 0.001의 Learning rate, Adam optimizer를 사용하였다. 다만 Transformer의 구조상 낮은 epoch으로는 학습이 충분하지 않은 문제가 있어 100의 epoch로 학습을 수행하였다. 학습 수행 결과 Loss Curve는 Figure 7.와 같다.

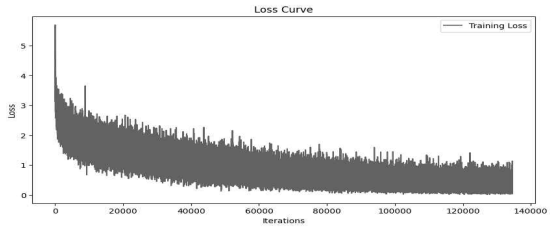

Figure 7. Loss Curve of Transformer case (epoch=100)

5.2 성능 측정 결과

Table 10. 은 Sentence BERT를 기반으로 생성한 문장 임베딩을 feature로 활용하여, ResNet-34, ResNet-Shallow, Transformer 분류기에 대해 학습시키고, 학습된 모델과 CNN 분류기(baseline)의 성능 비교를 진행한 결과이다.

baseline대비 성능의 향상은 ResNet-Shallow에서만 발견되었다. precision은 큰 변화가 없었으나 recall, f1 score, accuracy에서 baseline대비 좋은 성능을 보였다. 반면 깊은 네트워크일수록 더 좋은 성능을 발휘할 것으로 예상되었던 ResNet-34 구조의 경우, 동일 조건 하에서 오히려 baseline보다 성능이 낮아지는 모습을 보였다. Transformer의 경우 다른 분류기들과 동일조건에서 실험을 진행하고자 epoch=10을 설정하였으나, 실험이 불가능한 수준의 성능을 보여 epoch=100으로 대폭 증가시켰다. 그러나 이러한 변경에도 불구하고 baseline보다 낮은 성능을 보였다.

Table 10. Performance and Loss Values by Model

No.	Classifier	Loss	macro avg.			acc
			precision	recall	f1 score	
(1)	CNN baseline (epoch=10)	0.3692	**0.73**	0.65	0.65	0.70
(2)	ResNet-34 (epoch=10)	0.7017	0.64	0.58	0.59	0.65
(3)	ResNet-Shallow (epoch=10)	**0.0213**	**0.72**	**0.68**	**0.69**	**0.73**
(4)	Transformer (epoch=100)	0.2187	0.65	0.63	0.63	0.69

6. 결 론

4장에서 수행한 BERT 모델 선정 실험에서 Sentence BERT 2개 모델이 일반 BERT 모델보다 더 높은 성능을 보여주었다. 이는 Sentence BERT 모델이 생성한 임베딩이 문맥에 대한 정보를 실질적으로 더 깊게 담고 있기 때문으로 판단된다[5].

5장에서는 4장에서 선정한 Sentence BERT의 문장 임베딩을 feature로 사용하여, 다양한 분류기의 성능을 비교 분석하였다. 그 결과 ResNet (Shallow) 모델이 가장 뛰어난 성능을 보였다. 특이한 점은 일반적 통념과 달리 깊은 네트워크를 사용하더라도 성능 향상은 미비하였고, 오히려 얕은 네트워크를 사용한 CNN(baseline)과 ResNet-Shallow가 더 좋은 성능을 보였다는 점이다.

이러한 이유는 데이터 세트의 자연적 특성이 원인인 것으로 판단된다. 현실세계에 존재하는 상품의 명칭은 자연적으로 불균형 데이터 세트의 구조를 가진다. 가령 인간의 제품인 냉장고를 부를 때 그 명칭의 종류는 '양문형 냉장고', '김치냉장고', '미니냉장고' 등 다양할 수 있다. 반면 자연물일 경우 가령, 배추를 배추라는 이름으로 부르는 것 외에는 파생형이 많지 않다. 즉, 본 연구가 대상으로 삼고 있는 상품 명칭의 경우 자연적으로 데이터의 비대칭이 존재할 수밖에 없다.

데이터가 비대칭적으로 존재하는 상황에서 CNN 및 이를 기반으로 한 모델의 경우 과적합(Overfitting)이 발생할 확률이 높다[13]. 본 연구에서 더 깊은 깊이를 가진 ResNet-34의 경우 이러한 과적합 문제가 발생했을 가능성이 있다. 뿐만 아니라 Sentence BERT로 이미 잘 정제된 feature가 다시 깊은 네트워크에 들어가면서 정보가 소실되거나 훼손되었을 가능성 또한 존재한다.

실험에 사용된 Sentence BERT 모델은 multilingual 모델로, 한글 데이터 처리를 위한 pre-training 이 되어있지 않은 모델이다. 따라서 향후 기본 Sentence BERT 모델에 대하여 한글 Domain Adaptation 을 진행하여, 한글 문장 및 상품 명칭 식별 작업에 대해 추가 성능 향상을 이루도록 연구를 진행할 예정이다. 데이터 불균형을 극복하기 위한 데이터 증강 관련 연구를 추가 진행하고자 한다. 마지막으로 과적합 현상을 방지하기 위한 규제, 손실함수 검토, 가중치 부여 등 다양한 엔지니어링 기법들을 추가 적용해 성능 향상을 도모하고자 한다.

참고문헌

[1] Lim, J. H., Kim, H. K., & Kim, Y. K. (2020). Recent r&d trends for pretrained language model(딥러닝 사전학습 언어모델 기술 동향). *Electronics and Telecommunications Trends, 35*(3), 9-19. DOI : 10.22648/ETRI.2020.J.350302

[2] Song, J. S., & Rhew, S. Y. (2013). An Empirical Study on Quality Improvement by Data Standardization for Distributed Goods(유통 상품의 데이터 품질 관리를 위한 데이터 표준화에 대한 연구). *Journal of the Korea Society of Computer and Infor*

mation, 18(9), 101-109. DOI : 10.9708/jksci.2013.18.9.101

[3] Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805.* DOI : 10.48550/arXiv.1810.04805

[4] Lim, J., Whang, T., Oh, D., Yang, K., & Lim, H. (2019). Hypernews Detection using Sentence BERT Embedding. *In Annual Conference on Human and Language Technology*, 388-391.

[5] Reimers, N., & Gurevych, I. (2019). Sentence-bert: Sentence embeddings using siamese bert-networks. *arXiv preprint arXiv:1908.10084.* DOI : 10.48550/arXiv.1908.10084

[6] Kim, B., & Park, S. (2020). Sentence BERT for Measuring Sentence Similarity in Korean (한국어 문장 유사도 측정을 위한 Sentence BERT). *Proceedings of the Korea Software Congress 2020*, 1376-1378.

[7] Park, D., Lee, M., & Kim, Y. (2021). The Impact of Stopwords on BERT-Based Automatic Sentence Classifier (불용어의 BERT 기반 문장 자동분류기에 대한 영향). *Proceedings of the Korea Software Congress 2021*, 715-717.

[8] Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25.

[9] Lee, J. (2020). Kcbert: Korean comments bert. *Annual Conference on Human and Language Technology*, *437-440.*

[10] Park, S., Moon, J., Kim, S., Cho, W. I., Han, J., Park, J., ... & Cho, K. (2021). Klue: Korean language understanding evaluation. *arXiv preprint arXiv:2105.09680.* DOI : 10.48550/arXiv.2105.09680

[11] He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. *In Proceedings of the IEEE conference on computer vision and pattern recognition*, 770-778.

[12] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., ... & Polosukhin, I. (2017). Attention is all you need. *Advances in neural information processing systems*, *30.*

[13] Li, Z., Kamnitsas, K., & Glocker, B. (2020). Analyzing overfitting under class imbalance in neural networks for image segmentation. *IEEE transactions on medical imaging*, *40*(3), 1065-1077. DOI : 10.1109/TMI.2020.3046692

김 도 훈

2018년 연세대학교 철학과 (학사)

2022년 ~ 현재 고려대학교 컴퓨터정보통신대학원 석사과정
관심분야: 자연어처리, 언어생성, 텍스트 분류, 업무 자동화
E-Mail: mark_watney@korea.ac.kr

임 희 석

1992년 고려대학교 컴퓨터학과 (학사)
1994년 고려대학교 컴퓨터학과 (석사)
1997년 고려대학교 컴퓨터학과 (박사)

2008년 ~ 현재 고려대학교 컴퓨터학과 교수
관심분야: 자연어처리, 뇌신경 언어 정보 처리
E-Mail: limhseok@korea.ac.kr