

한글 텍스트 감정 이진 분류 모델 생성을 위한 미세 조정과 전이학습에 관한 연구⁺

(A Study on Fine-Tuning and Transfer Learning to Construct Binary Sentiment Classification Model in Korean Text)

김종수^{1)*}
(JongSoo Kim)

요약 근래에 트랜스포머(Transformer) 구조를 기초로 하는 ChatGPT와 같은 생성모델이 크게 주목받고 있다. 트랜스포머는 다양한 신경망 모델에 응용되는데, 구글의 BERT(bidirectional encoder representations from Transformers) 문장생성 모델에도 사용된다. 본 논문에서는, 한글로 작성된 영화 리뷰에 대한 댓글이 긍정적인지 부정적인지를 판단하는 텍스트 이진 분류모델을 생성하기 위해서, 사전 학습되어 공개된 BERT 다국어 문장생성 모델을 미세조정(fine tuning)한 후, 새로운 한국어 학습 데이터셋을 사용하여 전이학습(transfer learning) 시키는 방법을 제안한다. 이를 위해서 104개 언어, 12개 레이어, 768개 hidden과 12개의 집중(attention) 헤드 수, 110M 개의 파라미터를 사용하여 사전 학습된 BERT-Base 다국어 문장생성 모델을 사용했다. 영화 댓글을 긍정 또는 부정 분류하는 모델로 변경하기 위해, 사전 학습된 BERT-Base 모델의 입력 레이어와 출력 레이어를 미세 조정된 결과, 178M개의 파라미터를 가지는 새로운 모델이 생성되었다. 미세 조정된 모델에 입력되는 단어의 최대 개수 128, batch_size 16, 학습 횟수 5회로 설정하고, 10,000건의 학습 데이터셋과 5,000건의 테스트 데이터셋을 사용하여 전이 학습시킨 결과, 정확도 0.9582, 손실 0.1177, F1 점수 0.81인 문장 감정 이진 분류모델이 생성되었다. 데이터셋을 5배 늘려서 전이 학습시킨 결과, 정확도 0.9562, 손실 0.1202, F1 점수 0.86인 모델을 얻었다.

핵심주제어: 트랜스포머, 사전 학습된 버트, 미세조정, 전이학습, 한글 문장 감분류

Abstract Recently, generative models based on the Transformer architecture, such as ChatGPT, have been gaining significant attention. The Transformer architecture has been applied to various neural network models, including Google's BERT(Bidirectional Encoder Representations from Transformers) sentence generation model. In this paper, a method is proposed to create a text binary classification model for determining whether a comment on Korean movie review is positive or negative. To accomplish this, a pre-trained multilingual BERT sentence generation model is fine-tuned and transfer learned using a new Korean training dataset. To achieve this, a pre-trained BERT-Base model for multilingual sentence generation with 104 languages, 12 layers, 768 hidden, 12 attention heads, and 110M parameters is used. To change the pre-trained BERT-Base model into a text classification model, the input and output layers were fine-tuned, resulting in the creation of a new model with 178 million parameters. Using the fine-tuned model, with a maximum word count of 128, a batch size of 16, and 5 epochs, transfer learning is conducted with 10,000 training data and 5,000 testing data. A text sentiment binary classification model for Korean movie review with an accuracy of 0.9582, a loss of 0.1177, and an F1 score of 0.81 has been created. As a result of performing transfer learning with a dataset five times larger, a model with an accuracy of 0.9562, a loss of 0.1202, and an F1 score of 0.86 has been generated.

Keywords: Transformer, pre-trained BERT, fine-tuning, transfer learning, Korean text sentiment classification

* Corresponding Author: avantas@naver.com
Manuscript received June 26, 2023 / revised September 02,

2023 / accepted October 10, 2023
1) ㈜콜드브릿지 기업부설연구소, 교신저자

1. 서론

근래에 들어, OpenAI에서 트랜스포머(transformer) 모델을 응용해서 만든 딥러닝 언어 생성모델 ChatGPT(generative pre-trained transformer)가 크게 주목받으며 관련연구도 활발하게 진행되고 있다(O. Dongsuk, P. Sungjin, L. Hanna, J. Yoonna, and L. Heuiseok 2021; K. EunJung 2022; K. SeongAn, K. SoHui and R. Min Ho 2022; L. DoegGyu, Kyungkeun B, L. HyungDong, and S. Sunhee 2023).

ChatGTP는 트랜스포머 모델의 디코더만 가지고 있는 구조로, 문장생성에 유리한 모델이다. 신경망학습에 메타학습(meta-learning)을 적용했는데, 주요 특징으로는 175억 개의 파라미터를 가지도록 구성된 신경망에 대규모 코퍼스로 사전 학습된 GPT 기본모델을 바탕으로 지도미세조정(supervised fine-tuning) 과정을 거쳐서 생성된 모델에 인위적인 보상을 추가한 모델(reword model)로 다시 학습시키고, 또다시 강화학습정책(proximal policy optimization) 알고리즘을 적용해서 학습시켰다는 것이다(S. John, W. Filip, D. Prafulla, R. Alec and K. Oleg, 2017; B. Tom B, M. Benjamin, R. Nick, et al., 2020; L. Ouyang, W. Jeff, X. Jiang, A. Diogo, et al., 2022; B. Sébastien, C. Varun, E. Ronen, et al., 2023).

버트(BERT: bidirectional encoder representations from transformers)는 구글에서 트랜스포머를 더 발전시킨 모델이다. 주요 특징으로 학습할 문장의 중간에 마스크(mask)를 설정해서 들어갈 단어를 예측할 수 있도록 학습된다. 2018년 11월 처음 공개된 모델은 발표 당시 NLP (natural language processing, 자연어처리) 11개 분야에서, 현재 최고 수준(SOTA: state of the art)을 갱신했다(D. Jacob, C. Ming-Wei, L. Kenton and T. Kristina, 2019).

본 논문에서는 주어진 한글 문장이 긍정적인지 부정적인지를 판단하는 이진 분류모델을 생성하기 위해서, 사전 학습된 BERT-Base 다국어 문장생성 모델을 주어진 문장의 감정 이진 분류를 위한 모델로 미세조정(fine tuning)한 후,

새로운 학습 데이터셋을 사용하여 전이학습(transfer learning) 시키는 방법에 관해 연구한다. 본 논문에서 제시한 방법을 응용하면, 영화에 대한 리뷰에 대하여 긍정 부정의 감성을 분류하는 이진 모델 외에 다양한 분야에서 생성되는 텍스트의 속성을 파악할 수 있는 분류모델을 생성할 수 있다.

2. 관련연구

2.1 텍스트 감정 이진 분류 Colab 기본모델

영화 리뷰나 쇼핑몰 상품 구매 후기와 같이 어떤 텍스트가 긍정적인지 부정인지를 판별하기 위해 신경망을 구성하는 몇 가지 방법이 있다. 텐서플로우 저자들이 2019년에 코랩(Colab)에 공개한 영문 텍스트 감정 이진 분류모델의 신경망 구조는 Fig. 1과 같다(TensorFlow Authors, 2019).

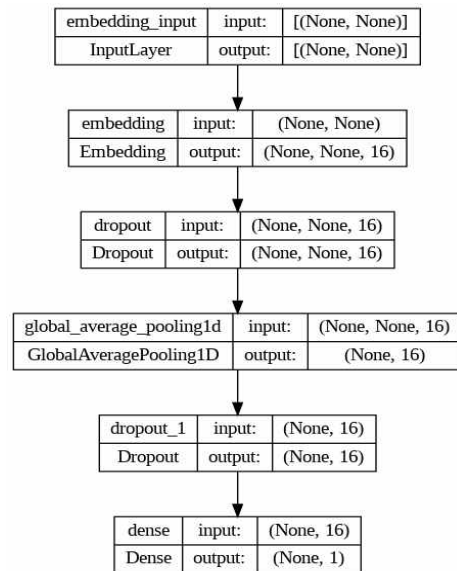


Fig. 1 Example of an English text sentiment binary classification model

keras.Sequential 클래스로 임베딩, 드롭아웃, 밀집(Dense) 레이어로 구성된 기본모델이라고 할 수 있다. 50,000개의 영화 리뷰 텍스트가 포

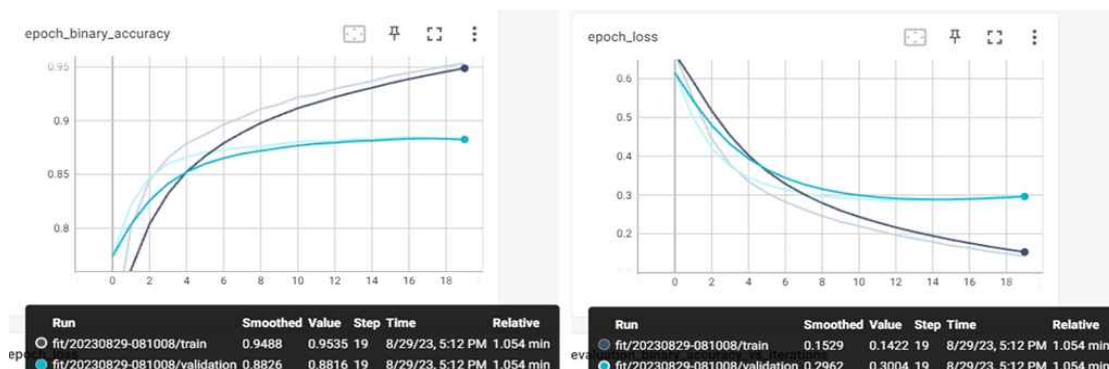


Fig. 2 Graph of learning and validation accuracy(left), training and validation loss(right)

함되어 있는 영문 IMDB 데이터셋을 학습용 리뷰 25,000개와 테스트용 리뷰 25,000개로 나누어 20회 학습시킨 신경망의 학습 횟수에 따른 성능 지표 변화는 Fig. 2에서 볼 수 있다.

그래프에서, 학습데이터의 정확도와 손실은 학습 횟수를 거듭할수록 향상되지만, 테스트 데이터셋을 이용한 검증에서는, 대략 15회 학습 전후로는 크게 성능향상을 보이지 않는다는 것을 확인할 수 있다.

BinaryCrossentropy 손실(loss)과 adam 최적화기, 정확도(accuracy) 와 F1 점수(score)를 계산하도록 모델을 컴파일 한 후, 테스트 데이터셋을 사용하여 성능 평가를 한 결과, 정확도 0.8678, 손실 0.3357인, F1 점수 0.8626의 성능을 보이는 모델을 얻었다.

2.2 RNN(Bidirectional) 기반 영문 텍스트 분류모델

구글 코랩에 소개된 순환신경망(recurrent

neural network) 기반 텍스트 감정 이진 분류모델의 Fig. 3과 같은 구조를 가진다(Google Colab, 2023).

앞의 기본모델에 비해 달라진 점은 TextVectorization 레이어의 추가와 입력 부분의 신호와 출력 신호의 상호간의 효율적인 가중치 전파를 위해 양방향 RNN(Bidirectional) 레이어를 추가시킨 것이다.

영어로 구성된 IMDB의 영화 리뷰 데이터셋을 사용하여 10회 학습된 결과, 정확도 0.8746, 손실 0.2968, F1 점수 0.8678의 성능을 보였다.

양방향 순환신경망을 중첩해서 신경망을 구성하면, 성능향상을 보이는 경우가 있다. 2개의 양방향 순환신경망 레이어를 사용하여 10회 학습된 결과, 정확도 0.8754, 손실 0.2955, F1 점수 0.8665의 성능을 보였다.

2.3 BERT 기반 영문 텍스트 분류 모델

코랩에 소개된 BERT 기반 영문 텍스트 감정

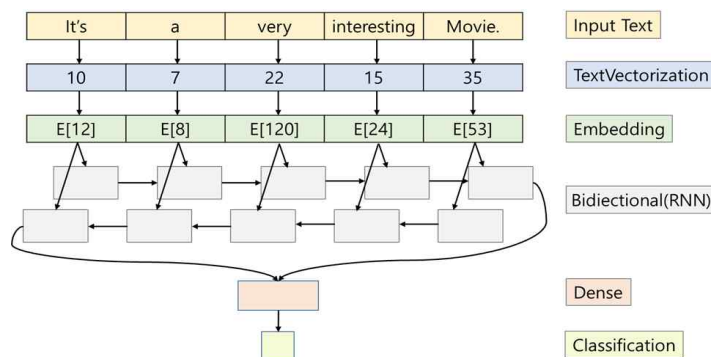


Fig. 3 RNN(recurrent neural network) model for text binary classification

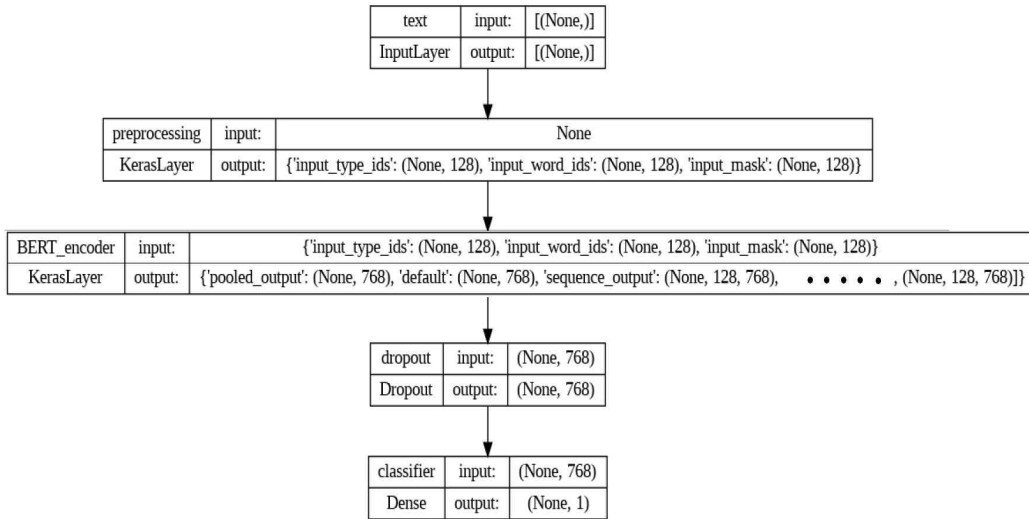


Fig. 4 Structure of a pre-trained BERT-based model for an English text binary classification

이진 분류모델은 TF2.0 저장되어 있는 모델(v4)로 Fig. 4와 같은 구조를 가진다(TensorFlow Hub Authors, 2020). 사전 학습된 bert_multi_cased_L-12_H-768_A-12 모델은 히든 레이어(hidden layers)인 트랜스포머 블록 L=12, 숨겨진 크기(hidden size) H=768 그리고, 집중 헤드(attention heads) A=12로 구조화되어 있다. AdamW 최적화기를 사용했으며, 다국어 위키피디아(Wikipedia) 수집된 여러 언어에 대해 학습되었고, 텍스트 입력은 "대소문자 구분" 방식으로 정규화 되었다.

영어로 구성된 IMDB의 영화 리뷰 데이터셋을 사용하여 5회 학습된 모델은 정확도 0.9805, 손실 0.0672, F1 점수 0.9792의 성능을 보였다.

2.4 거대파라미터 트랜스포머 연구 동향

S. John et al.(2017)이 발표한 논문에 의하면, 언어모델은 더 많은 학습 데이터를 사용하면 성능이 더욱 높아지며, K. Jared et al.(2020)의 연구에 의하면, 규모와 정확도 사이에 log(로그) 선형관계가 있어서, 언어모델 규모가 크면 클수록 정확도가 높아지는 경향을 보인다. 일반적으로 언어모델은 트랜스포머 모델 구조의 출력(output) 구조를 응용하는 경우가 많다. 트랜스포머는 다양한 데이터입력으로부터 원하는 출력을 얻을 수 있도록 고안된 신경망으로, RNN(recurrent neural network)과 다르게 학습에 GPU 병렬처리 연산을 사용할 수 있다는 것이

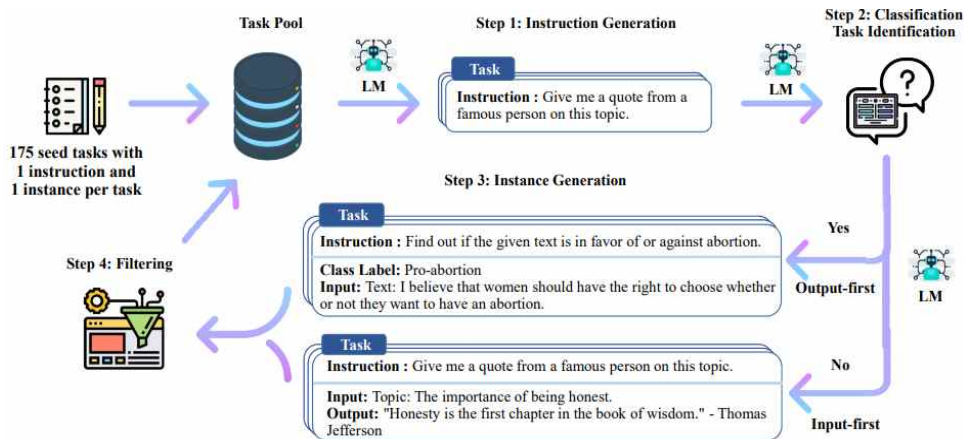


Fig. 5 A high-level overview of the self-instruct

Table 1 List of InstructGPT models provided by OpenAI

TRAINING METHOD	MODELS
SFT: A guided fine-tuning model for human demonstration	davinci-instruct-beta1
FeedME: Supervised fine-tuning model for samples rated 7/7 by human labelers and human-written demonstrations in overall quality score	text-davinci-001, text-davinci-002, text-curie-001, text-babbage-001
PPO: Reinforcement learning model using a reward model trained on human comparisons	text-davinci-003

대표적인 특징이다(V. Ashish, S. Noam, P. Niki, U. Jakob, J. Llion, G. Aidan N., K. Lukasz and P. Illia, 2017).

트랜스포머 구조 모델은 기계번역, 대화형 AI, 음성인식, 추천시스템, 이미지캡셔닝, 자동요약, 감정분석과 같은 자연어처리 분야에서 우수한 성능을 보인다. 거대파라미터 언어모델에서 self-instruct와 같이 사람의 손을 거치는 부분을 최소화하는 방향으로 보상(reward) 모델을 강화 학습시키는 관련연구도 있다(W. Yizhong, K. Yeganeh, M. Swaroop, et al., 2022). Fig. 5에서 모델의 학습방법을 볼 수 있다.

트랜스포머를 응용한 대표적인 연구로 BERT, GPT, T5(Text-to-Text Transfer Transformer), Alpaca 등이 있다.

GPT4와 관련된 연구로 버클리(UC Berkeley), CMU, 스탠포드(Stanford) 및 샌디에고(UC San

Diego)의 대학원생 그룹에서 ShareGPT와 데이터와 라마(LLaMA) 모델을 미세조정해서 만든 오픈소스 챗봇인 비쿠나(Vicuna)를 300\$ 정도의 비용을 들여서 학습시켜서 생성한 모델은, GPT-4를 예비평가자로 사용하여, 글쓰기, 역할극, 상식, 코딩, 수학, 시사, 지식, 페르미(Fermi) 추론, 반사실적의 성능을 평가한 결과, OpenAI의 ChatGPT3.5 및 구글 Bard의 90% 이상의 품질을 달성했다.

ChatGPT와 같은 모델을 만드는 데 사용되는 사전 학습된 InstructGPT 모델들의 학습방법과 모델명을 Table 1에서 볼 수 있다.

ChatGPT에 적용된 강화학습정책(proximal policy optimization) 알고리즘은 Fig. 6과 같은 방식이다.

프롬프트로 주어진 문장을 입력으로 사용하여, 초기 언어모델(initial language model)과 조

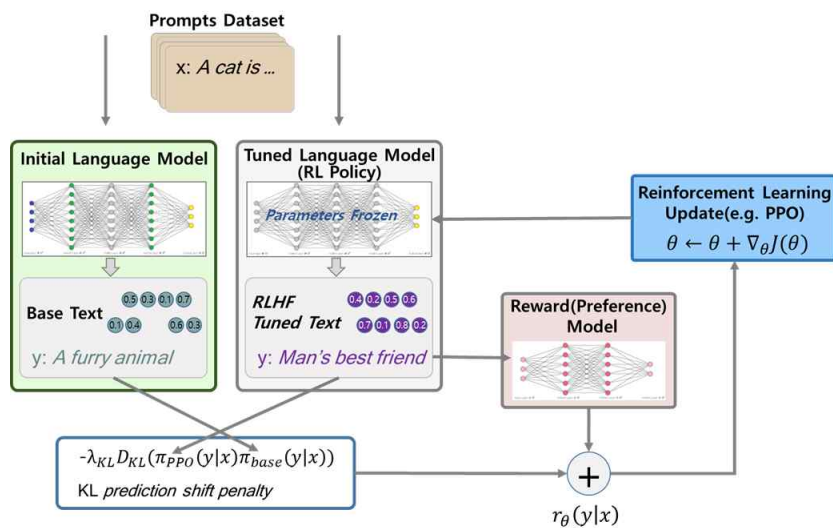


Fig. 6 GPT's proximal policy optimization algorithm

정된 언어모델(tuned language model)이 각각 예측한 결과 값을 합한 후, 인위적인 보상을 추가하는 모델(reward model)에서 주어진 보상을 추가하는 강화학습을 통하여 조정된 언어모델(tuned language model)을 계속해서 학습시킨다. 인간피드백을 통한 강화 학습(reinforcement learning with human feedback) 학습은 6B 이상의 데이터셋 생성을 위한 주식 선별, 엄밀한 데이터셋 등이 필요하다.

2.5 BERT 신경망 구조 분석

D. Jacob et al.에 의해 2019년에 발표된 BERT는 트랜스포머 모델을 기반으로 한 양방향 언어 모델로, 이전 단어와 이후 단어를 모두 고려하여 문장을 이해할 수 있는 인공지능으로 개발되었다. 일부 성능 평가에서는 인간보다 더 높은 성능을 기록했으며, 사전학습(pre-training), 파인 튜닝(fine turning)과 전이학습(transfer learning), 랜덤 마스킹, 양방향 레이어(bidirectional layer) 사용과 같은 주요 특징들이 있다.

구글에서 발표된 모델은 Base 모델과 Large 모델이 있다. 현재 최고 수준(SOTA)를 달성했던 모델은 BERT Large 모델이다. 각각의 하이퍼파라미터는 다음과 같다.

- ①BERT-Base(기본) 모델: 하이퍼파라미터
 - NUM_LAYERS: 12
 - D_MODEL: 768
 - NUM_HEADS: 12
- ②BERT Large 모델의 하이퍼파라미터
 - NUM_LAYERS: 24
 - D_MODEL: 1024
 - NUM_HEADS: 16

BERT-Base 모델은 약 33억(위키백과(25억) + BooksCorpus(8억)) 단어의 데이터셋(dataset), subword 인코딩을 위한 토큰화 진행 후, 15% 비율로 마스킹된 단어를 생성해서 모델이 단어를 예측할 수 있도록 학습시켰다. 앞 문장과 뒤 문장을 합한 최대 단어 수는 512개로 설정하고, 학습 횟수 100만 스텝(step) 학습시켰고, 아담(adam) 최적화기를 사용했다.

공개된 영어기반의 영화 리뷰 감정 이진분류에 BERT 모델이 기본모델이나 RNN모델보다 나쁘지 않은 성능을 보이므로, 거대언어모델은 기본모델이나 RNN 기반 모델보다, 다소 레이어의 과적합이 있다고 하더라도, GhatGPT와 같은 다양한 기능을 수행하는 신경망으로의 발전시킬 수 있다는 장점이 있다.

Table 2 Key features of artificial intelligence for Korean natural language processing

Function	Description
Text Analysis and Tokenization	Divides text into meaningful units such as words, phrases, and sentences.
Morphological Analysis	Extracts and analyzes morphemes in Korean, understanding the combination of stems and endings.
Part-of-Speech Tagging	Attaches part-of-speech information to morphemes, understanding the sentence structure.
Named Entity Recognition (NER)	Recognizes and extracts named entities (organizations, people, places, dates, etc.) in text.
Syntax Parsing	Analyzes sentence structure and grammatical relationships to understand meaning.
Sentiment Analysis	Analyzes the sentiment of text (positive, negative, neutral).
Machine Translation	Performs translation between Korean and other languages.
Question Answering	Extracts or generates accurate answers to user questions.
Automatic Summarization	Summarizes long text documents into key information.
Chatbots	Implements AI chatbots that mimic conversations with users, supporting meaningful interactions.

2.6 한글 자연어 처리에 관한 연구 조사

한글 자연어처리는 영어와는 언어적 특성, 문법, 어휘, 문화적 차이 등에서 차이가 있다. 한국말을 한국어족으로 분류하는 견해와, 우랄알타이어족에 속한다는 견해, 한국-만주어파로 분류하는 견해 등이 있다. 영어와 차이점은 주어-목적어-동사(SOV)의 어순 사용, 어근에 접미사를 붙여서 새로운 단어를 만드는 교착어, 어휘가 풍부하고, 은유와 비유를 많이 사용한다는 등의 특징이 있다. 한국어 자연어 처리를 위한 인공지능의 주요 기능은 Table 2에서 볼 수 있다. 한국어 자연어 처리에 대한 연구도 국내 주요 기업과 학교 및 연구 기관에서 지속적으로 진행되고 있다. github에 “Awesome Korean NLP Papers”라는 제목으로 한국정보과학회, 언어공학연구회, 한국어정보학회 등에서 발표한 한국어 자연어 처리 관련 논문을 소개하고 있다 (S. Chang-Uk, 2020).

한국어 자연어 처리를 위한 BERT 기반의 사전 학습된 언어 모델에 대한 비교 평가를 위한

연구로, L. Eunchan, L. Changhyeon과 A. Sangtae가 2022년에 발표한 연구가 있으며, 해당 논문에서는 사전 학습된 RoBERTa-BASE (RoBERTa), XLM-RoBERTa-BASE(XLM-RoBERTa), BERT-BASE-Multilingual-uncased (BERT-M-uncased), BERT-BASE-Multilingual-cased(BERT-M-cased), KoBERT-BASE(KoBERT), KLUE-RoBERTa-BASE(KLUE-RoBERTa), KLUE-BERT-BASE(KLUE-BERT)의 7개 모델을 선택하여 다중 클래스 분류를 위한 신경망을 구현한 후, 성능을 평가했다.

3. 한글 문장 감성 분류모델 생성

3.1 BERT-Base 미세조정 (fine tuning)

사전 학습된 모델을 미세조정하면, 학습 시간을 절약할 수 있다는 장점이 있다. 한글 문장 감성 이진 분류 모델구현은 텐서플로우와 케라스를 사용했다. 인터넷에 공개된 사전 학습된

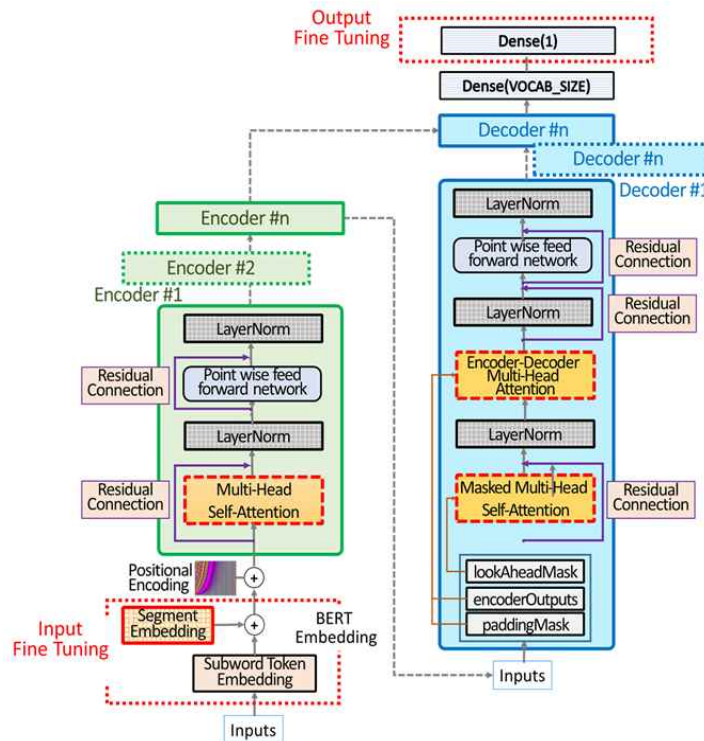


Fig. 7 A fine tuning model of BERT

BERT-Base 모델은 다음 같은 특징이 있다.

- 학습 데이터셋(dataset): 약 33억 단어(위키백과(25억) + BooksCorpus(8억))
- 104개의 언어로 학습됨
- Subword 인코딩을 위한 토큰나이저(tokenizer): 토큰화 진행 후, 15% 비율로 마스크된 단어를 생성해서 모델이 단어를 예측할 수 있도록 학습됨
- 앞 문장과 뒷 문장을 합한 최대 단어 수: 512
- 훈련 회수(epochs): 100만 스텝(step) 훈련 (33억 단어에 대해 40 epochs 학습)
- 최적화기(optimizer): 아담(adam)

한글 문장의 감정 이진 분류를 위한 BERT 모델의 구현과정은 다음과 같다.

- ① 학습을 위한 데이터셋 생성
- ② 사전 학습된(Pre-trained) BERT-Base 모델 설치
- ③ 사전 학습된 BERT-Base 모델과 가중치(weight) 로딩
- ④ 로딩된 번역기용 BERT-Base 모델에서 입력과 출력을 미세조정(fine tuning)하여, 주

어진 문장에 대해 긍정 또는 부정의 감성을 이진 분류하는 BERT 모델 생성

- ⑤ 데이터셋을 이용한 추가학습(transfer learning)
- ⑥ 학습된 모델을 이용한 평가

입력된 문장의 감정 분류를 수행할 수 있도록 Fig. 7와 같이 BERT-Base 모델의 입출력을 미세 조정(Fine Tuning)한다.

트랜스포머와 비교하여 BERT에서 새로 추가된 임베딩(embedding)은 서브워드 토큰 임베딩(subword token embedding)과 세그먼트 임베딩(segment embedding)이다. 입력(input)과 출력(output) 레이어의 미세조정 내용은 다음과 같다.

- ① Input Tuning: Subword Token Embedding + Segment Embedding
- ② Output Tuning: 다중 클래스 분류를 담당했던 Softmax 출력을 잘라낸 후, 영화 리뷰가 긍정인지 부정인지를 판단하는 Dense(1) 추가

사전 학습된 BERT-Base 모델을 한글 문장 감정 분석 분류모델로 미세 조정하는 과정은 다음과 같다.

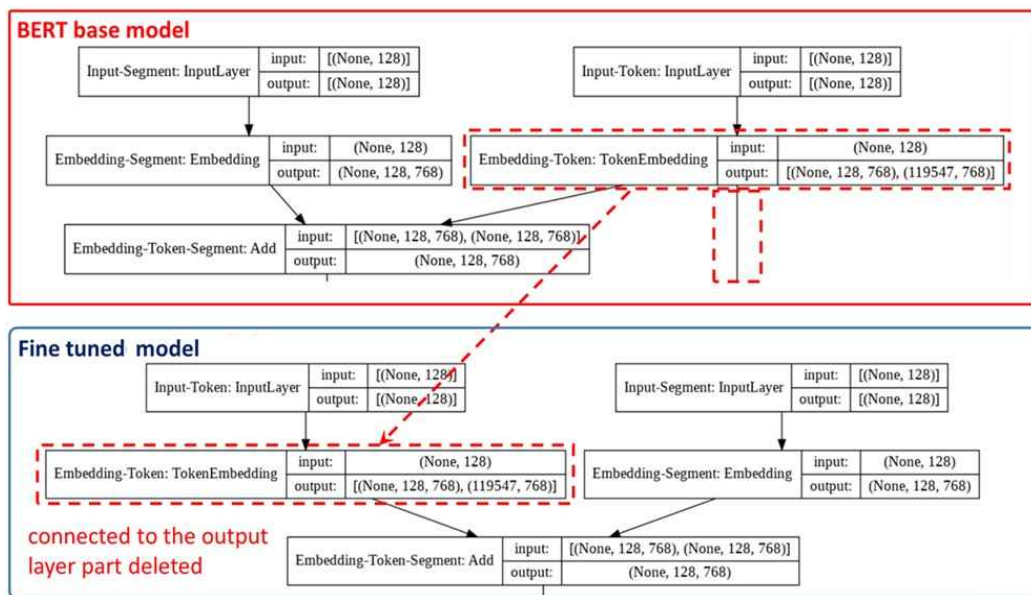


Fig. 8 Comparison of model input layers before(top) and after(bottom) refinement

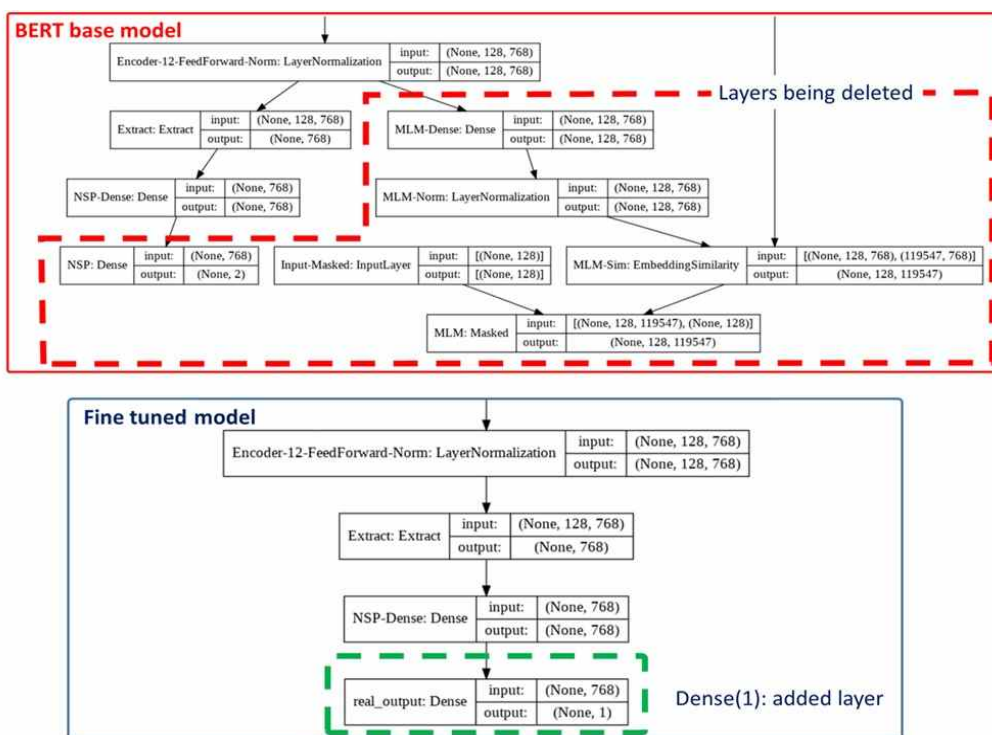


Fig. 9 Comparison of model output layers before(top) and after(bottom) refinement

- ① `inputs = model.inputs[:2]` → 서브워드 토큰 임베딩(subword token encoding)과 세그먼트 임베딩(segment encoding)을 수행하기 위한 레이어 정의
 - 서브워드 토큰 임베딩(subword token embedding): 입력(input)된 문장을 BERTEncoding으로 변환 후, 학습을 진행하는 레이어
 - 세그먼트 임베딩(segment embedding): 해당 문장이 앞에 나오는 문장인지 뒤에 나오는 문장인지 알려준 후, 학습을 진행하는 레이어
- ② `dense = model.layers[-3].output` → 전체 모델에서 출력을 맡은 마지막 3개 레이어 제거
- ③ `outputs = keras.layers.Dense(units=1)` → `binary_crossentropy` 손실함수를 사용하여 이진 분류를 수행하는 Dense(1) 추가
 - Dense(1)은 댓글이 부정에 가까우면 0, 긍정에 가까우면 1을 출력하는 레이어

Fig. 8에서 미세조정 전후에 대하여, 입력(input) 레이어의 변화를 비교할 수 있다.

BERT-Base 모델이 가지고 있던 “Embedding-Token: TokenEmbedding” 레이어와 출력(output) 레이어의 연결을 끊은 후, 주어진 문장의 감성이 긍정문인지 부정문인지를 최종적으로 계산하는 Dense(1) 레이어가 미세 조정될 모델의 가장 마지막에 추가된다. Fig. 9에서 미세조정 전후에 대하여 출력(output) 레이어의 변화를 비교할 수 있다.

기존의 BERT-Base 모델에서 삭제되는 레이어는 위부분에 빨간색 점선 테두리로 표시되어 있다. 트랜스포머의 토큰 임베딩(token embedding)을 더 세분화시킨 서브워드 토큰 임베딩(subword token embedding)을 수행한 후, 문장의 전후 관계를 알려주는 세그먼트 임베딩(segment embedding)을 추가하고, 다음으로 포지셔널 임베딩(positional embedding)을 추가한다. 미세 조정된 모델의 구조는 Fig. 10과 같다.

한글 문장 감성 분류를 위해 미세 조정된 모델은 177,559,297개의 학습 파라미터를 가진다는 것을 확인할 수 있다.

Layer (type)	Output Shape	Param #	Connected to
Input-Token (InputLayer)	[(None, 128)]	0	
Input-Segment (InputLayer)	[(None, 128)]	0	
Embedding-Token (TokenEmbedding)	(None, 128, 768)	91812096	Input-Token[0][0]
Embedding-Segment (Embedding)	(None, 128, 768)	1536	Input-Segment[0][0]
Embedding-Token-Segment (Add)	(None, 128, 768)	0	Embedding-Token[0][0] Embedding-Segment[0][0]
Embedding-Position (PositionEmbedding)	(None, 128, 768)	98304	Embedding-Token-Segment[0][0]
skipping the middle			
Encoder-12-FeedForward-Norm (LayerNormalization)	(None, 128, 768)	1536	Encoder-12-FeedForward-Add[0][0]
Extract (Extract)	(None, 768)	0	Encoder-12-FeedForward-Norm[0][0]
NSP-Dense (Dense)	(None, 768)	590592	Extract[0][0]
real_output (Dense)	(None, 1)	769	NSP-Dense[0][0]
Total params: 177,559,297			
Trainable params: 177,559,297			
Non-trainable params: 0			

Fig. 10 Structure of a fine-tuned Korean sentence sentiment classification model

3.2 한글 감성 분류 모델 학습데이터 생성

한글 문장 감성 분류모델로 미세 조정된 모델은 새로 추가된 레이어의 학습을 위해 데이터셋이 필요하다. 본 논문에서 사용된 네이버 영화 감성분석 데이터는 학습데이터 150,000건 테스트데이터 50,000건의 크기를 가진다. 신속한 학습을 위해 판다스(pandas) 라이브러리를 사용하여 학습 데이터 10,000건, 테스트 데이터는 5,000건으로 나누었다. Table 3에서 모델의 전이 학습을 위한 데이터셋의 샘플을 볼 수 있다.

Table 3 Example of model training dataset

document	label
아 더빙. 진짜 짜증나네요 목소리	0
흠...포스터보고 조딩영화줄...오버연기조차 가볍지 않구나	1
너무재밌었다그래서보는것을추천한다	0
교도소 이야기구면...솔직히 재미는 없다.평점 조정	0
사이몬텍의 익살스런 연기가 돋보였던 영화!스파이더맨에서 뉘어보이거만 했던 커스틴 ...	1

주요 컬럼은 다음과 같은 정보를 담고 있다.

- [document]: 사용자 리뷰
- [label]: 리뷰가 긍정(1)인지 부정(0)인지를 판별한 결과

주어진 문장의 의미를 미세 조정된 모델에게 인식시키기 위해서 토큰 생성과 임베딩과정을 거친다. 토큰을 생성하는 다양한 방법이 있는데, 토큰 생성 방식은 모델의 성능에도 큰 영향을 미친다. 한글문장 인식에 특화된 토큰을 생성하기 위해 BERTTokenizer를 구현했다. 특징은 다음과 같다.

- ① Subword 단위 토큰 생성
- ② 한국어를 형태소로 변환시켜야 하는 문제를 해결
- ③ 관련이 있는 단어들을 서로 밀접하게 연관 시킴
- ④ 단어에 포함되면서 단어의 시작이 아닌 문자는 ##이 붙음

“전이학습을 통한 텍스트 감성분류”라는 문장과 “사전 학습된 버트 파인튜닝”이라는 문장이 BERT Subword Tokenizer를 거치면, 단어 집합에는 Fig. 11과 같은 토큰들이 생성된다.

출력결과에서, [CLS] 토큰은 문장의 시작을 나타내고, [SEP] 토큰은 문장의 끝을 나타낸다. 연속되는 2문장이 서로 관련이 있으면, [CLS]→

```
['[CLS]', '전', '##이', '##학', '##습', '##을', '##통', '##한', '##텍', '##스트', '##감', '##정', '##분', '##류', '[SEP]']
['[CLS]', '사', '##전', '##학', '##습', '##된', '##버', '##트', '##파', '##인', '##류', '##닝', '[SEP]']
```

Fig. 11 An example of token generation

[SEP]→[SEP] 순으로 문장의 시작 토큰과 끝 토큰이 나타나고, 서로 관련이 없는 두 문장이 입력되면 [CLS]→[SEP]→[CLS]→[SEP] 순으로 토큰이 나타난다. 전체 데이터셋을 사용해서 생성된 토큰 사전으로 두 문장을 임베딩한 예를 Fig. 12에서 볼 수 있다.

```
[array([[ 101, 9519, 9074, ..., 0, 0, 0],
 [ 101, 100, 119, ..., 0, 0, 0],
 [ 101, 9004, 32537, ..., 0, 0, 0],
 ...,
 [ 101, 100, 12508, ..., 0, 0, 0],
 [ 101, 9451, 33077, ..., 0, 0, 0],
 [ 101, 8924, 118729, ..., 0, 0, 0]])]
```

Fig. 12 Example of embedding the dataset

배열 요소 값에서 101은 [CLS] 토큰을 나타낸다. BERT가 수행하는 Segment Embedding과 포지셔널 임베딩(Positional Encoding)은 Fig. 13과 같이 수행된다.

입력(input)에서 [CLS] 토큰은 BERT가 분류 문제를 풀기위해 사용하는 특별한 토큰이고, [SEP] 토큰을 문장의 시작과 끝에 추가하여 문장을 구분한다. 또한 “색다른”이라는 단어는 서브워드 토큰나이저(subword tokenizer)를 거치면 “색”, “##다”, “##른”으로 분리되고, “느낌”이라는 단어도 “느”, “##낌”이라는 단어로 분리된다. “##” 기호는 앞에 다른 단어가 있다는 뜻이다. 이러한 BERT 임베딩은 한국어를 형태소로 변환해야 하는 문제를 해결하고, 의미가 있는 단어들을 서로 밀접하게 연관시킬 수 있다는 장점이 있다.

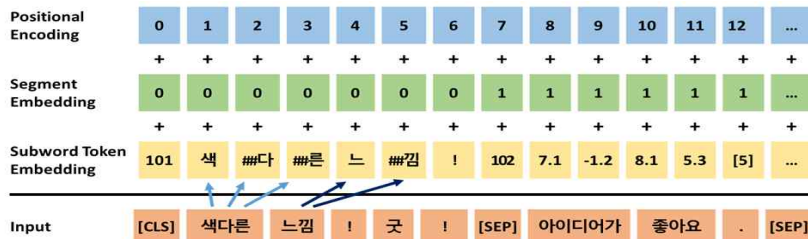


Fig. 13 BERT embedding procedure

(①Subword Token Embedding → ②Segment Embedding → ③Positional Encoding)

3.3 전이학습 과정

구글 코랩에서 T4 GPU 사용을 설정한 후, 모델을 학습시켰다. 학습을 위한 주요 하이퍼파라미터는 다음과 같다.

- ① batch_size: 16(크게 설정하면 메모리 초과 문제가 발생할 수 있음)
- ② epochs: 5(dataset을 이용한 학습 회수, 1 epochs에서도 높은 정확도가 나옴)
- ③ validation_data = (testX, testY): 정확도 계산을 위한 데이터 셋(testDataset)

성능이 우수한 머신과 충분한 데이터로 학습된 모델이 대부분의 경우, 우수한 성능을 나타내지만, 모델의 빠른 학습을 위해, 학습데이터셋의 크기를 10,000건, 테스트 데이터셋의 크기로 5,000건으로 조정된 후, 학습시켰다. 모델이 가진 추상메서드 fit()은 배치크기(batch_size) 16, 데이터 섞기(shuffle) 참(True), 전체 데이터 학습 횟수(epochs) 5회로 설정했다.

5epochs 학습시킨 결과에서, validation loss가 줄지 않고 있으며, validation accuracy가 많이 증가하지 않고 있으므로 과적합(overfitting)인지 의심해볼 여지가 있지만, 정확도가 0.9582인 모델을 얻었다. 신경망의 과적합을 방지하는 일반적인 방법은 다음과 같다.

- ① 더 많은 학습데이터를 사용한 학습
- ② 네트워크 용량 줄임
- ③ 가중치 정규화 레이어 추가함
- ④ 드롭아웃 추가

Table 4 Performance comparison according to the amount of learning data

Category.	Dataset		loss	Acc.	Val loss	Val Acc.	F1 Score
	Train	Test					
Basic Mode	IMDB(English)		0.1407	0.9536	0.3008	0.8818	0.86
RNN(1 layer)			0.2968	0.8678	0.3194	0.8562	0.85
RNN(2 layer)			0.2955	0.8754	0.3187	0.8484	0.84
BERT			0.0672	0.9805	0.7291	0.8604	0.86
BERT	NEWS Review		0.0929	0.9656	0.2488	0.9124	0.91
Proposed BERT model (dataset=nsmc)	10000	5000	0.1177	0.9582	0.5237	0.8258	0.81
	20000	10000	0.0669	0.9497	0.6543	0.8430	0.84
	30000	15000	0.1107	0.9604	0.4917	0.8467	0.85
	40000	20000	0.1423	0.9477	0.4109	0.8576	0.86
	50000	25000	0.1202	0.9562	0.4135	0.8621	0.86

4. 성능 평가

한글 학습데이터셋 10,000개, 테스트데이터셋 5,000개로 학습된 모델에 대해, 테스트 데이터셋을 사용하여 확인한 결과, F1 점수 0.81과 정확도 0.81이 일치함으로써, 모델 학습이 어느 한 쪽에 치우치지 않고 성공적으로 수행되었음을 알 수 있다.

미세 조정된 모델의 과적합 여부를 판단하기 위해, 데이터셋을 2~5배씩 증가시켜서 생성된 모델들의 학습결과와 2장 관련 연구에서 조사된 모델들과 논문에서 제안된 모델들의 성능을 Table 4에 정리했다.

영문 텍스트만 분류할 수 있는 기본모델과

RNN모델들에 비해, BERT로 구조의 모델들이 성능 지표 대부분에서 높은 것을 볼 수 있다. 논문에서 제안된 한글 텍스트 감정분류모델은 데이터셋의 크기를 높일수록 선형적으로 F1점수의 성능이 높아지는 것을 확인할 수 있으며, 특히 가장 많은 데이터인 5배로 학습된 모델이 가장 높은 성능을 나타낸 것을 확인할 수 있다.

추가로 모델의 성능과 데이터셋의 상관관을 분석하기 위해 Table 5와 같이 학습데이터 크기 25,252, 테스트데이터 크기 7,222인 신문기사 댓글 데이터셋을 사용하여 모델을 학습시킨 후, 성능을 분석하였다.

· 긍정리뷰 10개에 대한 예측 결과

리뷰: [너무 감동적인 영화였어요. 눈물이 멈추질 않았네요.] => 긍정적인 리뷰입니다.
 리뷰: [배우들의 연기가 정말 뛰어났고, 스토리도 흥미로웠어요!] => 긍정적인 리뷰입니다.
 리뷰: [이런 영화를 기대했었어요. 완전 만족스러운 결말이었어요.] => 긍정적인 리뷰입니다.
 리뷰: [시간 가는 줄 모르고 봤어요. 너무 재밌었어요! 음악과 영상이 조화로워서 정말 몰입감 있게 봤어요.] => 긍정적인 리뷰입니다.
 리뷰: [음악과 영상이 조화로워서 정말 몰입감 있게 봤어요.] => 긍정적인 리뷰입니다.
 리뷰: [감정에 넘치는 연기로 이야기에 더욱 감동받았어요.] => 긍정적인 리뷰입니다.
 리뷰: [캐릭터들이 너무 매력적이었고, 웃음과 감동을 동시에 선사했어요.] => 긍정적인 리뷰입니다.
 리뷰: [이런 퀄리티의 영화를 꼭 극장에서 봐야하는 거 같아요.] => 긍정적인 리뷰입니다.
 리뷰: [매사치 못한 전개와 깊은 메시지가 내 마음을 완전히 사로잡았어요.] => 긍정적인 리뷰입니다.
 리뷰: [감독의 섬세한 감각이 돋보이는 작품이었습니다.] => 긍정적인 리뷰입니다.

· 부정리뷰 10개에 대한 예측 결과

리뷰: [기대한 만큼 실망스러웠어요. 스토리가 너무 어설프게 흘러갔어요.] => 부정적인 리뷰입니다.
 리뷰: [배경 음악이 너무 시끄러워서 대사를 제대로 알아들을 수가 없었어요.] => 부정적인 리뷰입니다.
 리뷰: [관객을 고려하지 않은 지루한 전개로 지루함이 연속했어요.] => 부정적인 리뷰입니다.
 리뷰: [배우들의 연기가 여색해서 어이없게 참았어요. 몰입감이 떨어졌어요.] => 부정적인 리뷰입니다.
 리뷰: [예고편에서 기대했던 분위기와는 완전히 달라서 실망했어요.] => 부정적인 리뷰입니다.
 리뷰: [중간에 포기하고 싶은 정도로 지루한 스토리였어요.] => 부정적인 리뷰입니다.
 리뷰: [너무 뻔한 클리셰를 사용한 느낌이라 실망스러웠어요.] => 부정적인 리뷰입니다.
 리뷰: [캐릭터들의 행동이 무리하게 느껴져서 이해하기 힘들었어요.] => 긍정적인 리뷰입니다.
 리뷰: [영화의 분위기와 음악이 조화롭지 않아서 영화에 몰입하기 어려웠어요.] => 긍정적인 리뷰입니다.
 리뷰: [영화의 논리적 오류가 너무 많아서 전체적으로 실망하게 했어요.] => 부정적인 리뷰입니다.

Fig. 14 Predicted results of the model(F1 score = 0.81)

• 긍정리뷰 10개에 대한 예측 결과(데이터셋 5배)

리뷰: [너무 감동적인 영화였어요. 눈물이 멈추질 않았네요.] => 긍정적인 리뷰입니다.
 리뷰: [배우들의 연기가 정말 뛰어났고, 스토리도 흥미로웠어요.] => 긍정적인 리뷰입니다.
 리뷰: [이런 영화를 기대했었어요. 완전 만족스러운 결말이었어요.] => 긍정적인 리뷰입니다.
 리뷰: [시간 가는 줄 모르고 봤어요. 너무 재밌었어요! 음악과 영상미가 조화로워서 정말 몰입감 있게 봤어요.] => 긍정적인 리뷰입니다.
 리뷰: [음악과 영상미가 조화로워서 정말 몰입감 있게 봤어요.] => 긍정적인 리뷰입니다.
 리뷰: [감정이 넘치는 연기로 이야기에 더욱 감동받았어요.] => 긍정적인 리뷰입니다.
 리뷰: [캐릭터들이 너무 매력적이었고, 웃음과 감동을 동시에 선사했어요.] => 긍정적인 리뷰입니다.
 리뷰: [이런 퀄리티의 영화를 꼭 극장에서 보아야 하는 거 같아요.] => 긍정적인 리뷰입니다.
 리뷰: [예상치 못한 전개와 깊은 메시지가 내 마음을 완전히 사로잡았어요.] => 긍정적인 리뷰입니다.
 리뷰: [감독의 섬세한 감각이 돋보이는 작품이었습니다.] => 긍정적인 리뷰입니다.

• 부정리뷰 10개에 대한 예측 결과 (데이터셋 5배)

리뷰: [기대한 만큼 실망스러웠어요. 스토리가 너무 어설픔게 흘러갔어요.] => 부정적인 리뷰입니다.
 리뷰: [배경 음악이 너무 시끄러워서 대사를 제대로 알아들을 수가 없었어요.] => 부정적인 리뷰입니다.
 리뷰: [관객을 고려하지 않은 지루한 전개로 지루함이 연속했어요.] => 부정적인 리뷰입니다.
 리뷰: [배우들의 연기가 어색해서 이입되기 힘들었어요. 몰입감이 떨어졌어요.] => 부정적인 리뷰입니다.
 리뷰: [메고면에서 기대했던 분위기와는 완전히 달라서 실망했어요.] => 부정적인 리뷰입니다.
 리뷰: [도중에 포기하고 싶을 정도로 지루한 스토리였어요.] => 부정적인 리뷰입니다.
 리뷰: [너무 뻔한 클리셰를 사용한 느낌이라 실망스러웠어요.] => 부정적인 리뷰입니다.
 리뷰: [캐릭터들의 행동이 무리하게 느껴져서 이해하기 힘들었어요.] => 부정적인 리뷰입니다.
 리뷰: [영화의 분위기와 음악이 조화롭지 않아서 영화에 몰입하기 어려웠어요.] => 부정적인 리뷰입니다.
 리뷰: [영화의 논리적 오류가 너무 많아서 전체적으로 실망하게 됐어요.] => 부정적인 리뷰입니다.

Fig. 15 Predicted results of the model(F1 score = 0.86)

Table 5 Dataset of news article comments

document	label
나이 들어서 걸리면 치사 올드 늙을건데 전부 자기 들 자초 일이 의료 진 죽어나는거지	0
축하 합니다	1
저 것 들 한 곳 몰아 넣고 지 둘 끼리 감염 되서 울 뿐 고함 다 디지길	0
몇 일 전 부터 우한 중국 입국 시키더니 우한 폐렴 확 진자 급증 이 거 아무래도 ...	0
왜 또 교회 몰려서 이런 난리 피우는 건지 집단 장소 가지말라고 그렇게 당부 메시지...	0
주님 만나러 일찍 가니 좋겠네요	1
데려 리스트 들	0
저승 길 친구 들 많이 있어 좋겠다 적당히 좀해	0
미친 개 제 일 교회 확 진자 다 가라 하눌나라	0
빨리 죽 병상 확보 하자	0

표에서 알 수 있듯이 신문기사 댓글 데이터셋은 긍정 댓글(=1)보다 부정 댓글(=0)이 더 많다

는 것을 알 수 있다. 학습데이터셋의 크기를 10,000건, 테스트 데이터셋의 크기로 5,000건으로 조정한 후, 5회 학습시킨 결과, 정확도 0.9656, 손실 0.0929, 부정에 대한 F1 점수 0.95, 긍정에 대한 F1점수 0.31의 성능을 보이는 모델을 얻었다. 결과에서 긍정과 부정에 대한 F1 점수가 차이가 나는 것은 신문기사 댓글 데이터셋 자체에 부정 댓글이 많은 이유다.

ChatCPT로 생성된 영화 감상 댓글(긍정 10개, 부정 10개) 데이터와 F1 점수가 0.81인 모델을 사용하여 한글 텍스트 감정을 예측한 결과를 Fig. 14에서 볼 수 있다. 예측결과로부터, F1 점

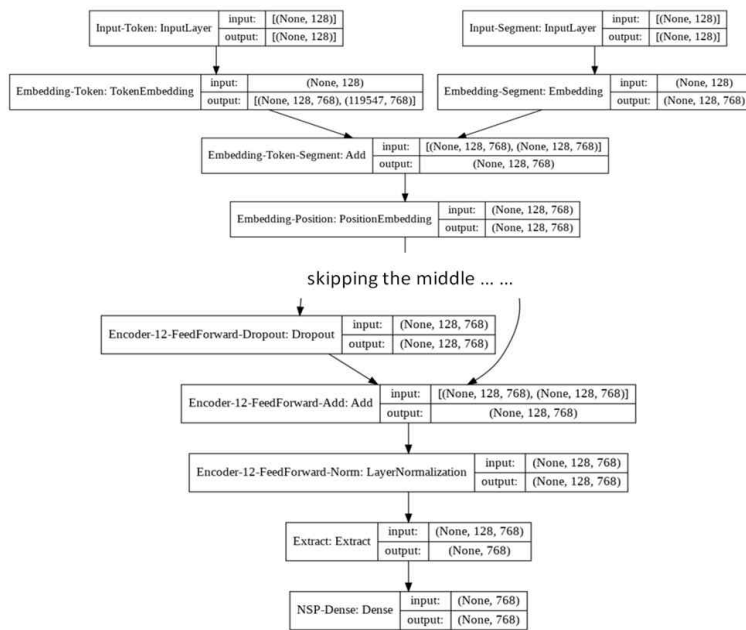


Fig. 16 Tensor flow diagram of the re-fine-tuned model for a 3D graph visualization

수 0.81인 모델은 부정 리뷰 예측에서 2개의 문장을 틀리게 예측한 것을 볼 수 있다.

앞의 모델이 비해 5배로 학습데이터량을 늘린 F1 점수 0.86 모델의 예측결과는 Fig. 15에서 볼 수 있다.

앞의 데이터와 같은 데이터를 사용하여 예측한 수행한 결과 긍정 텍스트 10개, 부정 텍스트 10개 모두를 정확하게 분류한 것을 볼 수 있다. F1 점수 0.86 이상의 모델이 영화감상댓글을 거의 100%에 가까운 정확도로 분류할 수 있다는 것을 유추할 수 있다.

주성분분석(principal component analysis)과 TSNE(t-distributed stochastic neighbor embedding) 클러스터링 알고리즘을 사용하면, 테스트 데이터셋으로 예측된 결과의 분포를 3차원 그래프로 시각화할 수 있다.

- 주성분분석(principal component analysis): 전체 데이터 분포를 서로 직교하는(orthogonal) 선형 벡터(linear vector)의 합으로 표현
- TSNE(t-distributed stochastic neighbor embedding) 클러스터링: 차원을 축소하여, 유사한 것끼리 클러스터를 생성하는 알고리즘

주성분분석과 TSNE 알고리즘을 적용하기 위해서는 outputs를 담당하는 구조에서 마지막 2개의 레이어를 삭제하여 새로운 모델을 생성한다. 새로 미세 조정된 모델 구조는 Fig. 16과 같다.

네이버 영화 댓글 데이터셋을 사용하여, BERT 문장 감정 이진 분류 모델 2개를 각각 생성한 후, 각 모델에 대해서, 예측된 데이터로 주성분분석과 TSNE 클러스터링 알고리즘을 수행한 후, 긍정(1), 부정(0) 부분의 클러스터를 확인한 결과는 Fig. 17과 같다.

상좌회전을 180°씩 회전하면서, 출력한 전체 그래프에서, 긍정(=1)을 나타내는 파란색 점들과 부정(=0)을 나타내는 빨간색 점들이 서로 섞여 있지 않고 구분되어 클러스터를 생성한 것을 확인할 수 있다.

5. 결론

본 논문에서는 대규모 언어모델로 개발이 가능한 사전 학습된 BERT-Base 다국어 문장 생성 모델을 사용하여 한글 문장 감정 이진 분류를 위한 모델을 생성하는 방법을 보였다. 사전 학습되어 제공되는 BERT-Base 모델을 텍스트 감정 분류를 위한 모델의 구조로 미세 조정했다. 178M개의 파라미터를 가지도록 미세 조정된 모델에 10,000건의 학습데이터셋과 5,000건의 테스트 데이터셋으로 5회 학습시킨 결과, 정확도 0.9582, 손실 0.1177, F1 점수 0.81인 한글 문장 감정 이진 분류 모델을 얻었다. 학습된 모델의 과적합 여부를 판단하기 위해, 학습데이터셋의 크기를 2~5배로 늘려서 새로운 모델들을 전이 학습시킨 후, 성능을 비교분석하였다. 데이터

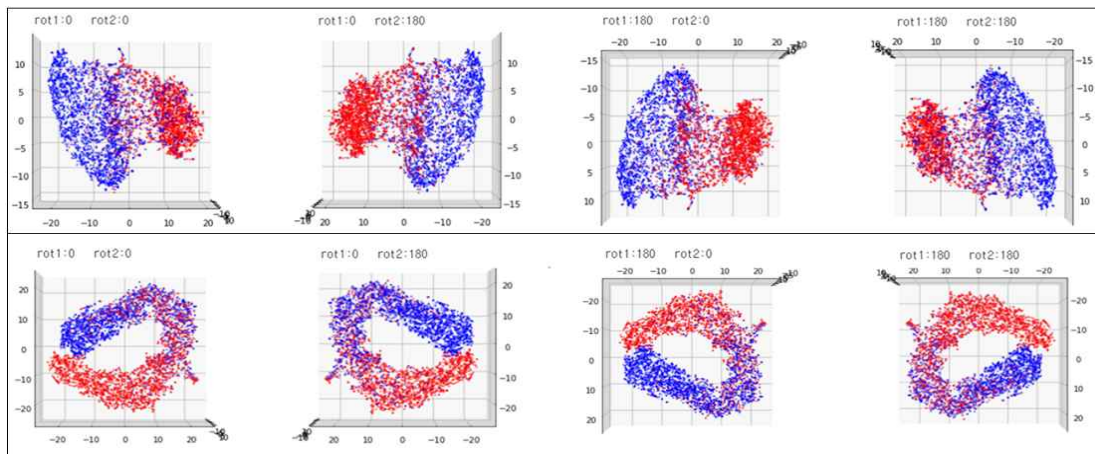


Fig. 17 Distribution of prediction result from two text sentiment classification models

셋을 5배 늘려 학습시킨 모델은 정확도 0.9562, 손실 0.1202, F1 점수 0.86의 성능을 보였다. 생성된 모델의 예측 성능을 파악하기 위해, ChatCPT로 생성된 각각의 긍정과 부정 댓글 10개에 대하여, F1 점수 0.86인 모델로 예측한 결과, 100%의 정확도를 보였다. 주성분분석과 TSNE 클러스터링 알고리즘을 사용하여 테스트 데이터셋으로 예측된 결과의 분포를 3차원 그래프로 시각화한 결과, 긍정(=1)과 부정(=0)을 나타내는 점들이 서로 섞여 있지 않고 구분되어 클러스터를 생성한 것을 확인할 수 있었다.

현재 거대언어모델(large language model) 분야에서, 추론을 위한 전용 AI 프로세서의 성능을 개선하기 위한 연구가 지속해서 진행되고 있다. 텍스트 감정 이진 분류를 위해, 전용으로 만들 수 있는 기본모델이나 RNN 기반 모델과 비교하여, 트랜스포머 구조를 바탕으로 한 BERT가 많은 레이어와 파라미터의 사용함으로써, 다소 과적합이 발생할 수도 있지만, 본 논문에서 제안된 방법을 사용하면 우수한 연구기관에서 사전 학습된 모델로 공개해 주는 모델을 사용하여, 한국어를 바탕으로 하는 모델을 신속하게 만들어서 테스트해 볼 수 있다는 장점이 있다. 그리고 인간피드백을 통한 강화학습과 같은 알고리즘들을 추가로 적용하면, 한국어에 특화된 ChatGPT와 같이 상호대화가 가능한 인간 친화적인 생성형 AI로 발전시켜 나갈 수 있다는 장점이 있다.

References

- O. Dongsuk, P. Sungjin, L. Hanna, J. Yoonna, and L. Heuseok, (2021), KoDialoGPT2 : Modeling Chit-Chat Dialog in Korean, *Proceedings of the 33th Korean Language and Korean Information Processing Conference*, Oct. 14 - 15, 457-460, Korea
- K. EunJung. (2022), A study on the difficulty adjustment of programming language multiple-choice problems using machine learning, *Journal of Korea Industrial Information Systems Research*, 27(2), 11-24
- K. SeongAn, K. SoHui and R. Min Ho. (2022), Analysis of Hypertension Risk Factors by Life Cycle Based on Machine Learning, *Journal of Korea Industrial Information Systems Research*, 27(5), 73-82
- L. DoegGyu, Kyungkeun B, L. HyungDong, and S. Sunhee. (2023), The Prediction of Survival of Breast Cancer Patients Based on Machine Learning Using Health Insurance Claim Data, *Journal of Korea Industrial Information Systems Research*, 28(2), 1-9
- S. John, W. Filip, D. Prafulla, R. Alec and K. Oleg, (2017), *Proximal Policy Optimization Algorithms*, Aug. 28, <https://doi.org/10.48550/arXiv.1707.06347>
- B. Tom B., M. Benjamin, R. Nick, et al., (2020), Language Models are Few-Shot Learners. *NIPS'20: Proceedings of the 34th International Conference on Neural Information Processing*, Jun. 22, 1877 - 1901, <https://doi.org/10.48550/arXiv.2005.14165>
- L. Ouyang, W. Jeff, X. Jiang, A. Diogo, et al., (2022), Training language models to follow instructions with human feedback, *Journal of Advances in Neural Information Processing Systems*, 35, 27730-27744, <https://doi.org/10.48550/arXiv.2203.02155>
- B. Sébastien, C. Varun, E. Ronen, et al., (2023), *Sparks of Artificial General Intelligence: Early experiments with GPT-4*, Apr. 13, <https://doi.org/10.48550/arXiv.2303.12712>
- D. Jacob, C. Ming-Wei, L. Kenton and T. Kristina, (2019), BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding, *2019 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, May. 24, <https://doi.org/10.48550/arXiv.1810.04805>
- TensorFlow Authors, (2019), *Basic text classification / TensorFlow Core*, <https://www>.

tensorflow.org/tutorials/keras/text_classification?hl=ko(Accessed on Oct. 3th, 2023)

Gooble Colab, (2023), *Text classification with an RNN*, https://www.tensorflow.org/text/tutorials/text_classification_rnn?hl=en(Accessed on Oct. 3th, 2023)

TensorFlow Hub Authors, (2020), *Classify text with BERT*, https://github.com/tensorflow/text/blob/master/docs/tutorials/classify_text_with_bert.ipynb(Accessed on Oct. 3th, 2023)

K. Jared, M. Sam, H. Tom, B. Tom B, et al., (2020), *Scaling Laws for Neural Language Models*, OpenAI, Jan. 23, <https://doi.org/10.48550/arXiv.2001.08361>

V. Ashish, S. Noam, P. Niki, U. Jakob, J. Llion, G. Aidan N., K. Lukasz and P. Illia, (2017), Attention Is All You Need, *The Thirty-first Annual Conference on Neural Information Processing Systems*, Dec. 6, <https://doi.org/10.48550/arXiv.1706.03762>

W. Yizhong, K. Yeganeh, M. Swaroop, et al., (2022), SELF-INSTRUCT: *Aligning Language Model with Self Generated Instructions*, Dec. 20, <https://doi.org/10.48550/arXiv.2212.10560>

Team with members from UC Berkeley, CMU, Stanford, and UC San Diego, (2023). *Vicuna: An Open-Source Chatbot Impressing GPT-4 with 90%* ChatGPT Quality*, <https://vicuna.lmsys.org/>(Accessed on Jun. 25th, 2023)

S. Chang-Uk, (2020), *Awesome Korean NLP Papers*, <https://github.com/changukshin/Awesome-Korean-NLP-Papers>(Accessed on Oct. 4th, 2023)

L. Eunchan, L. Changhyeon and A. Sangtae, (2022), Comparative Study of Multiclass Text Classification in Research Proposals Using Pretrained Language Models, *applied sciences*, <https://www.mdpi.com/2076-3417/12/9/4522>(Accessed on Oct. 4th, 2023)



김 종 수 (JongSoo Kim)

- 정회원
- 부경대학교 냉동공학과 공학사
- 부산외국어대학교 전자·컴퓨터학과 공학석사
- 동의대학교 컴퓨터소프트웨어공학과 공학박사
- (현재) ㈜골드브릿지, 기업부설연구소, 소장
- 관심분야: 소프트웨어설계, 인공지능, 클라우드