변 혜 원 교수 지도
석사학위 청구논문

앙상블 사전학습 모델 기반의 SNS 문맥인지 이모티콘 추천

2023

성신여자대학교 대학원
미래융합기술공학과
김 지 현

앙상블 사전학습 모델 기반의 SNS 문맥인지 이모티콘 추천

변 혜 원 교수 지도

이 논문을 석사 학위 논문으로 제출함

2023년 05월

성신여자대학교 대학원

미래융합기술공학과

김 지 현

인 준 서

김지현의 석사학위 논문으로 인준함

2023년 05월

심사위원장 오 장 민 (서명 또는 인)

심 사 위 원 유 재 현 (서명 또는 인)

심 사 위 원 변 혜 원 (서명 또는 인)

성신여자대학교 대학원

논 문 개 요

이모티콘 추천은 수천 개의 이모티콘 중에서 사용자가 원하는 적절한 이모티콘을 용이하게 찾도록 도와주는 중요한 태스크이다. 기존의 이모티콘 추천 방법들은 채팅 플랫폼을 대상으로 하며 사용자들이 많이 사용하는 감정 이모티콘 위주로 추천한다. 그러나 인스타그램 등 SNS 플랫폼에서는 감정 전달보다는 업로드한 짧은 게시글의 내용을 보완하거나 강조하는 용도로 이모티콘을 사용하는 경향이 있다.

이 연구에서는 SNS 플랫폼에서 한국어 게시글의 문맥을 파악하여 이모티콘을 추천하는 새로운 방법론을 제안한다.

이모티콘 추천 문제에 계층적 KoBERT를 도입하여 한국어 게시글의 문맥을 파악하고 이에 적합한 다양한 이모티콘을 추천한다. 314개 이모티콘 카테고리에 속하는 616개의 이모티콘 추천은 SNS 게시글의 함축적인 단문을 보다 정확하게 전달하는 데 유용하다.

인스타그램 게시글을 수집하여 실제 세계를 반영하는 데이터셋을 구성하고 각 텍스트에 삽입되어 있는 이모티콘의 계층적 카테고리를 학습하기 위해 계층적 KoBERT 모델을 구축한다. 실험 결과에서 DNN, LSTM, Bi-LSTM, GRU 모델과 비교하여 계층적 KoBERT 모델이 이모티콘 추천에서 높은 성능을 보이는 것을 검증하였다.

또한, 성능 향상을 위해 KoBART, KoGPT2, KoELECTRA, KcELECTRA 등의 한국어 전이학습 모델을 추가로 도입하여 모델 성능을 비교 분석하였고 스태킹 앙상블 기법을 적용하였다.

목 차

그 림 목 차

표 목 차

Ⅰ. 서 론

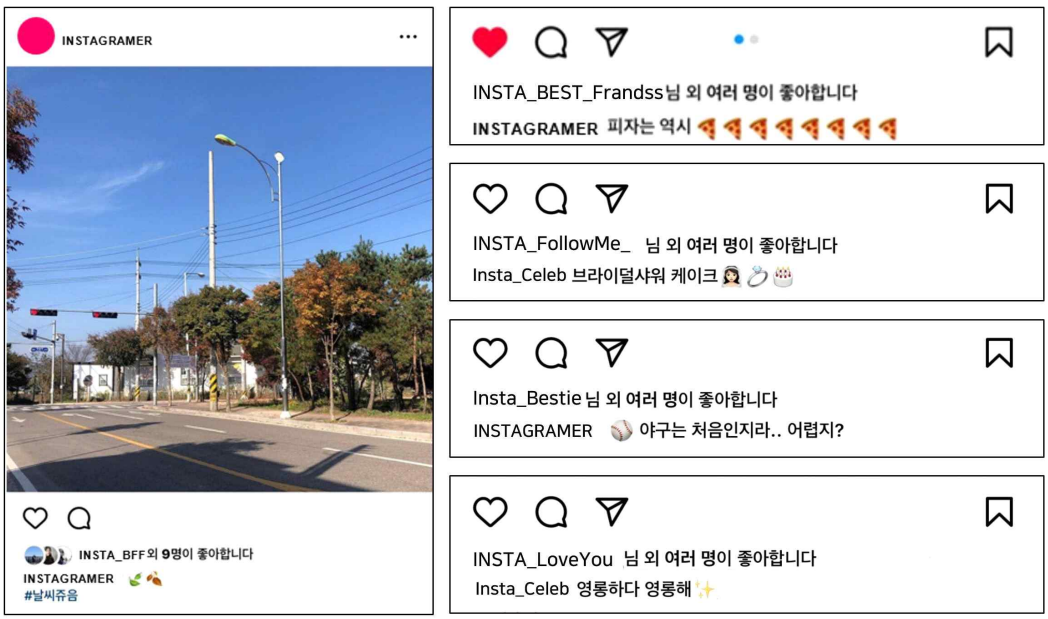
1. 연구배경

모바일 기기의 발전으로 시간과 장소의 제약 없이 자신의 생각과 일상을 공유할 수 있는 SNS의 사용량이 지속적으로 증가하고 있다. SNS는 마이크로 블로그 형태의 서비스로서 단문과 몇 개의 이미지들을 사용하여 짧은 내용의 콘텐츠를 구성하여 업로드하는 형태이다. SNS의 대표적인 예시로는 인스타그램, 트위터 등이 존재한다. 카카오톡 등 채팅 서비스에서 주로 사용되어 온 이모티콘은 SNS 플랫폼에서도 활발하게 사용되고 있다[1]. 2022년 9월에 출시된 유니코드 이모티콘 버전 15.0에서는 4,526개의 이모티콘을 제공하고 있다. 수 천개의 이모티콘은 표현의 다양성을 제공하는 장점이 있는 반면에, 텍스트에 적합한 이모티콘을 검색하고 선택하는 데 오랜 시간이 걸린다는 불편함이 있다[2].

SNS 플랫폼에서 이모티콘을 사용하는 방식은 채팅 플랫폼에서 이모티콘을 사용하는 양상과는 차이를 보이고 있다. 채팅에서는 주로 텍스트를 전달하는 사람의 희로애락 등의 감정을 표현하고자 이모티콘을 사용하는 반면에, 인스타그램 등의 SNS에서는 게시글의 내용을 보완 또는 강조하거나 시각적인 효과를 부가적으로 표현하고자 하는 데 그 보편적 사용 의도가 있다[3][4]. 예를 들어, [그림 1-1]을 보면, 피자, 케이크, 야구 등의 단어를 시각

1) Choi, Ji-Eun, "The Influence of Motivation and Usage Patterns of Emoticons on Social Capital Formation in SNS," Management Studies, Vol.32, No.3, pp.1-20, 2017.
2) Henning Pohl, Dennis Stanke, and Michael Rohs, "EmojiZoom: emoji entry via large overview maps," Association for Computing Machinery, pp.510 - 517, 2016.
3) Eun Ji Lee, "Motivations for the Using Emoticon : Exploring the Effect of Motivations and Intimacies between Users on the Attitude and Behaviors of Using Emoticon," *Journal of the HCI Society of Korea*, Vol. 12, No. 2, pp. 5-12, 2017.

적으로 강조하기 위하여 피자 모양, 케이크 모양의 이모티콘을 사용하고 있으며, '영롱하다'의 느낌을 보다 정확하게 전달하기 위하여 이를 시각화한 이모티콘을 사용하고 있다. 반면에, 채팅 서비스에서는 피자나 케이크 이모티콘 보다는 음식을 먹는 행복한 감정을 담은 표정 이모티콘을 전달하는 경향이 있다.



【그림 1-1】Example of emoticon usage on Instagram

기존의 연구들은 주로 채팅 상황에서 감정 기반의 이모티콘 추천에 집중되어 있다. 이런 방식은 감정을 기반으로 텍스트를 분류하고 30~100 개 정도의 이모티콘을 추천하므로 SNS 게시글의 다양한 내용과 문맥을 표현하는 데에는 한계가 있다[5][6][7][8][9]. 따라서, 본 논문에서는 SNS 플랫폼 중 하나인

4) Young Il Hong, Sun Kyung Yim, "The Effect of Emoticon Expression Type on User Satisfaction Factors," *The Treatise on The Plastic Media,* Vol. 25, No.1, pp. 33-41, 2022.
5) V. N. Durga, Pavithra Kollipara, V. N. Hemanth Kollipara, and M. Prakash, "Emoji Prediction from Twitter Data using Deep Learning Approach," *Asian Conference on Innovation in Technology (ASIANCON)*, 2021.
6) Xuanzhi Zheng, Guoshuai Zhao,Li Zhu, Xueming Qian, "PERD: Personalized Emoji

인스타그램에서 사용자가 작성한 게시글의 문맥을 파악하여 문맥 기반으로 이모티콘을 추천하는 새로운 시스템을 제안하고자 한다.

Recommendation with Dynamic User Preference," *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval,* pp. 1922 - 1926, July 2022.

7) Seongmin Lee, Eunseo Lee, and Daeyoung Park, "Emoticon Recommendation using Emotional Analysis," *Proceedings of Symposium of the Korean Institute of communications and Information Sciences*, pp. 864-865. 2021.

8) Luda hao and Connie Zeng, "Using Neural Networks to Predict Emoji Usage from Twitter Data," *Computer Science*, 2017.

9) Kazuyuki Matsumoto, Minoru Yoshida, and K. Kita, "Classification of Emoji Categories from Tweet Based on Deep Neural Networks," *Proceedings of the 2nd International Conference on Natural Language Processing and Information Retrieval(NLPIR),* pp 17-25, Sep. 2018.

2. 연구목적

본 논문의 주된 목표는, SNS 플랫폼에서 사용자가 작성한 게시글의 문맥을 파악하여 이모티콘을 추천하는 것이다. 이를 위해, 인스타그램 SNS 플랫폼에서 문맥 기반으로 이모티콘을 추천하는 새로운 시스템을 제안한다.

자연어 처리 분야에서 성능이 뛰어난 사전학습 모델 BERT를 도입하여 문맥 인지 이모티콘 추천 문제에 적용한다. 변화의 범위가 매우 큰 사용자 게시글의 문맥에 적합한 이모티콘을 다양하게 추천하기 위하여 이모티콘 추천 문제를 314개의 멀티 클래스를 가지는 한국어 텍스트 분류 문제로 정의하고 성능 향상을 극대화하기 위하여 계층적 KoBERT 모델을 구축한다.

616개의 이모티콘 데이터에 계층적 군집화를 적용하여 메인 카테고리와 서브 카테고리로 분류하고 텍스트 데이터의 라벨로 사용함으로써 대량의 텍스트 데이터를 자동 라벨링하는 장점을 가진다. 또한, DNN, LSTM, Bi-LSTM, GRU 등의 모델들과 성능을 비교 분석하여 본 연구에서 구축한 계층적 KoBERT 모델의 성능이 우수함을 보여준다.

더 나아가, 성능 향상을 위해 KoBART, KoGPT2, KoELECTRA, KcELECTRA와 같은 다양한 한국어 전이학습 모델을 도입하여 성능 비교 실험을 수행하였다. 또한, 학습 된 한국어 전이학습 모델을 이용해 스태킹 앙상블 방법을 적용하였다. 서로 다른 메인모델들의 예측 결과를 합쳐 학습 데이터로 활용함으로써, 계층적 모델의 성능 향상을 이루었다.

3. 논문 기여 및 구성

이 연구에서 기여하는 점은 다음과 같다.

첫째, 이전의 연구들은 채팅창에서의 감정 기반 이모티콘 추천에 집중한 반면 본 연구에서는 SNS 플랫폼에서의 문맥 인지 이모티콘 추천 방법을 새롭게 제안한다.

둘째, 기존 연구들은 사용자들이 주로 사용하는 30~100개 정도의 이모티콘을 추천하고 있으나, 본 연구에서는 314여 개의 다양한 이모티콘 카테고리를 추천하여 문맥 인지 이모티콘 추천을 시도한다.

셋째, 이모티콘 계층적 군집화를 통해 학습용 대규모 텍스트 데이터를 자동 라벨링하는 방법을 제시한다.

넷째, 계층적 KoBERT 기반으로 분류기를 생성하여 314개의 멀티 클래스로 SNS 게시글을 분류하는 문제를 해결하고 이모티콘 추천 성능을 향상시켰다.

다섯째, 계층적 KoBERT 모델의 우수성을 입증하기 위해 DNN, LSTM, GRU, Bi-LSTM과의 비교 분석을 진행하였다.

여섯째, 이모티콘 추천 성능 향상을 위해 KoBART, KoGPT2, KoELECTRA, KcELECTRA등의 한국어 전이학습 모델을 추가 도입하였으며 앙상블 모델을 구축하였다.

본 논문은 2장에서 관련 연구를 소개하고, 3장에서는 시스템 구조를 설명한다. 4장에서는 데이터 구성에 대해, 그리고 5장에서는 모델 학습 방법에 대해 상세히 다룬다. 6장에서는 실험 결과와 그 분석을 제시하고, 마지막으로 7장에서 연구의 결론을 서술한다.

Ⅱ. 관련 연구

1. 이모티콘 추천 시스템

최근에 이모티콘의 사용행태를 분석하고 이모티콘을 추천하는 연구들이 몇 년간 진행되어 왔다. 국내에서 한국어 데이터셋에 대한 이모티콘 추천은 주로 카카오 채팅 플래폼에서 텍스트의 감정을 분석하여 해당 감정 카테고리에 속하는 이모티콘을 추천하는 방식이다. 이성민(2021)[7] 연구는 AI Hub에서 제공하는 '한국어 감정 정보가 포함된 단발성 대화 데이터셋'을 활용하였다. 연구에서는 LSTM을 사용하여 데이터셋에서 공포, 놀람, 분노, 슬픔, 행복, 혐오, 중립의 7가지 감정 유형을 추출하였고, 이를 바탕으로 적절한 이모티콘을 추천하는 시스템을 제안하였다.

영어 데이터셋에 대한 이모티콘 추천에 관한 연구는 주로 트위터 데이터셋을 사용하고 딥러닝 모델을 도입하여 텍스트를 분류하는 연구가 진행되었다. 대부분의 연구들에서 CNN과 LSTM 계열의 학습 네트워크를 도입하여 텍스트를 분류하고 적합한 이모티콘을 추천한다. 예를 들어, Luda hao.(2017)[8]의 연구는 트위터 트윗을 활용하여 텍스트를 가장 가능성 있는 이모티콘으로 매핑하는 다중 클래스 분류 문제를 설정한다. 이 과정에서 LSTM-RNN 및 CNN 모델을 이용하여 성능을 비교하였다. Kazuyuki Matsumoto.(2018)[9] 연구에서는 트위터의 감정 분석을 위해 이모티콘 카테고리를 감정 레이블로 활용하는 방식을 제안한다. 이를 위해, CNN, BiLSTM, BiGRU 등의 기계 학습 알고리즘을 활용하였다. V. N. Durga.(2021)[5] 연구와 J Shobana.(2020)[10]연구는 텍스트와 이모티콘 간의

관계를 분석하여 텍스트에 가장 적합한 이모티콘을 예측하는 방법을 제안한다.

채팅 플랫폼에서의 이모티콘 추천은 다자간의 주고받은 대화문 히스토리를 대상으로 문맥을 파악하는 방향으로 발전되었다. 이를 위해, LSTM 모델이 주로 활용되었다. Ruobing Xie.(2016)[11]의 연구는 답장 문장만을 고려하는 Single-LSTM, 전체 대화를 순차적으로 분석하는 Flatten-LSTM, 그리고 계층적으로 대화를 처리하는 Hierarchical-LSTM 등 세 가지 다른 접근법을 통해 이모티콘을 추천하는 방식을 제안하였다. 김준겸(2020)[12] 연구에서는 작성 중인 답장과 그에 앞선 최대 5개의 대화를 LSTM을 활용하여 벡터화하는 방법을 도입하였다. 이 과정에서 이모티콘 또한 emoji2vec을 통해 벡터화하였으며, 이렇게 변환된 벡터를 기반으로 대화문 벡터와 가장 가까운 이모티콘을 순차적으로 추천하였다. 한편, Gaël Guibon.(2018)[13]의 연구는 이모티콘 추천에 있어 감정을 고려하기 위해 이모티콘 감정 사전인 EmojiSentimentRanking을 도입하여 감정 이모티콘만을 추출하였다. 더불어, 사용자 대화의 분석을 통해 38가지 감정 중 하나를 도출하였고, 이러한 감정 정보를 랜덤 포레스트 알고리즘에 적용하여 이모티콘 추천 시스템을 구현하였다.

대화문 분석을 위해서 사전학습 언어모델을 도입한 방법들도 시도되었다.

---

10) J Shobana, S Amudha, and S Kumar, “Emoji Anticipation and Prediction Using Deep Neural Network Model,” International Conference on Power, Energy, Control and Transmission Systems (ICPECTS), 2022.

11) Ruobing Xie, Zhiyuan Liu, Rui Yan and Maosong Sun, “Neural Emoji Recommendation in Dialogue Systems,” 2016.

12) Joon Gyum Kim, Tae Sik Gong, Bo Goan Kim, Jae Yeon Park, Woo Jeong Kim, Evey Huang, Kyung Sik Han, Ju Ho Kim, Jeong Gil Ko, and Sung Ju Lee, “No More One Liners: Bringing Context into Emoji Recommendations,” ACM Transactions on Social Computing, Vol. 3. pp. 1-25, 2020.

13) Gaël Guibon, Magalie Ochs, and Patrice Bellot, “Emoji Recommendation in Private Instant Messages,” Proceedings of the 33rd Annual ACM Symposium on Applied Computing(SAC), pp. 1821-1823. 2018.

B. Felbo.(2017)[14]의 연구에서는 사전학습 개념을 LSTM에 적용한 DeepMoji 모델을 도입하였다. 이 모델은 텍스트에서의 감정과 냉소적 표현을 탐지하여 64개의 주요 이모티콘을 효과적으로 추천할 수 있었다. Tomihira, T.(2020)[15] 연구는 트위터 데이터를 활용하여 사전학습된 자연어 처리 모델인 BERT를 이용해 이모티콘 예측 모델을 설계하였다. Xuanzhi Zheng.(2022)[6]연구는 BERT 모델을 활용하여 이용자의 트윗 히스토리를 분석하고 시간에 따른 이용자의 변화하는 선호도를 반영한 개인화된 이모티콘 추천 방법을 제안하였다.

이후의 이모티콘 추천 연구는 트위터 텍스트뿐만 아니라 시각적인 부가정보 등을 추가로 사용하는 멀티모달 접근법을 시도하였다. Peijun Zhao.(2018)[16]연구에서는 트위터에서 사용자의 텍스트, 업로드된 이미지, 그리고 위치 정보를 함께 학습하는 mmGRU 모델을 제안하였다. 이 모델은 단순히 이모티콘을 추천하는 것뿐만 아니라 텍스트 내에서 이모티콘의 위치를 함께 추천함으로써 더욱 고도화된 이모티콘 사용을 가능하게 하였다. Guoshuai Zhao.(2020)[17]의 연구는 이모티콘 추천에 있어 개인의 성향을 고려하는 새로운 접근법을 제안하였다. 이 연구에서는 텍스트의 문맥뿐만 아니라 사용자의 선호도, 성별, 그리고 텍스트 입력 시간 등 개인적 특성을 함께 고려하는 행렬 분해 기법(matrix factorization method)을 도입하였다.

---

14) B. Felbo, A. Mislove, A. Søgaard, I. Rahwan, and S. Lehmann, “Using Millions of Emoji Occurrences to Learn Any-Domain Representations for Detecting Sentiment, Emotion and Sarcasm,” in Proceedings of the Conference on Empirical Methods in Natural Language Processing, pp. 1615 - 1625, 2017.

15) Tomihira T., Otsuka A., Yamashita A., and Satoh T., “Multilingual Emoji Prediction using BERT for Sentiment Analysis,” International Journal of Web Information System, Vol. 16, No. 3, pp. 265-280, 2020.

16) Peijun Zhao, Jia Jia, Yongsheng An, Jie Liang, Lexing Xie, and Jiebo Luo, “Analyzing and Predicting Emoji Usages in Social Media,” WWW '18: Companion Proceedings of the The Web Conference, pp. 327-334, 2018.

17) Guoshuai Zhao, Zhidan Liu, Yulu Chao, and Xueming Qian, “CAPER: Context-Aware Personalized Emoji Recommendation,” IEEE Transactions on Knowledge and Data Engineering, pp. 1-1, 2020.

Kim. 등(2023)[18]의 연구에서는 SNS 플랫폼에서 문맥을 분석하기 위하여 KoBERT를 기반으로 이모티콘을 추천하는 연구를 진행하였다. 본 연구에서는 이를 확장하여 KoBART, KoELECTRA 등 한국어 사전학습 모델들과의 비교 분석을 수행하고 다양한 앙상블 모델을 제시한다.

2. 사전학습 언어 모델

사전학습 언어모델은 트랜스포머 아키텍처(Transformer Architecture)를 기반으로 대규모 데이터셋을 통해 미리 학습된 딥러닝 모델이다. Transformer[19]는 인코더-디코더 구조를 가지고 있다. 인코더는 입력 데이터를 고차원의 표현으로 변환하며, 디코더는 이 표현을 사용하여 출력 데이터를 생성한다. 각각의 인코더와 디코더는 다수의 셀프 어텐션 레이어(Self-Attention Layer)와 포지션 와이즈 피드 포워드(Position-wise Feed-Forward Networks) 네트워크로 구성되어 있어, 복잡한 패턴을 학습하고 다양한 자연어 처리 작업에 적용할 수 있는 유연성을 제공한다. 트랜스포머 기반의 사전학습 언어모델들은 다양한 자연어 처리 문제에 적용되어, 뛰어난 성능을 보여주고 있다. 이러한 사전학습 언어모델의 발전은 데이터 부족 문제를 해결하고 학습 시간을 단축하는데 크게 기여하였다. 이로 인해, 이들 모델은 다양한 연구와 실질적인 응용 분야에서 널리 활용되고 있다.

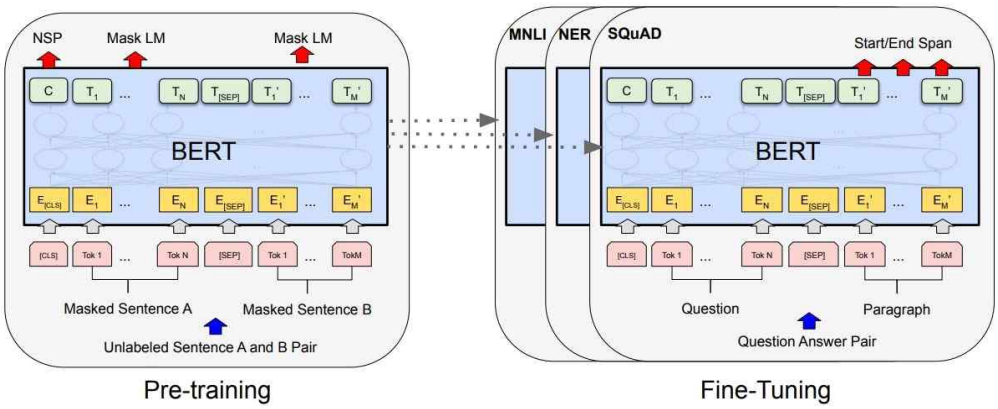
BERT(Bidirectional Encoder Representations from Transformers)

Google에서 2018년에 발표한 BERT[20]는 방대한 언어 데이터셋을 활용하

18) JeeHyun Kim, YeRim Kim, HaeWon Byun, “SNS Context-based Emoji Recommendation using Hierarchical KoBERT,” Journal of Digital Contents Society,24(6), 2023.6.
19) Vaswani, Ashish, et al, "Attention is all you need," Advances in neural information processing systems 30, 2017.

여 자기 지도 학습(Self-Supervised Learning)을 수행하는 대규모 사전 학습 언어 모델이다. 이 모델은 트랜스포머의 인코더 구조를 기반으로 하여 양방향 문맥을 고려하는 것이 주요 특징이다. 이를 통해 BERT는 텍스트의 전방과 후방 정보를 모두 활용하여 더욱 정밀한 문맥 이해를 가능케 한다.

BERT의 학습 과정은 크게 Pre-Training과 Fine-Tuning 두 가지 단계로 구성되며, 이 두 단계는 [그림 2-1]에 상세히 설명되어 있다. Pre-Training 단계에서는 Masked Language Modeling(MLM)과 Next Sentence Prediction(NSP)이라는 두 가지 전략을 활용하여 학습을 진행한다. MLM은 학습 과정에서 입력 데이터의 약 15%를 [MASK] 토큰으로 치환하는 방식을 사용한다. 이때, [MASK] 토큰은 양방향 문맥 정보를 활용하여 원본 토큰으로 예측되어야 한다. NSP은 두 문장이 연속성을 가지는지 아닌지를 판단하는 학습 과정을 포함한다.



【그림 2-1】 Pre-training and fine-tuning of BERT[17)]

이러한 Pre-Training을 거친 후, 미리 학습된 모델 파라미터를 바탕으로 Fine-Tuning 단계에서는 레이블이 지정된 데이터를 활용하여 특정 자연어

20) Devlin, Jacob, et al, "Bert: Pre-training of deep bidirectional transformers for language understanding," 2018.

처리 작업을 수행한다. 이런 학습 과정을 통해 BERT는 텍스트 분류(Text Classification), 개체명 인식(Named Entity Recognition), 감정 분석(Sentiment Analysis) 등과 같은 다양한 자연어 처리 작업에서 뛰어난 성능을 보여준다. BERT 모델은 여러 버전이 존재하며, BERT-Base는 12개의 레이어를, BERT-Large는 24개의 레이어를 사용한다. 모델의 크기가 증가함에 따라 처리할 수 있는 데이터의 양과 문제의 복잡성이 증가한다. 한국어에 적용된 BERT 모델로는 SKT-Brain의 KoBERT[21], ETRI의 KorBERT[22]등이 있다. 이러한 특성들로 인해, BERT는 최근에 자연어 처리 분야의 여러 연구에서 널리 활용되며 뛰어난 성과를 보여주고 있다.

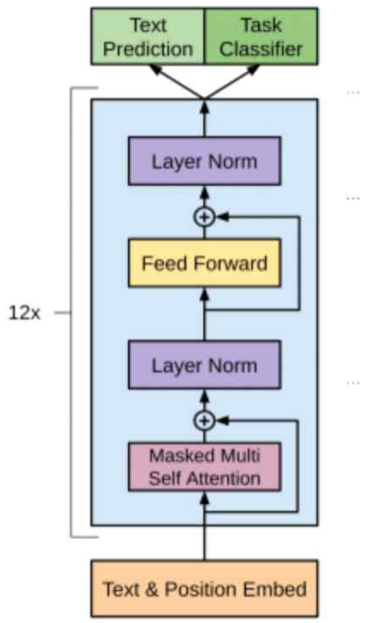
GPT2(Generative Pre-trained Transformer 2)

GPT-2[23]은 트랜스포머 구조를 활용하여 대규모 데이터셋을 학습한 사전 학습 언어모델로 2019년 OpenAI에서 공개하였다. 이 모델은 기존 GPT-1[24]의 한계를 극복하고자 개발되었는데, GPT-1이 비지도 학습에도 불구하고 특정 태스크에 적용할 때 성능 향상을 위해 Fine-Tuning과정과 Input Transformation이 필요했던 문제를 해결하기 위함이다. 이를 달성하기 위해, GPT-2는 Layer Normalization의 위치를 변경하는 등의 구조적 변형을 진행하였고, 다양한 도메인의 데이터를 활용하기 위해 Web Text를 추가적으로 고려하는 등의 데이터 변형을 수행하였다. 그러나 여전히 특정 태스크를 적용하기 위해서는 Fine-Tuning과정과 Input Transformation이 필요한 한계가 남아있다.

GPT-2는 [그림 2-2][25]에서 보이듯이 트랜스포머의 디코더 구조를 기반으

21) SKTBrain KoBERT, GitHub repository, https://github.com/SKTBrain/KoBERT, 2019.
22) AIOPEN ETRI, AI API·DATA, https://aiopen.etri.re.kr/bertModel, 2019.
23) Radford, Alec, et al, "Language models are unsupervised multitask learners," 2019.
24) Radford, Alec, et al, "Improving language understanding by generative pre-training," 2018.
25) Perez, Luis, Lizi Ottens, and Sudharshan Viswanathan, "Automatic Code Generation

로 단방향적으로 입력 데이터를 처리하며, 비지도 학습을 수행한다. 이 모델은 주어진 텍스트의 이전 단어를 활용하여 다음 단어를 예측하는 언어 모델링 작업에 중점을 두어 학습되었다. 이러한 학습방식은 GPT-2를 기계번역(Machine Translation), 대화시스템(Chatbot)과 같은 텍스트 생성 작업에 특화되게 하며, 자연어 이해 작업에도 우수한 성능을 발휘하게 한다. 레이어 수에 따라 다양한 버전의 모델이 존재하며, GPT-2는 12개의 레이어를 가지고 있고, GPT2-Medium은 24개, GPT2-Large는 36개, GPT2-XL는 48개의 레이어를 가진다. 이들 모델은 레이어가 증가함에 따라 더 많은 파라미터를 활용하여 복잡한 모델을 학습할 수 있는 특성이 있다. 한국어 GPT-2의 대표적인 예시로는 SKT-AI의 KoGPT-2[26]가 있다.



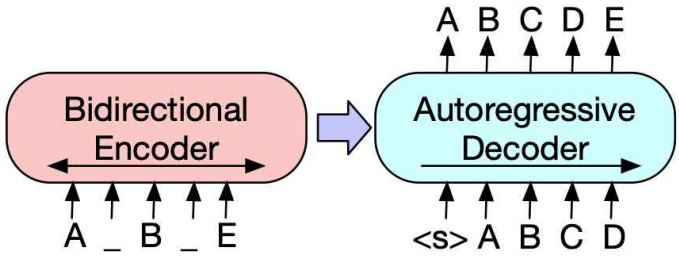
【그림 2-2】 GPT-2 Architecture[22]

using Pre-Trained Language Models," 2021.

26) SKT-AI KoGPT2, Github repository, https://github.com/SKT-AI/KoGPT2, 2020.

BART(Bidirectional Auto-Regressive Transformers)

Facebook에서 2020년에 발표한 BART[27]는 대규모 자연어 데이터를 자기 지도 방식으로 학습하는 트랜스포머 구조를 기반의 사전학습 언어 모델이다. 이 모델은 [그림 2-3]에서 보여주는 것처럼, BERT의 양방향 인코더와 GPT의 자기 회귀 디코더를 결합하여 구성되어 있다. 이러한 구성은 BART가 노이즈를 추가한 입력 데이터를 원본 데이터로 복구하는 denoising autoencoder 방식의 학습을 가능하게 한다.

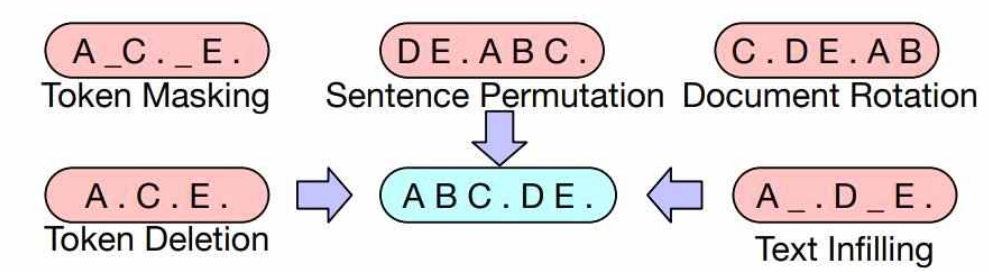


【그림 2-3】 BART Architecture[24]

BART의 학습 전략은 [그림 2-4]에 제시된 다양한 노이즈 추가 기법을 통해 구체화된다. Token Masking은 BERT와 유사하게 특정 토큰을 [MASK]로 대체하고 이를 복구하는 방식이다. Token Deletion은 일부 토큰을 삭제하고 이를 복구하는 방식이다. Text Infilling은 일정 범위의 토큰을 [MASK]로 대체하고 그 개수를 예측하는 방식이다. Sentence Permutation은 문장 단위로 나눈 후 임의로 순서를 변경하는 방식이다. 마지막으로, Document Rotation은 문서 내의 특정 토큰을 시작점으로 설정하여 문서의 시작과 끝을 구분하는 방식이다. 이런 학습 방법을 통해, BART는 텍스트 생성과 이해에 관련된 다양한 작업에서 뛰어난 성과를 보여준다. 특히 기계

27) Lewis, Mike, et al, "Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension," 2019.

번역, 감성 분석, 텍스트 요약(Text Summarization) 등의 분야에서 그 유용성이 입증되었다. BART는 두 가지 주요 버전, 즉, 6개의 인코더 레이어와 6개의 디코더 레이어를 사용한 BART-Base와, 12개의 인코더 레이어와 12개의 디코더 레이어를 사용한 BART-large를 제공한다. 이러한 구조는 레이어의 수가 많아짐에 따라 데이터가 적더라도 더 높은 성능을 달성할 수 있다는 장점을 제공한다. 한국어에 대한 BART 모델의 대표적인 예로는 SKT-AI에서 제공하는 KoBART[28]가 있다.



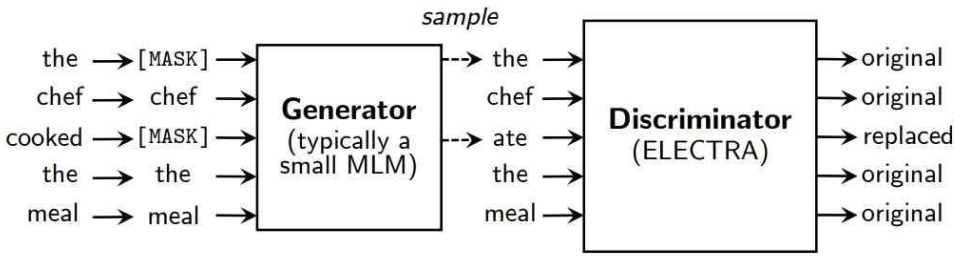
【그림 2-4】 Noise Injection Techniques[24]

ELECTRA(Efficiently Learning an Encoder that Classifies Token Replacements Accurately)

ELECTRA[29]는 2020년에 Google에서 발표된, 트랜스포머의 인코더 구조를 활용한 사전학습 언어 모델이다. 이 모델은 BERT의 Masked Language Modeling (MLM) 전략을 변형하여 자기 회귀 학습을 수행한다. MLM은 일부 입력 단어를 [MASK] 토큰으로 대체하고, 이를 양방향 문맥 정보를 활용하여 원본 토큰으로 예측하는 방식을 사용한다. 그러나 Fine-Tuning 과정에서는 [MASK] 토큰을 사용하지 않으므로 토큰 미스매치 문제가 발생한다. ELECTRA는 이 문제를 해결하기 위해 Replaced Token Detection

28) SKT-AI KoBART, Github repository, https://github.com/SKT-AI/KoBART, 2020.
29) Clark, Kevin, et al, "Electra: Pre-training text encoders as discriminators rather than generators," 2020.

(RTD) 방식을 제안한다. RTD는 입력 문장의 일부를 [MASK] 토큰이 아닌 가짜 토큰으로 대체하고, 이 토큰을 원본으로 복원하도록 학습한다. 이 과정은 Generator-Discriminator 구조를 통해 이루어지며 [그림 2-5]에서 보다 자세하게 확인할 수 있다. Generator는 원본 토큰을 가짜 토큰으로 대체하는 역할을 하며, Discriminator는 가짜 토큰을 식별하는 역할을 수행한다. 또한, BERT는 학습 시 전체 데이터의 15%만을 대상으로 변형을 수행하지만, ELECTRA는 모든 데이터에 대해 변형을 적용한다. 이런 방식은 BERT에 비해 학습 시간과 비용을 절감하면서도 높은 성능을 보여준다.



【그림 2-5】 Overview of Replaced Token Detection in the ELECTRA[26)]

ELECTRA는 레이어 수와 히든 사이즈에 따라 ELECTRA-Small (12 레이어, 히든 사이즈 256), ELECTRA-Base (12 레이어, 히든 사이즈 768), ELECTRA-Large (24 레이어, 히든 사이즈 1024)의 세 가지 버전이 존재한다. 레이어 수와 히든 사이즈를 늘릴수록 모델은 더 강력한 전이학습을 할 수 있지만, 이는 모델의 복잡도와 연산 비용을 증가시키는 문제로 이어진다. 한국어에 대한 ELECTRA 모델로는 KoELECTRA[30)]와 KcELECTRA[31)]가 대표적이다. 이들 모델은 개인 연구자들이 방대한 한국어 데이터를 구성하여 사전학습한 결과, 한국어 이해력을 향상시킨 모델들이다.

30) monologg KoELECTRA, GitHub repository, https://github.com/monologg/KoELECTRA, 2022.
31) Beomi KcELECTRA, GitHub repository, https://github.com/Beomi/KcELECTRA, 2022.

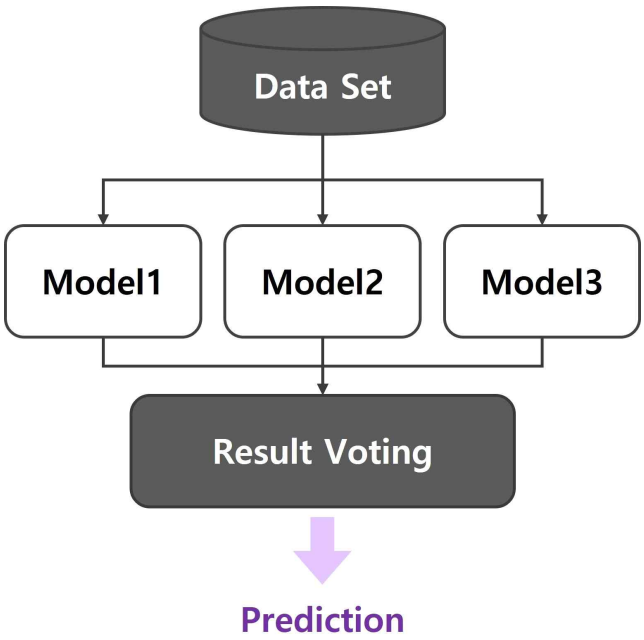
3. 앙상블 분류기

앙상블 기법은 머신러닝에서 널리 사용되는 방법론으로, 이는 여러 모델을 결합하여 예측 성능을 향상시키는 전략을 추구한다. 이 방법은 개별 모델들의 예측 불확실성을 줄이며, 일반화 능력을 향상시키는 데 중점을 두어 전체적인 성능 향상에 기여한다. 주요한 앙상블 방법으로는 보팅(Voting), 배깅(Bagging), 부스팅(Boosting), 그리고 스태킹(Stacking)이 존재한다. 일반적으로, 앙상블의 성능은 기본 모델들이 서로 독립적이고 다양할수록 더욱 향상되는 경향이 있다. 여기서 중요한 점은, 기본 모델들이 각각의 작업에 대해 충분한 성능을 보여줘야 앙상블 전체의 성능이 향상된다는 것이다. 앙상블 기법은 다양한 머신러닝 알고리즘과 딥러닝 모델에 성공적으로 적용되어 왔으며, 많은 연구들은 앙상블 기법이 높은 성능을 달성하는데 있어 효과적인 방법임을 입증하였다.

보팅(Voting)

보팅(Voting)은 여러 기본 모델들의 예측 결과를 집계하여 최종 예측을 도출하는 앙상블 기법으로 자세한 학습 구조는 [그림 2-6]를 통해 확인할 수 있다. 이 방법은 일반적으로 하드 보팅(Hard Voting)과 소프트 보팅(Soft Voting)의 두 가지 주요 방식으로 구분된다. 하드 보팅의 경우, 각 모델의 예측 결과가 다수결 원칙에 따라 결정되며, 이때 모든 모델들은 동일한 가중치를 부여받는다. 이 방식에서는 가장 많은 표를 얻은 클래스가 최종 예측 결과로 선정된다. 소프트 보팅은 각 모델이 제공하는 예측 확률을 평균화하여 최종 결과를 도출하는 방식이다. 이 방법은 각 모델의 예측 신뢰도를 고려하기 때문에, 개별 모델 간의 성능 차이를 반영하는 데 더욱 유리하다.

보팅의 주요 장점은 다양한 모델들의 예측 결과를 결합함으로써 각각의 단일 모델이 가지는 불확실성을 줄이고, 이를 통해 일반화 성능을 개선할 수 있다는 점이다. 그러나 이 방법은 다수결 원칙에 기반하여 모델들의 예측 결과를 결합하는 단순한 방식이기 때문에, 복잡한 상호 작용이나 비선형 관계를 잘 포착하는 데는 제한적이다. 더불어, 모델들이 독립적으로 학습되기 때문에, 각 모델의 오차가 상호 보완되지 않아 모델 간의 불일치가 클 경우 보팅 방식이 효과적이지 않을 수 있다. 이러한 제한 사항들에도 불구하고, 보팅은 여러 모델의 예측을 쉽게 통합할 수 있다는 장점 때문에 매우 효과적인 앙상블 기법으로서 광범위하게 활용되고 있다.

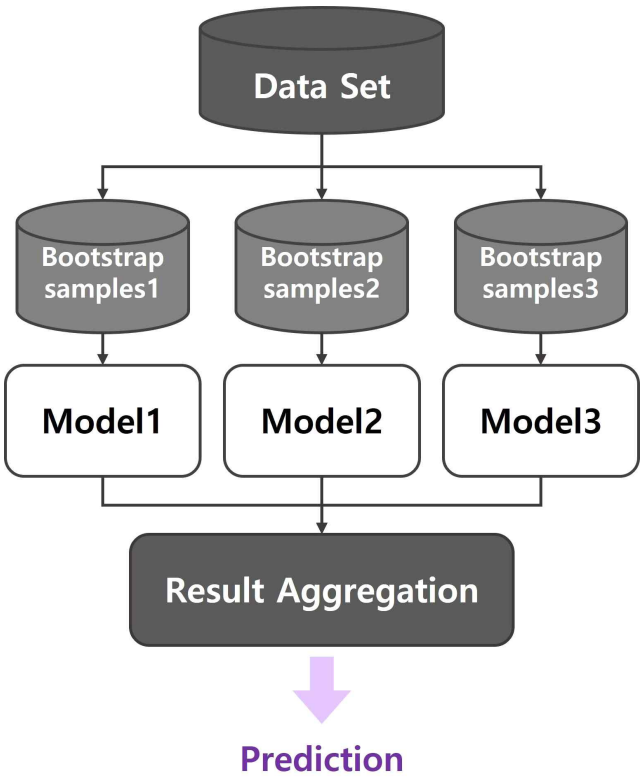


【그림 2-6】 Voting Ensemble Learning Structure

배깅(Bagging)

배깅(Bagging)은 여러 개의 독립적인 모델을 병렬로 학습시키고, 그 결과를 집계하는 앙상블 기법을 가리킨다. 이 기법은 [그림 2-7]과 같이 동일한

알고리즘을 기반으로 다수의 모델을 독립적으로 학습시키고, 그들의 예측 결과를 평균내거나 투표를 통해 최종 예측을 결정하는 방식이다. 대표적인 배깅 알고리즘으로는 랜덤 포레스트(Random Forest)가 있다. 랜덤 포레스트는 여러 개의 결정 트리를 생성하고, 각각의 결정 트리가 독립적으로 예측을 수행한 후 결과를 집계하여 최종 예측을 도출한다. 이를 통해, 결정 트리의 고질적인 오버피팅 문제를 완화하고 모델의 안정성을 높일 수 있다.



【그림 2-7】 Bagging Ensemble Learning Structure

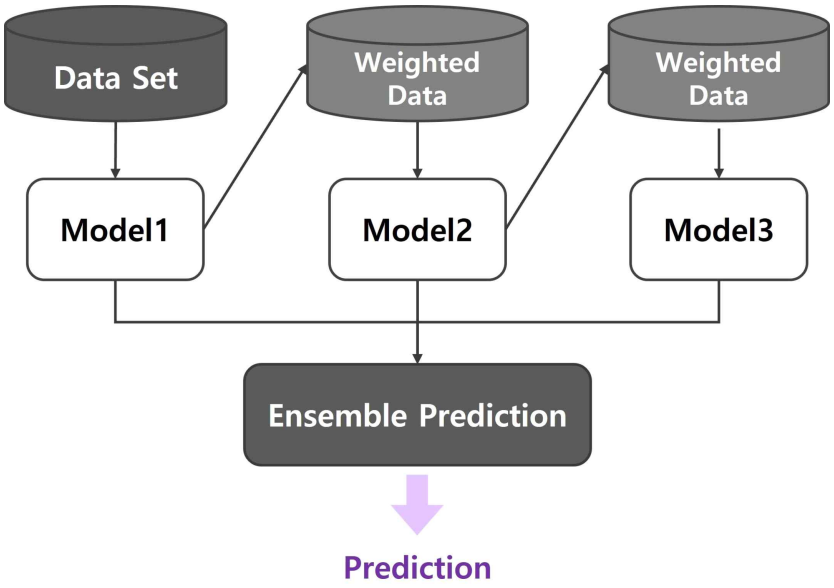
배깅의 주요 장점 중 하나는 훈련 데이터의 샘플링 변동에 대한 모델의 견고성을 향상시키는 것이다. 이는 각각의 모델이 서로 다른 부트스트랩 샘플(Bootstrap sample)을 사용하여 독립적으로 학습되기 때문에 가능하다. 부

트스트랩 샘플링은 원본 데이터셋에서 복원 추출을 통해 새로운 데이터셋을 생성하는 방법이다. 이러한 방법으로 생성된 다양한 데이터셋을 통해 모델은 다양한 패턴에 대해 학습하게 되므로, 새로운 데이터에 대한 일반화 성능이 향상된다. 그러나 기본 모델이 독립적으로 학습되기 때문에, 서로의 오차를 상호 보완하는 데에는 한계가 있다. 또한, 모델의 해석력이 다소 떨어질 수 있는데 그 이유는 각 모델의 결정 경계가 복잡해지고, 이를 해석하기 어려워질 수 있기 때문이다.

부스팅(Boosting)

부스팅(Boosting)은 앙상블 학습 기법 중 하나로서, [그림 2-8]과 같이 순차적으로 모델을 학습시키며 이전 모델의 오차를 개선하는 새로운 모델을 추가하는 방식을 채택한다. 주로 사용되는 부스팅 알고리즘에는 에이다부스트(AdaBoost)와 그레디언트 부스팅(Gradient Boosting)이 포함되어 있다. 에이다부스트는 각 학습 단계에서 잘못 분류된 샘플에 대해 가중치를 증가시킴으로써, 이전 모델이 놓친 부분을 후속 모델이 더 잘 학습할 수 있게 한다. 한편, 그레디언트 부스팅은 손실 함수의 그래디언트를 활용하여 오차를 최소화하는 새로운 모델을 순차적으로 추가하는 방식을 취한다.

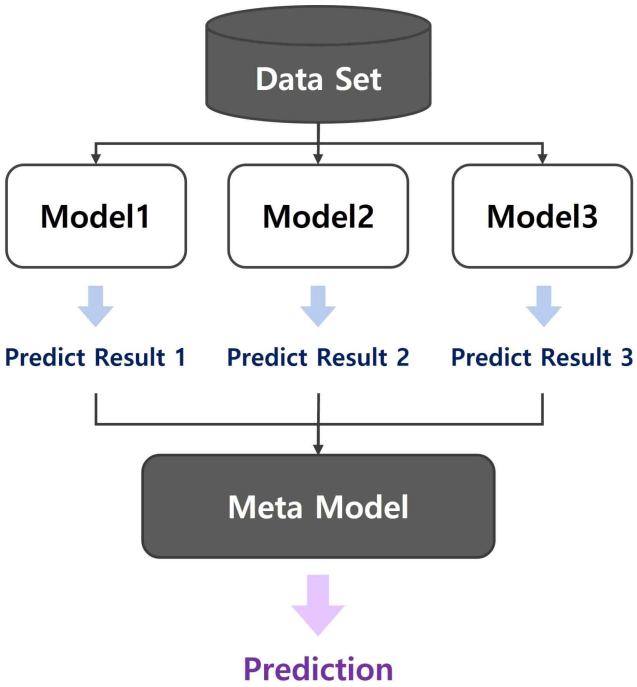
부스팅의 중요한 장점은 이전 모델의 약점을 보완하고, 여러 개의 약한 학습기를 결합하여 강력한 학습기를 구축한다는 점이다. 이러한 방식은 각 순차적 모델이 이전 모델의 오차를 최소화하도록 학습되므로, 최종적으로는 성능의 향상을 이룰 수 있다. 그러나 부스팅의 일부 단점도 고려해야 한다. 모델들이 순차적으로 학습되어야 하므로, 이 과정은 상대적으로 많은 학습 시간을 필요로 한다. 더불어, 부스팅은 훈련 데이터에 대해 과도하게 학습되는 오버피팅의 위험을 갖고 있다.



【그림 2-8】 Boosting Ensemble Learning Structure

스태킹(Stacking)

스태킹(Stacking)은 앙상블 학습 기법의 일종으로, 다양한 단일 모델들의 예측 결과를 복합적으로 활용하여 최종 예측을 도출하는 메타 모델을 중심으로 작동한다. 이 방법은 보팅(Voting), 배깅(Bagging), 부스팅(Boosting) 등의 전통적인 앙상블 방법들이 갖는 일부 단점을 극복함으로써, 모델의 일반화 성능을 심화시키는 데 목표를 둔다. 스태킹의 핵심적인 구성 요소인 메타 모델은 단일 모델들의 예측 결과를 취합하여 새로운 입력 데이터로 활용하며 학습과정은 [그림 2-9]를 통해 자세하게 확인할 수 있다. 이 과정을 통해 단일 모델들이 독립적으로 인식하지 못하는 상호 작용과 비선형 관계를 고려하여 모델의 예측 성능을 강화시킨다. 이 특성은 스태킹 기법이 다른 앙상블 방법에 비해 오버피팅을 방지하고 각 모델의 오차를 상호 보완하는 능력을 갖게 한다.



【그림 2-9】 Stacking Ensemble Learning Structure

스태킹은 다양한 종류의 단일 모델을 통합하여 활용할 수 있음을 특징으로 한다. 이를 통해 모델의 다양성을 활용하고, 각기 다른 알고리즘을 기반으로 한 모델들의 장점을 최대화하며 단점을 상호 보완함으로써 성능 향상을 도모할 수 있다. 이에 따라, 선형 모델, 결정 트리 등 다양한 알고리즘을 단일 모델로 사용하여 각 모델의 장점을 극대화하고 단점을 보완하여 최종적으로 높은 성능의 모델을 구축할 수 있다. 스태킹은 단일 모델들의 예측 결과를 통합하는 메타 모델의 도입과 모델의 다양성을 통해 개별 모델이 가진 한계를 극복하고, 각 모델의 성능을 강화하는 효과적인 앙상블 기법으로서, 다양한 연구들이 이러한 스태킹의 성능을 입증해 왔다.

예를 들어, 허지혜(2021)[32]의 연구에서는 X-ray/CT 영상 데이터인

32) Ji Hye Heo, Su Bin I, Won Hyuk Yang, Dong Hoon Lim, “Transfer learning-based ensemble deep learning for image classification of COVID-19 patients,” Journal of the

COVID-Xray-5k 데이터, Mendeley COVID-19 데이터, 그리고 Kaggle COVID-19 데이터를 활용하여, COVID-19 환자를 감지하기 위한 전이학습 모델을 기반으로 한 스태킹 앙상블 모델을 제시하였다. 이를 위해 기존 딥러닝 모델의 한계를 극복하기 위한 전이학습 모델인 AlexNet, ResNet, Inception, 그리고 DenseNet을 활용하여 실험을 진행하였고, DNN을 메타 모델로 활용하여 스태킹 앙상블을 실행하였다. 이 앙상블 모델은 정확도, 정밀도, 특이도, 재현율, F1-Score 등의 성능지표에서 높은 수치를 기록하였다.

이신행(2022)[33]의 연구에서는 유튜브 댓글 데이터를 통해 악플 여부를 판별하는 스태킹 앙상블 모델을 학습하였다. 이를 위해 댓글 데이터에서 언어적 특성을 추출하였고, 이를 바탕으로 Logistic Regression, Random Forest, 그리고 Support Vector Machine과 같은 다양한 기계 학습 알고리즘을 활용하여 기본 모델을 학습시켰다. 이후, Logistic Regression 모델을 메타 모델로 이용하여 각 기본 모델의 예측 결과를 통합, 스태킹 앙상블을 구축하였다. 실험 결과, 이 스태킹 앙상블 모델은 평가 데이터 세트의 댓글에서 악플을 분류하는 데 있어 정확도와 F1-Score 기준에서 단일 알고리즘을 사용한 방식에 비해 상당히 우수한 성능을 보였다. 이를 통해 스태킹 앙상블 모델의 효과적인 적용 가능성을 확인하였다.

김민기(2022)[34]연구는 과일의 품질 세분화에 대한 마케팅 요구가 증가하는 추세를 고려하여, 컴퓨터 비전 기술을 이용해 과일의 등급을 자동으로 분류하는 스태킹 앙상블 모델을 도입하였다. 연구에서는 과일의 시각적 특

Korean Data And Information Science Society, 32(6), pp. 1219-1235, 2021.

33) Lee Shin Haeng, “Machine Learning for Detecting Malicious Comments on YouTube: Focusing on the Application of Stacking Ensemble Model,” Journal of The Korean Data Analysis Society, 24(4), pp. 1583-1598, 2022.

34) Min-Ki Kim, “Automatic Fruit Grading Using Stacking Ensemble Model Based on Visual and Physical Features,” Journal of Korea Multimedia Society, 25(10), pp. 1386-1394, 2022.

징을 파악하기 위해 컨볼루션 신경망(CNN)을, 물리적 특징을 추출하기 위해 퍼셉트론 신경망을 각각 학습시켰다. 두 모델의 출력을 완전연결층으로 이어 스태킹 앙상블을 형성하였고, 이 방식을 통해 높은 정확도인 99.9%를 달성하였다. 이는 스태킹 앙상블 방법이 높은 정확도를 요구하는 과일 등급 분류 작업에서 효과적으로 활용될 수 있음을 보여준다.

남명우(2021)[35] 연구는 원격 진단의 비대면화가 강조되는 현재 상황에서 폐음 데이터를 활용하여 이상 호흡음을 분류하는 스태킹 앙상블 모델을 제시하였다. 이를 위해 K-NearestNeighbors, Decision Tree, Support Vector Machine, Gaussian Naive Bayes, Random Forest, 그리고 Gradient Boosting 등의 다양한 분류 알고리즘을 조합하여 사용하였다. 이 모델들의 예측 결과를 통합하여 최종적인 예측을 수행하는 메타 모델로는 Simple Linear Model인 Logistic Regression을 도입하였다. 이를 통해 달성한 실험 결과는 기본 모델 대비 약 6.1%의 정확도 향상을 보였다. 이는 스태킹 앙상블 방법이 이상 호흡음 분류의 성능 향상에 기여하였음을 입증하는 중요한 결과로 해석할 수 있다.

35) Myung-woo Nam, Young-Jin Choi, Hoe-Ryeon Choi, Hong-Chul Lee, “Parallel Network Model of Abnormal Respiratory Sound Classification with Stacking Ensemble,“ Journal of the Korea Society of Computer and Information, 26(11), pp. 21-31, 2021.

Ⅲ. 시스템 구성

본 논문에서 제안하는 이모티콘 추천 시스템은 크게 (1) 데이터 수집 및 전처리, (2) 컨텍스트 인지 이모티콘 클러스터링 (3) 계층적 모델 학습, (4) 추천 시스템으로 구성되며 전체적인 시스템 구조는 [그림 3-1]과 같다.

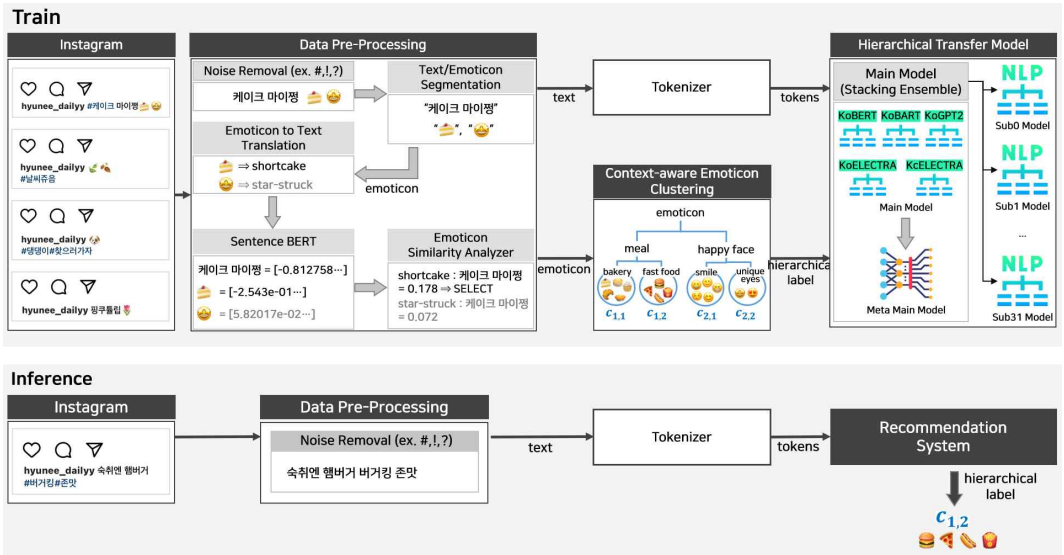
첫 번째, 데이터 수집 및 전처리 단계는 잡음 제거, 이모티콘/텍스트 분할, 이모티콘-텍스트 변환, 그리고 이모티콘-텍스트 유사도 검사의 4개의 부분으로 구성된다. 잡음 제거 부분에서는 인스타그램 게시글을 수집한 후 노이즈와 중복 문장을 제거한다. 이모티콘-텍스트 변환 부분에서는 문장으로부터 텍스트와 이모티콘을 분할하고, 이모티콘-텍스트 변환 부분에서는 이모티콘 사전을 이용하여 이모티콘을 텍스트로 변환한다. 이후 이모티콘 유사도 검사 부분에서 여러 개의 이모티콘들 중 텍스트와 유사도가 가장 높은 이모티콘 한 개를 선정한다.

두 번째, 컨텍스트 인지 이모티콘 클러스터링 단계에서는 이모티콘 벡터를 대상으로 계층적 K-means 클러스터링을 적용한다. 거리 기반으로 그룹 간 비유사도를 최소화하는 방식으로서 그룹 내에 컨텍스트가 유사한 텍스트들이 모이는 효과가 있다. 이 과정은 자율 학습으로서 레이블이 없는 이모티콘 데이터에 계층적 레이블(메인 카테고리, 서브 카테고리)을 자동으로 달아주는 역할을 수행한다.

세 번째, 계층적 학습 단계에서는 (텍스트, 메인 카테고리, 서브 카테고리) 데이터셋을 사용하여 텍스트를 이모티콘 레이블로 분류하는 학습을 진행한다. 전 단계에서 생성한 계층적 이모티콘 레이블을 학습하기 위하여 메인 카테고리를 분류하는 메인모델과 서브 카테고리로 분류하는 서브모델로 구

성 된 계층적 모델을 구축하였다. 이 단계에서는 KoBERT, KoBART, KoGPT2, KoELECTRA, KcELECTRA등의 한국어 전이학습 모델을 도입하여 비교 실험을 수행한다. 이때, 모델의 정확도를 향상시키기 위해 메인모델에 스태킹 앙상블을 적용 하였다.

네 번째, 이모티콘 추천 단계에서는 계층적 학습 단계에서 구축한 모델을 활용하여 SNS 게시글에 대해 계층적 이모티콘 카테고리를 추론한다. 해당 서브 카테고리에 있는 이모티콘들을 무작위로 추천한다.



【그림 3-1】 System Overview

Ⅳ. 데이터 구성 및 전처리

1. 데이터 수집

본 논문은 인스타그램 문맥 인지를 통해 적절한 이모티콘을 추천하고자 한다. 이 과정에서 계층적 KoBERT 모델 학습에 필요한 대량의 텍스트 데이터를 확보하기 위해 다양한 이모티콘의 사용 상황을 반영한 실제 인스타그램 게시글을 수집하였다. 이 과정은 파이썬의 Selenium 패키지를 이용하여 구현된 웹 크롤러를 통해 자동화되었다.

데이터 수집의 첫 단계에서는, Selenium을 이용하여 인스타그램 웹사이트에 접속하여 로그인을 한다. 이후 단계에서는 특정 해시태그를 통해 게시글을 검색하고, 검색된 게시글 중 최신 게시글부터 순차적으로 데이터를 수집하였다. 이때, 본 연구가 텍스트 분류를 기반으로 이모티콘 추천 문제를 다루는 점을 고려하여, 게시글의 본문 내용만을 수집 대상으로 설정하였다. 해시태그 선정 과정에서는 일상적인 생각과 경험을 공유하는 데 주로 사용되는 #일상, #일상스타그램, #일기, #일기스타그램 등을 선택하였다. 이러한 해시태그 선택을 통해 앞서 언급하였듯이 실제 세계에서 다양한 이모티콘 사용 상황을 반영하는 데이터셋을 구축하기 위해서다. 최종적으로 총 119,148개의 인스타그램 게시글을 수집하였으며 [표 4-1]은 일부 예시를 보여준다.

【표 4-1】Example of an Instagram post

“#일상” 수집 결과	“#일기” 수집 결과
더워서 녹을 것 같아 😂	7월 중순이 넘어서야 올리는 7월 일기📒
예뻤던 강릉 숙소! 울집이었음 좋겠네🍊	쏘 럭키걸 🍀
현진이 전용 포즈✌✌	보라보라한 다꾸~💜💟
나랑 데이트 할 사람 🌈💕🌟	대청소 한날🧹
오랜만에 한남동 나들이😍 #콘하스	책읽었어요📚
Arte 🎨 . . . . . . #첫줄반사 #좋아요 #좋반 #일상 #소통	🌻 해바라기랑 은방울꽃 다꾸 은방울꽃동숲에서 엄청 좋아하는꽃인데 동숲 다꾸 할때 쓸까 했다가 아끼똥될까봐 얼른 해바라기 스티커랑 같이 양면다꾸!! 이번에는 꼭 여름스티커 산거 다쓸꺼야!!
꽃 너무 이뻐 💐🌷 맨날 사주세요,,😍	2022.07.17 2022 싸이 흠뻑쇼 대기중💦💦
공감100퍼네... 실컷 소리 지르고 울고싶은 지금인데... 노래방가서 해야되나?ㅋㅋㅋ🎶	상큼이 시트러스도 스토어에 올라갔습니다~🍋 습도도 높고 날도 더우니까 상틈한게 엄청 땡기는것 같아요!
여기보세요 찰칵📸	
#첫줄 연희동에서 부런치🥗	오늘의 간식은 ~~🍩🍫 이마트 편의점 사장님의 추천으로 커피랑 같이 먹기☕
WATERBOMB 🔫 #00년생 #23 #카페 #일상	뱃속 아가와 보내는 첫크리스마스🎅🎄 그 어느때보다 행복한 크리스마스였다
🥨 프레첼 먹고파요	구름위에 나는날☁☁
아키토깽이🐰❤	#정신차리고보니_텅장러💸
퓨딩🍮... 🥄	유니크함에 색감까지✨
적셔~~~🌊	운동을 하시오🏋🏋🏋

2. 데이터 전처리

인스타그램 사용자들이 '좋아요'를 얻기 위해 게시물에 여러 개의 해시태그를 활용하는 특성을 고려하여 중복 게시물을 제거하는 과정을 수행하였다 가장 먼저, 수집된 게시물에서 데이터 임베딩을 위한 전처리 과정으로 해시태그 기호인 #, 감탄사나 의문문을 나타내는 !, ? 등의 특수기호를 제외하고, 각 문장은 줄 바꿈을 기준으로 분리하였다. 그 다음으로, 각 문장에서 이모티콘을 분리하여 '(텍스트, 이모티콘)' 쌍으로 구성된 데이터셋을 구축하는 작업을 진행하였다. 이를 위해 한국어 형태소 분석을 위한 오픈소스 라이브러리인 'Mecab-Ko'를 활용하였다. Mecab-Ko는 입력된 한국어 문장을 형태소 단위로 분리하며, 이 과정에서 이모티콘은 기호를 의미하는 [SY]라는 품사태그로 분리되었다. 본 연구에서는 모든 특수 기호를 제거한 후에 형태소 분석을 수행하였기 때문에, [SY] 태그로 분리된 토큰은 이모티콘만을 의미한다. 이때, 하나의 한국어 문장은 2개 이상의 이모티콘을 포함할 수 있다. 이 경우, 본 논문에서는 문장과 이모티콘 간 코사인 유사도([수식 4-1] 참고)를 계산하여 가장 유사도가 높은 이모티콘 하나만을 선택한다. 이를 위해, [표 4-2]의 이모티콘 사전을 사용하여 이모티콘을 텍스트로 변환하고, Sentence BERT[36]를 구현한 문장 트랜스포머(Sentence Transformer) 패키지를 사용하여 이모티콘 변환 텍스트와 문장을 각각 벡터화한다. 768차원의 임베딩 벡터로 변환된 이모티콘과 문장 간의 코사인 유사도를 계산하여 가장 유사도가 높은 이모티콘 하나만을 선택하고, 나머지 이모티콘은 삭제하였다. 또한, 품사 태깅 과정을 통해 [SY]태그가 없는, 즉 이모티콘이 포함되

36) Reimers, Nils and Iryna Gurevych, "Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks," Conference on Empirical Methods in Natural Language Processing, 2019.

지 않은 문장들을 제거하여, 이모티콘이 포함된 문장들로만 이루어진 데이터셋을 구축하였다.

$$\cos(\Theta) = \frac{A \cdot B}{\| A \| \| B \|} = \frac{\sum_{i=1}^{n} A_i \times B_i}{\sqrt{\sum_{i=1}^{n} (A_i)^2} \times \sqrt{\sum_{i=1}^{n} (B_i)^2}}$$

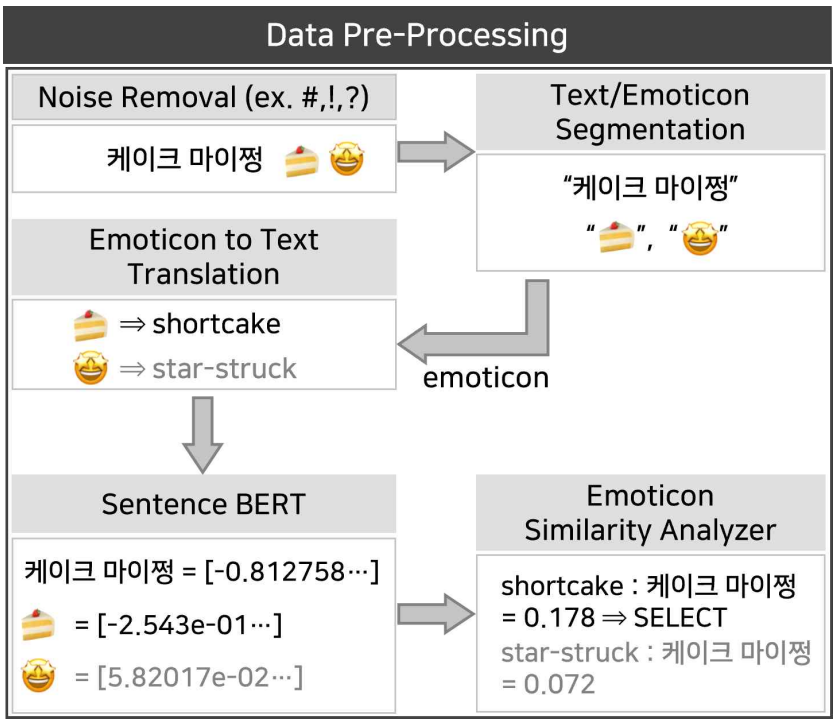
【수식 4-1】 cosine similarity

【표 4-2】 Emoticon Dictionary

Emoticon	Emoticon name	Emoticon	Emoticon name
😿	crying cat	😽	kissing cat
🤣	rolling on the floor laughing	😊	smiling face with smiling eyes
💑	couple with heart	🙂	slightly smiling face
🍴	fork and knife	🍑	peach
🎂	birthday cake	🍵	teacup without handle
🦦	otter	🐹	hamster
💞	revolving hearts	❤️	read hearts
🌈	rainbow	⛈️	cloud with lightning and rain
🌨️	cloud with snow	⛄	snowan
💃	woman dancing	🏃	person running

예를 들어, [그림 4-1]에서 보여지는 것처럼 “#케이크 마이쩡🍰🤩”이라는 문장이 입력으로 들어왔을 때, 첫 단계로 특수기호인 #을 제거하는 작업을 진행한다. 이 입력은 여러 문장으로 이루어져 있지 않으므로, 문장 분리 과정을 생략한다. 이후에는 Mecab-Ko를 이용하여 문장 내의 형태소에 대해 품사 태깅을 진행한다. 결과적으로 [('케이크', 'NNG'), ('마이쩡', 'VA'), ('

🍰', 'SY'), ('🤩', 'SY')]라는 품사태그가 붙은 형태소 집합을 얻게 되며, [SY]태그가 붙은 형태소를 분리하여 ("케이크 마이쩡"과 "🍰", "🤩")같이 '(텍스트, 이모티콘)' 쌍을 구성한다. 해당 문장은 2개의 이모티콘을 포함하고 있기에, 문장과 이모티콘 간의 코사인 유사도를 계산하는 과정이 필요하다. 이를 위해, sentence BERT모델을 사용하여 "케이크 마이쩡"를 [-0.81275827 -0.09309326 ... 0.3127357]와 같은 768차원의 임베딩 벡터로 변환한다. 이후에는 "🍰", "🤩" 이모티콘을 동일하게 768차원의 벡터로 변환한다. 임베딩 된 문장 벡터와 이모티콘 벡터 간의 코사인 유사도를 계산하여 tensor(0.1780)과 tensor(0.0719) 중 더 높은 유사도를 보이는 "🍰"를 선택한다.



【그림 4-1】 Data Preprocessing in System Overview

마지막으로, 학습에 방해가 될 수 있는 노이즈를 줄이기 위해 10회 이하로 사용된 이모티콘은 제거하였다. 이 과정에서 초기 데이터셋에 포함되었

던 1,336개의 이모티콘 수는 최종적으로 616개로 감소하였고 학습 데이터도 179,078건에서 172,149건으로 감소하였다.

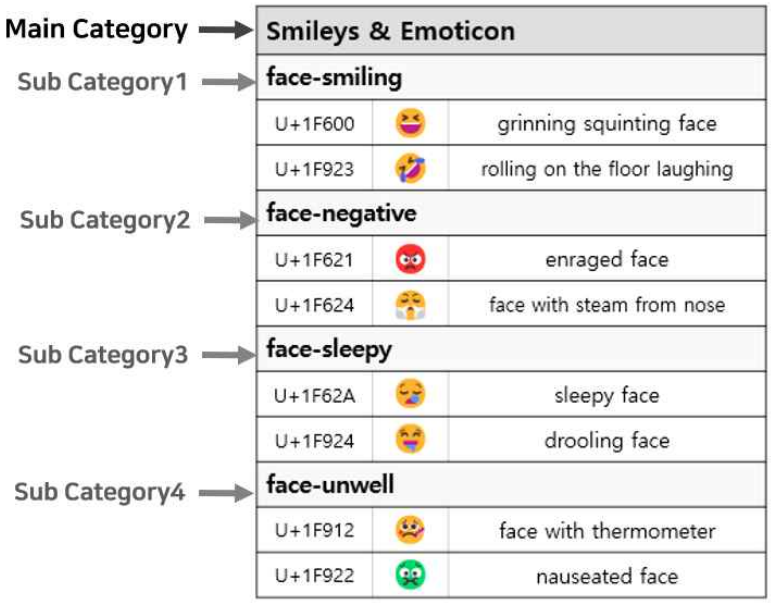
이러한 전처리 과정을 거쳐 구축한 데이터는 총 172,149건으로 학습 데이터 120,489건, 검증 데이터 34,338건, 평가 데이터 17,322건으로 대략 7:2:1로 나누어 학습과 성능 평가를 실시하였다. 자세한 전처리 과정에서의 데이터 개수 변화는 [표 4-3]에서 확인할 수 있다.

【표 4-3】 Changes in Data Count During Data Preprocessing

Preprocessing Stage	Data Count
Post Collection	119,148
Duplicate Post Removal	106,199
Multi-sentence Splitting	632,567
Removal of Sentences without Emoticons	179,078
Low-frequency Emoticon Removal	172,149

3. 이모티콘 계층적 클러스터링

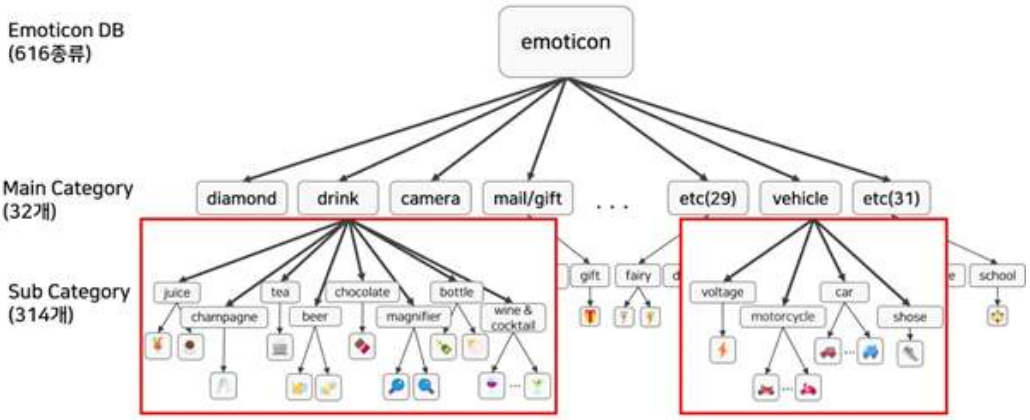
수 천개의 다양하고 방대한 이모티콘은 일반적으로 [그림 4-2]와 같이 트리 형태의 계층적 이모티콘 카테고리 형태로 제공되고 있다. 유니코드 이모티콘의 경우, 이모티콘 카테고리는 메인 카테고리와 서브 카테고리로 구성된다. 예를 들어, 'Smileys & Emoticon' 이모티콘 메인 카테고리 하위에 'face-smiling', 'face-negative', 'face-unwell' 등의 이모티콘 서브 카테고리가 있는 형태이다.



【그림 4-2】 Unicode Emoji Category and Subcategory

유니코드 이모티콘의 서브 카테고리에 착안하여 이 연구에서는 32개의 메인 카테고리 하위에 314개의 서브 카테고리로 구성되는 계층적 카테고리를 [그림 4-3]과 같이 생성하였다. 인스타그램 게시글 119,148개를 수집하여 게

시글에 포함되어 있는 이모티콘 데이터를 분석하고 의미 있게 사용되고 있는 이모티콘 616개를 추출하였다. 계층적 클러스터링을 통해 이모티콘 카테고리를 2단계로 구성하였다. 카테고리는 음식, 사람, 동물, 꽃 등의 큰 분류를 포함하며, 이 중 음식 카테고리 하위에 있는 서브 카테고리는 피자, 떡볶이, 컵 케익 등을 포함한다.



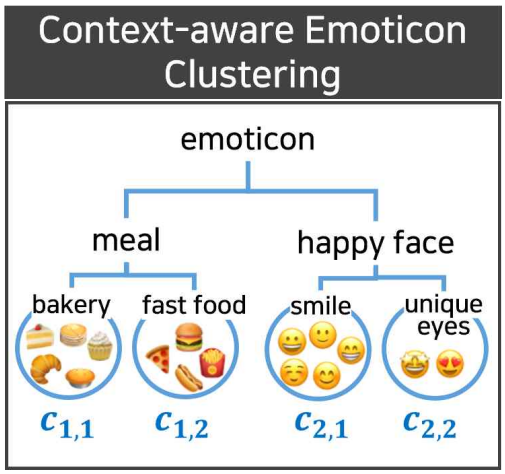
【그림 4-3】 Emoticon Main Category and Sub Category

4.2절의 데이터 전처리 과정에서 이모티콘은 이모티콘 사전을 이용하여 텍스트로 변환하고, Sentence BERT를 도입하여 768차원의 이모티콘 벡터로 변환하였다. 이번 절에서는 616개의 이모티콘 벡터를 대상으로 컨텍스트가 유사한 이모티콘들을 계층적으로 클러스터링하여 메인 그룹과 각 메인 그룹에 속하는 서브 그룹을 생성한다. Sentence BERT를 이용하여 추출한 이모티콘 벡터를 대상으로 K-means 클러스터링을 두 차례 수행함으로써 컨텍스트가 유사한 이모티콘들이 계층적으로 군집화하는 역할을 한다.

이모티콘 벡터 데이터셋 $x_1, x_2, ..., x_{616}$에 대해서 탑다운 방식의 계층적 K-means 군집화를 적용한다. 1단계 K-means 군집화를 수행하여 메인 그룹 32개(G_i, $i = 0, 2, ..., 31$)를 생성하고, 2단계에서 각 G_i에 속한 데이터들에 대해서 재귀적으로 K-means 군집화를 수행하여 서브 그룹 314개

($G_{i,j}, i = 0, 2, ..., 31, j = 0, 1, ..., \#\ of\ clusters$)를 생성한다. 이모티콘 계층적 군집화 결과물인 $G_{i,j}$는 텍스트 데이터를 자동으로 라벨링 하는데 사용된다.

예를 들어, [그림 4-4]와 같이 베이커리 이모티콘 "🍰", "🥐"등과 패스트푸드 이모티콘 "🍕", "🍔" 등, 그리고 "🙂", "😊", "😍"와 같이 다양한 표정을 담고 있는 얼굴 이모티콘이 입력으로 들어왔다고 가정한다. 먼저, 이 이모티콘은 4.2절과 동일하게 이모티콘 사전을 통해 ("🍰"-"shortcake"), ("🍕"-"pizza"), ("🙂", "slightly smiling face")와 같이 텍스트로 변환된다. 이러한 텍스트 변환 이후, sentence BERT를 이용하여 각 텍스트는 [-0.4829146 -0.02738248 ... -0.02582658]와 같은 768차원의 임베딩 벡터로 변환된다. 이어서, K-means 클러스터링 방법을 사용하여 크로아상, 피자, 케이크, 햄버거와 같은 식품 이모티콘들이 "meal"라는 메인 카테고리로 행복한 표정을 담은 얼굴 이모티콘들이 "happy face"라는 메인 카테고리로 분류된다. 그 다음 단계에서, "meal" 카테고리 내부에서 K-means를 한번 더 수행하여 크로아상, 케이크는 "bakery"라는 서브 카테고리로, 피자와 햄버거는 "fase food"라는 서브 카테고리로 분류된다. 마찬가지로 "happy face" 카테고리 내에서도, 눈 모양이 특이한 얼굴 이모티콘은 "unique eyes", 웃고 있는 얼굴 이모티콘은 "smile"이라는 서브 카테고리로 각각 분류된다. 이와 같은 문맥 인지 이모티콘 군집화를 통해 계층적 카테고리를 구성한다.



【그림 4-4】 Context-awere Emoticon Clustering in System Overview

K-means 클러스터링 과정에서 임계치는 이모티콘 군집화 결과의 품질에 큰 영향을 미친다. K값이 커지면 다른 컨텍스트를 가지는 이모티콘들이 동일한 그룹으로 분류되고, K값이 작아지면 과다한 상세 분류가 되어 계층적 KoBERT 학습 성능을 저하시키는 결과를 초래한다. 실제, 본 논문에서는 K값을 증가시켰을 때, ETC로 분류된 메인 카테고리의 이모티콘이 세부적으로 분류되길 기대하였지만, 오히려 같은 카테고리에 속해야 하는 하트 카테고리의 이모티콘이 세분화되는 현상이 관찰되었다. 반대로 K값을 감소시켰을 때, 과일 카테고리와 식물 카테고리 등이 합쳐지는 의미 있는 결과를 기대하였지만, 실제로는 ETC 카테고리들이 더 거대한 그룹을 형성하여 모델 학습의 정확도를 감소시켰다. 이러한 문제를 해결하기 위해, 인공지능 전공 석사 과정에 있는 2명의 연구원이 참여하여 그룹화 결과 각 그룹에 있는 데이터들이 유사한 컨텍스트를 가지는지 판단하는 역할을 수행하였다. 연구원들은 주어진 데이터셋에 다양한 임계치를 적용한 결과 그룹의 적절성을 평가하고 논의를 통해 군집화 품질을 최적화하는 임계치를 경험적으로 도출하

였다. 향후 연구에서 임계치를 자동으로 결정하는 최적화 알고리즘이 필요하다.

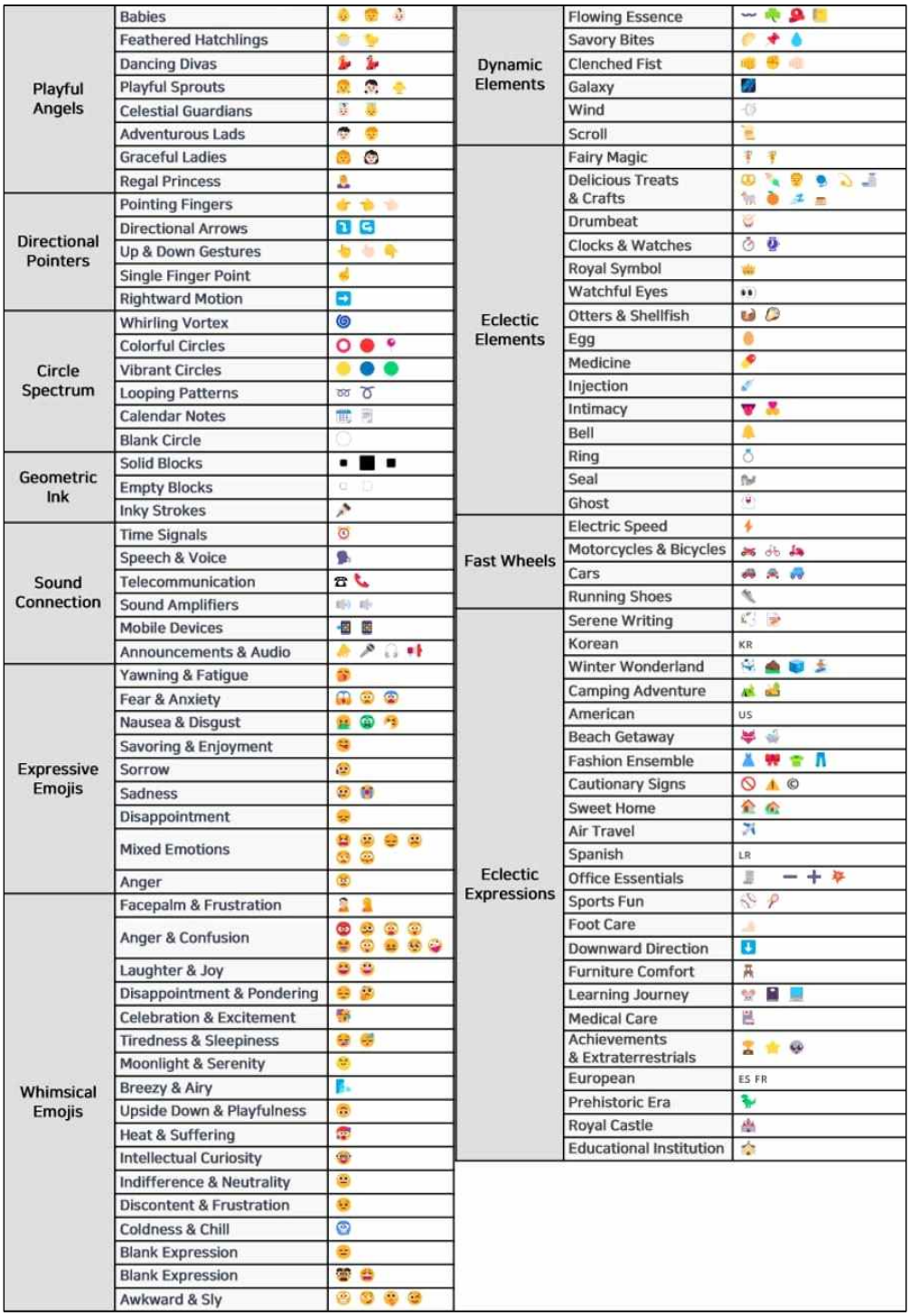
[그림4-5]은 이모티콘 계층적 군집화 결과를 보여준다. 1단계 32개의 메인 카테고리와 2단계 314개의 서브 카테고리로 구성된 트리 구조의 군집화 결과이다.

main	sub	emoticon
Elemental Gems	Sparkling Triad	
	Stones of Brilliance	
	Diamond Solitaire	
Drink	Exotic Brews	
	Bubbly Celebration	
	Refreshing Quench	
	Cheers of Brew	
	Search & Discover	
	Sweet Indulgence	
	Celebrate & Cheers	
	Joyful Bottles	
Media Masters	Hospitality Havens	
	Visual Canvas	
	Capture Chronicles	
	Selfie Snapshots	
	Screen Stories	
Gifted	Shopping Bonanza	
	Thoughtful Mail	
	Prepared Essentials	
	Knowledge Bound	
	Payment Power	
	Inbox Treasures	
	Scholar's Pursuit	
	Carry Essentials	
	Deliver Express	
	Gifted Surprises	
Critter Crew	Jolly Santa	
	Feline Friends	
	Laughing Kitties	
	Canine Companions	
	Affectionate Smooches	
Whimsical Waves	Thoughtful Chatter	
	Serene Tides	
	Whirling Breezes	
	Mystical Revelry	
	Enchanting Whales	
Action Icons	Fast Motion	
	Emergency Alert	
	Controlled Action	
	Successful Completion	
Active Poses	Relaxed Rest	
	Swimming Strokes	
	Expressive Poses	
	Active Running	
	Energetic Poses	
	Mindful Meditation	
	Respectful Bow	
	Strength Training	
	Assertive Stance	
	Salon Styling	
	Golfing Pros	
	Cycling & Walking	

main	sub	emoticon
Nature's Harvest	Warm Embrace	
	Root Harvest	
	Lush Trees	
	Blooming Beauties	
	Sunlit Elegance	
	Autumn Splendor	
	Natural Treasures	
	Juicy Apple	
	Fresh Greens	
	Evergreen Wonderl&	
	Blooming Tulips	
	Lucky Clover	
	Vibrant Tomatoes	
	Growing Greens	
	Zesty Lemon	
Tender Hearts	Dual Love	
	Heartfelt Emotions	
	Soothing Affection	
	Romantic Connection	
	Decorative Love	
	Refreshing Love	
	Warm Affection	
	Mysterious Love	
	Pure Heart	
	Harmonious Union	
Gestures	Approval Gestures	
	Celebratory Gestures	
	Feedback Gestures	
	Raising H& Gestures	
	Rock-On Gestures	
	Greetings & Farewells	
	Hang Loose Gestures	
	Stop Gestures	
	Companionship Gestures	
	Writing Gestures	
	Wishful Gestures	
	Prayer Gestures	
	Helpful Gestures	
	Peace Gestures	
	Boxing	
	Open H&sGesture	
	PartnershipGesture	
	Raised PalmGesture	
	ApplauseGesture	
	Gloves	
Emotive Signals	Directional Arrows	
	Attention Signs	
	Gender Expressions	
	Anger Burst	
	Confirmation Check	
	White Information Signs	
	Symbolic Marks	
	Red Information Signs	
	Warning Flags	

Culinary Delights	Gourmet S&wiches	
	Savory Seafood	
	Specialty Beverages	
	Satisfying Soups	
	Decadent Desserts	
	Flavorful Rice Dishes	
	Fine Dining Experience	
	Sweet Indulgences	
	Italian Delights	
	Hearty Stews	
	Artisan Breads	
	Celebration Delights	
	Fiery Spice	
	Flaky Pastry Delights	
	Corn Creations	
	Buttery Baked Goods	
	Exquisite Sushi	
	Eggcellent Creations	
Vibrant Emojis	Affectionate Smiles	
	Playful Expressions	
	Emotional Sweats	
	Confused Reactions	
	Covered Faces	
	Curious Glances	
	Sickly Symptoms	
	Angry Outbursts	
	Radiant Sun	
	Exasperated Reactions	
	Money-Faced	
	Skeptical Looks	
Joyful Emojis	Smiling Expressions	
	Cat Cuteness	
	Angel vs. Devil	
	Adoring Affection	
	Tears of Joy	
	Warm Embrace	
	Contented Smiles	
	Cool Confidence	
Artistic Blaze	Whirling Rides	
	Nail Art	
	Cosmic Dreams	
	Creative Tools	
	Starry Delights	
	Enchanting Ideas	
	Creative Tools	
	Action Props	
	Dazzling Flames	
	Eclectic Expressions	
	Footprints	
	Frosty Wonders	
	Hot Delights	
	Mind-Blown	
	Alert Signals	
	Paintbrush Strokes	
	Crafting Cuts	

Literary Realm	Chronicles	
	Volumes	
	Literary	
Treats	Fruity Delights	
	Frozen Indulgences	
	Berry Blossoms	
	Cake Delights	
	Sweet Treats	
	Cheesy Delights	
	Nutty Temptations	
	Waffle Delights	
Dynamic Ensemble	Dynamic Delights	
	Achievement Accolades	
	Majestic Peaks	
	Celestial Phases	
	Melodic Harmony	
	Financial Gains	
	Global Explorations	
	Powerful Strength	
	Perfection Benchmark	
	Hilarious Laughter	
	Precision Strikes	
	Timekeeping Essentials	
	Upcoming	
	Calendar Reminders	
Weather	Precipitation	
	Winter Wonderl&	
	Sunlit Escapes	
	Umbrella Chronicles	
	Tropical Oasis	
	Beach Retreat	
Animal	Mystic Monkey	
	Seaside Delicacies	
	Bear Hugs	
	Equestrian Wonderl&	
	Ocean Serenade	
	Burger Brigade	
	Whimsical Menagerie	
	Feline Majesty	
	Meat Medley	
	Butterfly Bliss	
	Aquatic Wonders	
	Hoppy Companions	
	Feathered Friends	
	Enchanted Equines	
	Sizzling Savor	
	Primate Playmates	
	Marine Marvels	
	Smoky Delights	
	Foxy Charm	
	Alien Encounter	



【그림 4-5】Hierachical Emoticon

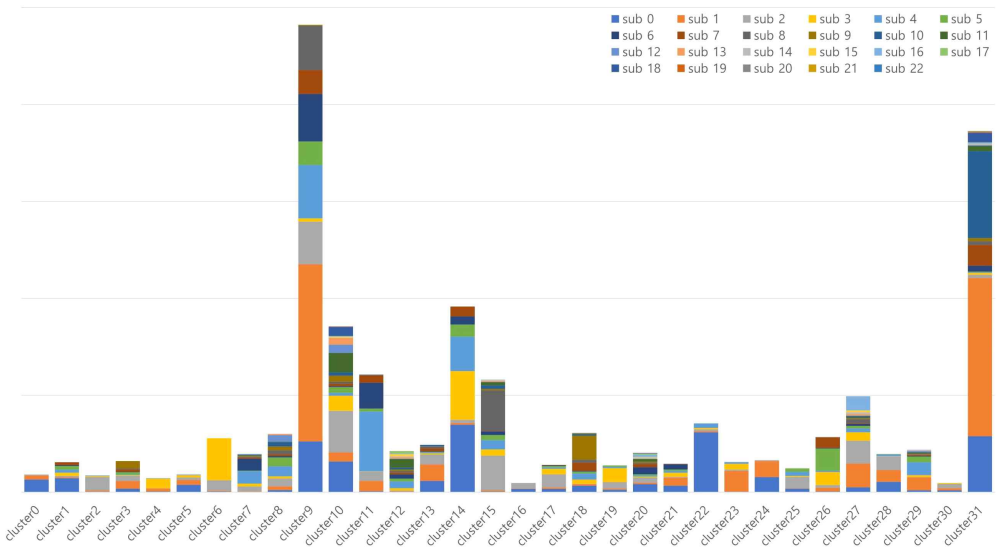
4. 데이터 라벨링 및 분석

전처리 과정을 통해 (텍스트, 이모티콘) 데이터셋을 구축하고 계층적 이모티콘 카테고리를 생성한다. 모든 이모티콘은 계층적 이모티콘 카테고리에 속해 있으므로 모든 게시글은 (텍스트, 계층적 이모티콘 카테고리)로 자동 레이블링한다. 여기에서 계층적 이모티콘 카테고리는 메인 카테고리 번호와 서브 카테고리 번호로 구성되며, $C=\{c_{0,0}, c_{0,1}, \cdots, c_{0,k}, c_{1,0}, c_{1,1}, \cdots, c_{m,n}\}$으로 표현 가능하다. 이때 m은 메인 카테고리의 개수, 즉 32개이고, n은 1단계에 있는 각 메인 카테고리가 보유한 서브 카테고리의 개수이다. 이와 같은 표기를 이용하여 172,149개의 모든 문장에 대해 해당 이모티콘 카테고리 (메인 카테고리 번호, 서브 카테고리 번호)로 레이블을 자동 생성한다([표 4-4] 참조). 여기서 생성한 레이블은 한국어 사전학습 모델 기반의 계층적 네트워크에서 활용되며, 1단계에서 메인 카테고리 번호를 학습하고 2단계에서 서브 카테고리 번호를 학습하는데 사용된다.

본 논문에서 구축한 이모티콘 추천 데이터셋은 심각한 클래스 불균형 문제를 갖고 있음을 [그림 4-6]에서 확인할 수 있다. 일부 이모티콘은 빈번하게 등장하는 반면, 다른 이모티콘들은 상대적으로 드물게 나타난다. 이런 현상으로 인해, 모델은 대중적인 이모티콘에 대해 효과적으로 학습하지만, 희귀한 이모티콘에 대해서는 성능 저하가 발생하게 된다. 이를 해결하기 위해, 학습 과정에서 각 클래스에 가중치를 적용하는 WeightedLoss 방법을 도입하였다. 이 방법에 대한 자세한 내용은 5.1절에서 더욱 상세히 다루게 된다.

【표 4-4】 Example of Experimental Data Configuration

Main Category	Sub Category	Text Data	Hierarchical Label
Main Category 1 (Drink)	Sub Category 1 (Juice)	시원한 에이드 한잔 서비스 드려요	$c_{1,1}$
		제주 여행 펍	$c_{1,1}$
		레인보우는 내가 꼭 먹겠다고 벼르고 있었즤	$c_{1,1}$
	Sub Category 2 (Champaign)	월요병 없는 월요일을 위하여	$c_{1,2}$
		안 놀아줘서 인생 첫 혼술 해본다	$c_{1,2}$
		우아한 스파클링 와인 이랍니다	$c_{1,2}$
Main Category 30 (vehicle)	Sub Category 3 (car)	즐거웠 던 오전 드라입	$c_{30,3}$
		오래 타자 무사고로	$c_{30,3}$
		라붕이 타고 데이트	$c_{30,3}$
	Sub Category 4 (shose)	운동화 잠수 시킨 날	$c_{30,4}$
		융겨미와 함께 하는 조던	$c_{30,4}$
		나이키 골프화	$c_{30,4}$



【그림 4-6】 Imbalanced Data Distribution

Ⅴ. 모델 설계 및 학습

1. 사전학습 언어 모델 학습

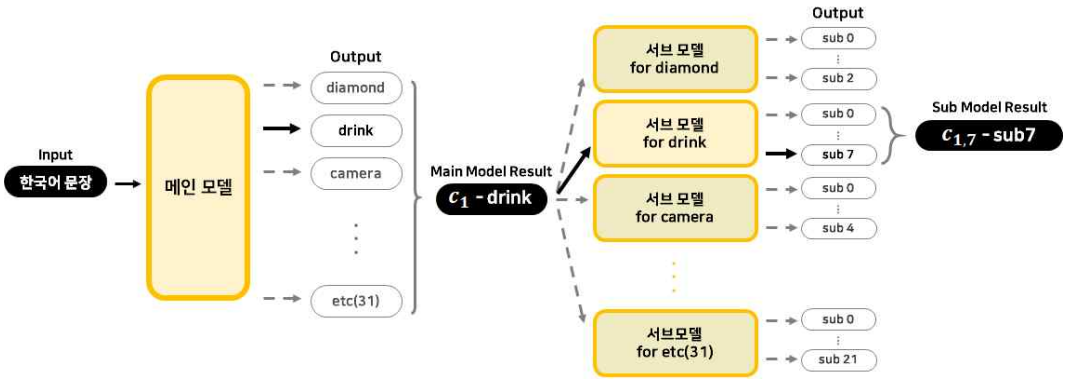
본 논문에서는 BERT 모델의 한국어 버전인 KoBERT 모델을 활용하여 계층적 모델을 구성한다. BERT(Bidirectional Encoder Representations from Transformers)는 트랜스포머 모델의 인코더만을 활용하여 사전 학습된 모델로서 다양한 자연어 처리 태스크에서 뛰어난 성능을 보여주고 있다. 이러한 양방향 인코딩 기능은 문장 내 앞뒤 문맥 정보를 모두 고려하여 한국어 텍스트의 중의성 해소와 문맥 이해력을 향상시키는 장점이 있다. KoBERT 모델은 BERT모델을 기반으로 한국어 데이터로 추가 학습하여 한국어 성능을 향상 시킨 모델이다. 이러한 특성으로 인해 한국어의 다양한 표현과 감정을 함축적으로 포함한 SNS 단문을 파악하는데 유리하다.

그 외에도 KoBART, KoGPT2, KcELECTRA, KoELECTRA 등 다른 트랜스포머 기반의 한국어 사전학습 모델들도 추가로 도입하여 계층적 모델을 구성하였다. 이 모델들 역시 트랜스포머 아키텍처를 기반으로 하여 한국어 데이터로 사전 학습되었으며, 다양한 자연어 처리 작업에서 뛰어난 성능을 보여준다. 대규모 텍스트 데이터를 학습하는 이 모델들은 한국어의 문법, 구문, 의미 등을 보다 정확하게 이해하는 데 큰 장점이 있다. KoBERT, KoBART, KoGPT2, KcELECTRA, KoELECTRA의 기본 사양은 [표 5-1]에서 확인할 수 있다.

【표 5-1】 Korean Pre-learning Model's Configure

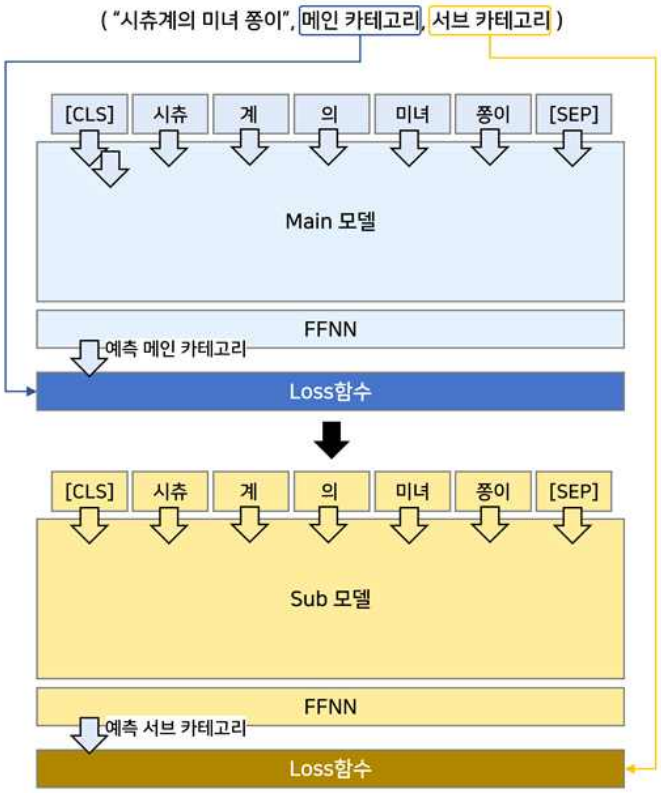
model	develop	data	vocab	params
KoBERT	SKT	위키피디아 50M	8,002	92M
KoBART	SKT	위키피디아, 뉴스, 모두의 말뭉치, 청와대 국민청원 등 약 40GB	30,000	124M
KoGPT2	SKT	위키피디아, 뉴스, 나무위키, 네이버 영화 리뷰 등 약 40GB	51,200	125M
KoELECTRA	개인	뉴스, 위키, 나무위키 약 14GB 모두의 말뭉치(신문, 메신저 등) 약 20GB	35,000	112M
KcELECTRA	개인	뉴스 기사들의 댓글과 대댓글 17.3GB	30,000	127M

본 논문에서는 구축한 데이터셋은 모든 문장이 1개의 이모티콘을 가지고 있고 이모티콘은 계층적 이모티콘 카테고리에 속해 있다. 계층적 카테고리는 32개의 메인 카테고리와 각 메인 카테고리 하위에 3~23개 사이의 서브 카테고리로 구성되어 있다. 이런 구조를 바탕으로 [그림 5-1]과 같이 메인모델과 서브모델로 구성되는 계층적 모델을 구축한다. 이러한 계층적 접근방식은 모델의 정확도를 향상시키는 데 중요한 역할을 하며 이모티콘이 지속적으로 추가되고 변화하는 상황에서 새로운 이모티콘이 도입될 때마다 계층적 모델의 서브모델만을 학습시키면 되므로, 신속한 대응이 가능하다는 장점이 있다.



【그림 5-1】 Hierarchical Model

메인모델은 SNS 게시글을 32개의 메인 카테고리로 분류하는 역할을 하고, 전체 데이터셋을 사용하여 학습한다. 메인모델 하위에 32개의 서브모델을 구성하여 2차 분류를 진행한다. 32개의 서브모델은 각 서브 카테고리에 속하는 데이터셋을 사용하여 학습하고 서브모델에 따라 클래스 개수와 데이터셋 개수는 각각 다르다. 각 서브모델의 클래스 개수는 3~23 범위 내에 있고 모든 서브모델의 클래스들의 총 개수는 314개이다. [그림 5-2]는 계층적 모델의 미세 조정(Fine-Tuning) 과정을 상세하게 설명한다. 본 과정에서는 한국어 문장과 메인 카테고리, 서브 카테고리를 입력으로 받는다. 이를 바탕으로 메인모델은 학습 과정을 통해 메인 카테고리를 예측하게 된다. 그 후, 예측된 메인 카테고리와 실제 메인 카테고리 사이의 차이를 Loss 함수를 통해 측정하게 된다. 이렇게 산출된 Loss는 역전파 과정(Back-propagation)을 통해 가중치를 조정하는데 사용되며, 이를 통해 메인모델의 학습이 이루어진다. 비슷한 방식으로, 서브모델도 학습을 통해 서브 카테고리를 예측한다. 그리고 예측된 서브 카테고리와 실제 서브 카테고리 사이의 Loss를 계산하며, 이 Loss를 역전파 과정에 활용해 서브모델의 학습을 진행한다. 이런 식으로 메인모델과 서브모델 각각의 미세 조정을 통해, 우리는 최종적으로 계층적 모델의 세부적인 조정을 이루어낸다.



【그림 5-2】 Hierarchical Model Fine-Tuning

본 논문에서 구축한 데이터셋에는 데이터 불균형 문제가 존재한다는 점에 주목해야 한다. 특정 카테고리의 데이터가 다른 카테고리에 비해 훨씬 많거나 적을 수 있는데, 이는 모델이 특정 카테고리를 과도하게 학습하거나, 반대로 충분히 학습하지 못하는 문제를 일으킬 수 있다. 데이터 불균형 문제는 모델의 일반화 능력을 저하시키고, 새로운 데이터에 대한 예측 성능을 감소시킬 수 있다. 이러한 문제를 해결하기 위해 weighted loss라는 방법을 도입하였다. 이 방법은 빈도가 낮은 클래스의 손실에 더 큰 가중치를 부여하고, 반대로 빈도가 높은 클래스의 손실에는 더 작은 가중치를 부여하는 방식으로 작동한다. weighted loss에 관한 자세한 수식은 본문의 [수식 5-1]

에서 확인할 수 있으며, 이 수식에서 'L'은 전체 손실을 의미하며, 'N'은 총 샘플 수를 나타낸다. 또한, 'w_i'는 i번째 클래스의 가중치를, 'y_i'는 i번째 클래스의 실제 레이블을, 그리고 '$p(y_i)$'는 모델이 i번째 클래스에 대해 예측한 확률을 각각 나타낸다. 이를 통해 모델이 모든 클래스를 공정하게 학습할 수 있게 된다.

$$L = -1/N \times \sum(w_i \times \log(p(y_i)))$$

【수식 5-1】 Weighted Loss

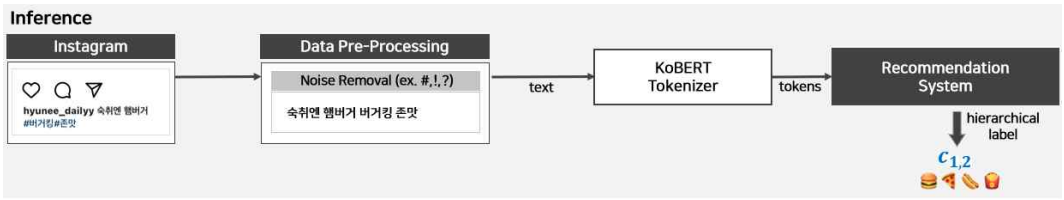
또한, 모델의 학습 속도, 정확도, 그리고 과적합을 결정하는 중요한 요인인 하이퍼파라미터 선택은 극도로 중요하다. 본 논문에서 사용된 학습에 관련된 하이퍼파라미터의 상세한 정보는 [표 5-2]에서 볼 수 있다. 배치 크기(batch size)나 최대 길이(max length)와 같이 세부적으로 조절이 필요한 하이퍼파라미터는 적절한 선택이 더욱 중요하다. 이런 하이퍼파라미터의 최적값은 6장에서 각 모델별 하이퍼파라미터 실험을 통해 결정되었다.

【표 5-2】 Hyperparameters

parameters	value
optimizer	AdamW
epochs	1,000
early stopping	20
learning rate	5e-5

2. 계층적 모델 및 이모티콘 추천

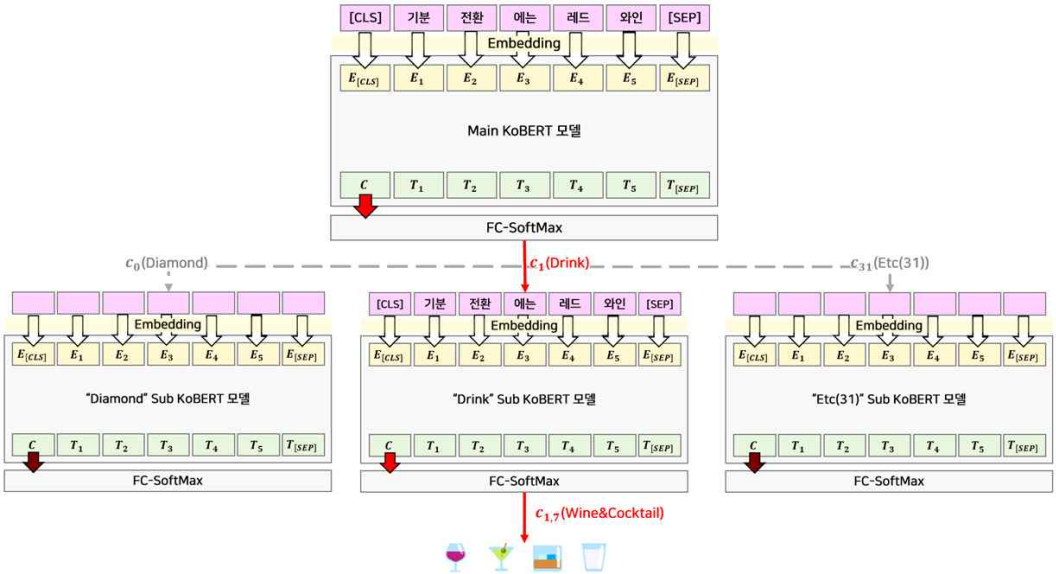
본 논문에서 제안하는 계층적 모델은 메인 카테고리를 분류하는 메인모델 1개와 그 하위에 위치해 서브 카테고리를 분류하는 32개의 서브모델 구성되어 있다. 학습이 완료된 메인모델과 서브모델은 연결되어, 이모티콘 추천 시스템의 구성요소로 작동된다. [그림 5-3]와 같이 “숙취엔 햄버거 #버거킹 #존맛”이라는 한국어 문장이 입력으로 주어지면, 해시태그를 의미하는 #를 제거한다. 이렇게 특수기호가 제거된 텍스트는 토크나이저를 통해 토큰화 되며, 이후 이모티콘 추천 시스템에 입력되어 계층적 분류를 거친다. 그결과, 해당 텍스트는 $c_{1,2}$ 즉, fast food라는 서브 카테고리로 최종 분류되고, 이에 속한 이모티콘들이 무작위로 추천된다.



【그림 5-3】 Inference in System Overview

더욱 구체적으로, 본 연구에서 구성한 데이터를 활용하여 학습한 계층적 KoBERT 모델의 작동 방식은 [그림 5-4]에서 보여진다. 입력 데이터가 제공되면, 먼저 메인모델의 Tokenizer가 데이터를 토큰화하고 임베딩 층을 거쳐 문맥 정보를 추출한다. 이렇게 얻은 문맥 정보는 완전 연결 신경망에 입력되어, 메인 카테고리를 결정하게 된다. 이후 메인모델의 출력을 기반으로, 적절한 서브모델이 선택되며, 해당 서브모델이 서브 카테고리를 결정한다. 이런 식으로 계층적 분류 과정을 거치며, 이모티콘 추천 시스템은 다양한 SNS 문맥에서 가장 적절한 이모티콘 카테고리를 선정할 수 있게 된다. 예를 들어, "기분 전환에는 레드와인"이라는 문장이 주어지면, 메인모델은 32개의 메인 카테고리 중 'Drink'를 선택하고, 이어서 'Drink' 서브모델은 이 카테고리 내에서

'juice', 'champaign', 'wine&cocktail' 중 'wine&cocktail' 서브 카테고리를 결정한다. 그리고 이 카테고리에 속한 이모티콘들이 추천된다. 이렇게 사용자가 입력한 게시글은 $c_{i,j}$와 같은 특정 카테고리로 최종 분류되며, 해당 카테고리에 속한 이모티콘들이 무작위로 추천되는 방식으로 작동한다.

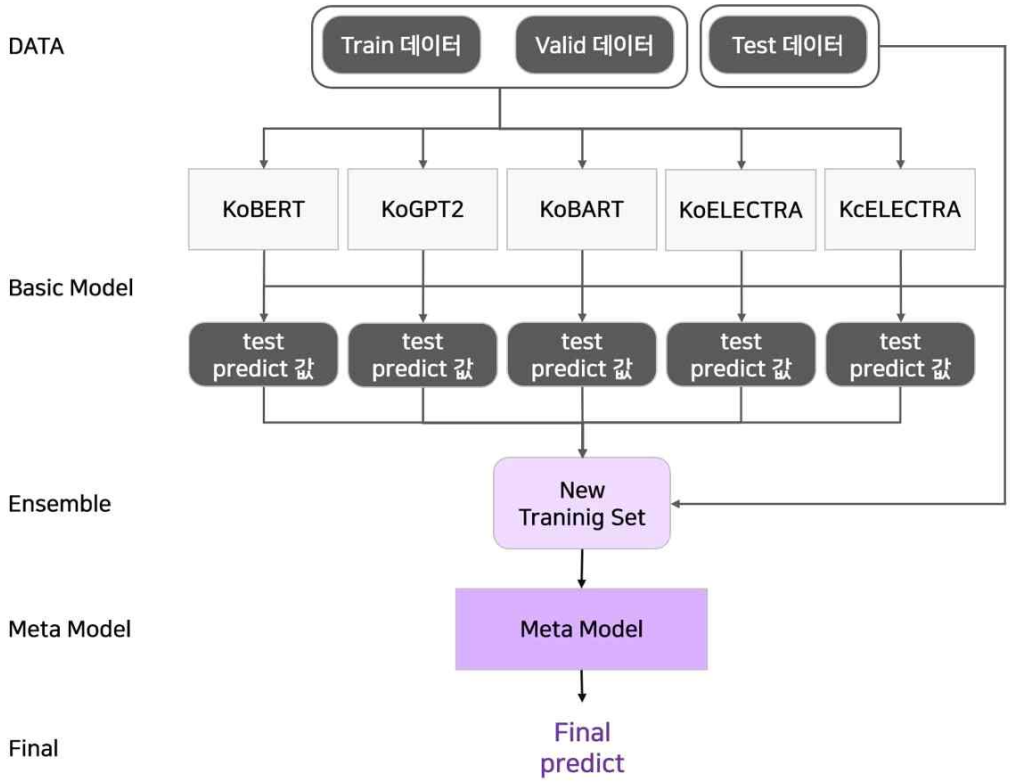


【그림 5-4】 Hierarchical KoBERT Model for Emoticon Recommendation

3. 스태킹 앙상블 모델 학습

계층적 모델의 핵심 요소인 메인모델은 보통 광범위한 이모티콘 카테고리를 포괄하게 되며, 이는 각 메인 카테고리의 독특한 특징이 명확하게 구분되지 않을 가능성을 야기한다. 이로 인해, 분류의 정확도가 저하될 수 있을 뿐만 아니라 광범위한 카테고리를 학습하는 과정에서 모델의 복잡도가 증가하게 되면서 새로운 데이터에 대한 이모티콘 예측 성능이 손상될 수 있다는 문제점도 존재한다.

이 문제에 대한 해결책으로, 본 연구는 메인모델에 스태킹 앙상블을 적용하는 방법을 제안하였다. 이 방법론은 메인모델의 성능을 향상시켜 계층적 모델의 성능을 향상시키고 있다. [그림 5-5]에 표현된 바와 같이, 한국어 사전학습 언어 모델 기반의 스태킹 앙상블은 level-0와 level-1로 이루어져 있다. level-0에서는 KoBERT, KoBART, KoGPT2, KcELECTRA, KoELECTRA를 활용하여 5개의 독립적인 분류 모델을 구성한다. 이들 각각의 모델은 테스트 데이터를 독립적으로 처리하여 예측값을 생성하며, 이러한 예측값들은 결합되어 새로운 데이터셋을 구성한다. 이후, level-1에서 이 새로 생성된 데이터셋을 메타 모델이 학습하여 최종 예측값을 도출하게 된다.



【그림 5-5】Stacking Ensemble Based on Korean Pretrained Language Models

본 논문에서는 스태킹 앙상블 기법을 메인모델에 적용하였다. 이 기법에 대한 자세한 구현 방식은 [표 5-3]의 유사 코드(Pseudo code)에서 확인할 수 있다. 이 유사 코드는 이미 학습이 완료된 한국어 전이학습 모델을 기반으로 한 메인모델을 활용하여 새로운 데이터셋을 생성하는 과정을 설명하고 있다. 이후 이 새로 생성된 데이터셋을 바탕으로 메타 모델을 학습시키며, 이렇게 학습된 메타 모델은 다시 메인모델로 활용된다. 이를 통해, 메인모델과 서브모델이 서로 연결되는 구조를 확인할 수 있다. 본 과정의 이해를 돕기 위해 [그림 5-6]을 참고하면 더욱 명확한 이해가 가능하다. 이외에도, Logistic Regression, XGB (eXtreme Gradient Boosting), 그리고 LGBM (Light Gradient Boosting Machine)을 포함한 총 세 가지 메타 모델을 도입

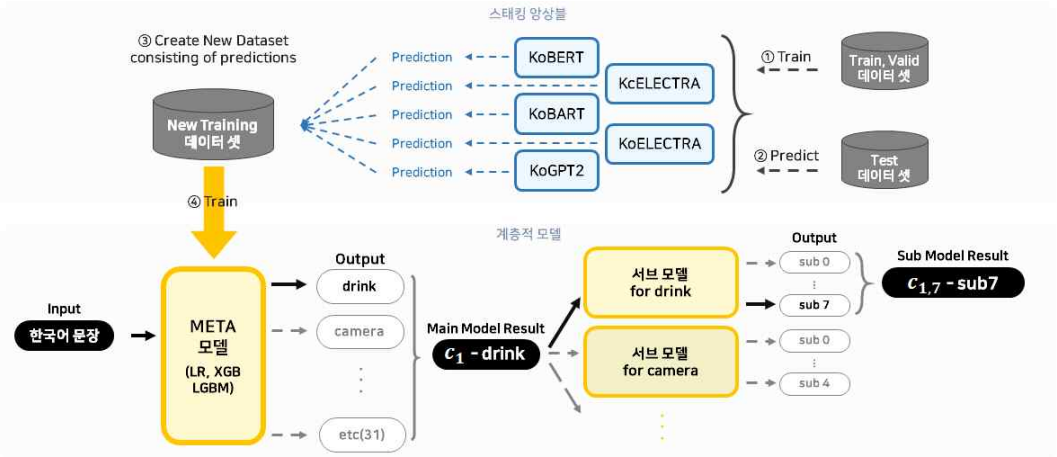
하였으며, 이들 중 가장 적합한 메타 모델을 선정하기 위한 실험을 수행하였다. 더 나아가, 스태킹 앙상블의 효과를 입증하기 위해 가장 널리 사용되는 보팅 앙상블과의 비교 실험을 병행하였다.

【표 5-3】Pseudocode: Constructing Hierarchical Model After Applying Stacking Ensemble to Main Model

Input : test data

Output : test predict label

```
###Main Model's Load and Predict
main_models = [KoBART(main), KoBERT(main), KoGPT2(main),
          KoELECTRA(main), KcELECTRA(main)]
main_predictions = [model.predict() for model in main_models]
###Main Stacking Ensemble
main_new_data = {'input': main_predictions, 'class': test_data['class'] }
main_meta_model = LogisticRegression() or XGB() or LGBM()
main_meta_model.train(main_new_data)
main_predict = main_meta_model.predict()
###Sub Model Load and Predict
sub_model = KoBART # KoBART, KoBERT, KoGPT2, KoELECTRA,
                   # KcELECTRA 중 연결할 하나의 서브모델 선택
sub_models = [sub_model(sub0), sub_model(sub1), ..., sub_model(sub31)]
sub_predictions = [model.predict() for model in sub_models]
###Model Predict Concatenation
correct = 0
for i in range(test_data.size()):
          main_label = main_predict[i]
          sub_label = sub_predictions[main_label][i]
          if main_label == main_true_label and sub_label == sub_true_label:
               correct += 1
```



【그림 5-6】 Hierarchical Model with Stacking Ensemble

Ⅵ. 실험 설계 및 결과

1. 성능지표

본 연구에서는 이모티콘 추천 시스템의 성능을 평가하기 위해 혼동행렬(Confusion Matrix)을 사용하였다([표 6-1] 참조). 이를 기반으로 정확도(Accuracy), 정밀도(Precision), 재현율(Recall), 그리고 F1-Score라는 총 4가지 평가 지표를 도출하였으며, 해당 계산법은 [수식 6-1]부터 [수식 6-4]에서 확인할 수 있다. 이러한 다양한 지표를 활용하는 이유는 각각이 모델의 성능을 서로 다른 측면에서 측정할 수 있기 때문이다.

【표 6-1】 Confusion Matrix

		predicted	
		negative(0)	positive(1)
actual	negative(0)	TN	FP
	positive(1)	FN	TP

정확도는 모델이 문맥을 얼마나 정확하게 파악하고, 그에 따라 적절한 이모티콘을 예측하는 능력을 평가한다. 한편, 정밀도와 재현율은 모델이 예측을 얼마나 잘 수행하는지를 알아보는 지표로 활용된다. 정밀도는 클래스 A로 예측한 데이터 중 실제로 A인 데이터의 비율을, 재현율은 실제 A 클래스 데이터 중에서 모델이 A로 올바르게 예측한 비율을 나타낸다. 하지만, 이 두 지표는 trade-off 관계에 있어 하나가 높아지면 다른 하나는 낮아질

수 있다는 한계가 있다. 이를 보완하기 위해, 정밀도와 재현율을 합쳐 평균을 내는 F1-Score도 함께 고려한다.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

【수식 6-1】 Formula for Accuracy

$$\mathrm{Precision} = \frac{TP}{TP + FP}$$

【수식 6-2】 Formula for Precision

$$Recall = \frac{TP}{TP + FN}$$

【수식 6-3】 Formula for Recall

$$F1 - Score = 2 \times \frac{Recall \times \mathrm{Precision}}{Recall + \mathrm{Precision}}$$

【수식 6-4】 Formula for F1-Score

더불어, 본 연구에서는 Top-K 정확도([수식 6-5])라는 추가적인 평가 지표를 도입하였다. 이는 동일한 내용의 텍스트에서도 사용자의 개성에 따라 선택되는 이모티콘이 다를 수 있고, 문맥에 따라 여러 이모티콘이 사용될 수 있음을 고려하기 위함이다. 이렇게 총 5개의 성능지표를 활용함으로써, 이모티콘 추천 시스템의 성능을 다양한 측면에서 평가하고, 이를 통해 모델의 강점과 약점을 종합적으로 이해하고 분석할 수 있다.

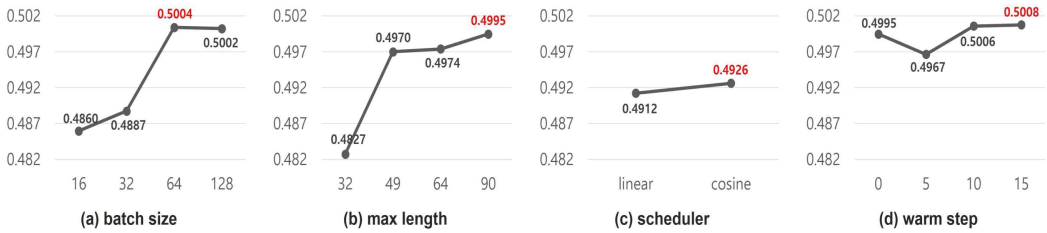
$$Top-K\ Accuracy(y, \hat{f}) = \frac{1}{n_{sample}} \sum_{i=0}^{n_{sample}-1} \sum_{j=1}^{k} 1\left(\widehat{f_{i,j}} = y_i\right)$$

【수식 6-5】 Formula for Top-k Accuracy

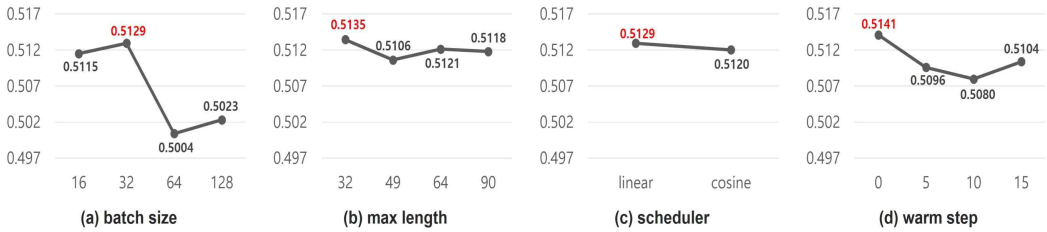
2. 하이퍼파라미터 실험

본 논문에서는 KoBERT, KoBART, KoGPT2, KcELECTRA, KoELECTRA 모델에 대하여 다양한 하이퍼파라미터 설정에 따른 성능 변화를 살펴보았다. 우리는 배치 크기(Batch size), 최대 길이(Max Length), 그리고 스케줄러(Scheduler) 및 웜업 단계(Warmup step)를 조정함으로써 각 모델의 성능을 최적화하려고 노력하였다.

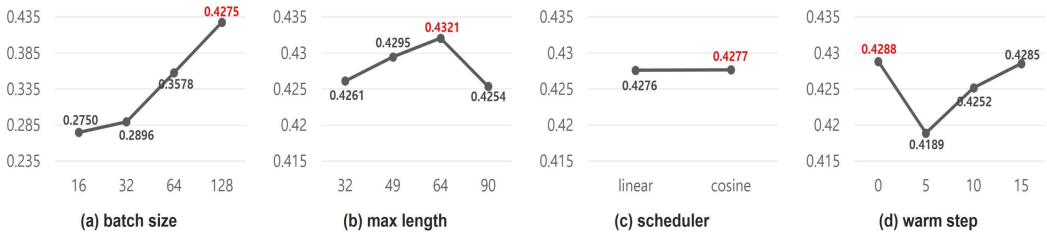
배치 크기는 학습 시간과 데이터 다양성 사이의 균형을 맞추는 역할을 한다. 작은 배치 크기는 다양한 데이터를 더 많이 처리하면서도 학습 시간이 증가하는 반면, 큰 배치 크기는 학습 시간을 단축시키지만, 데이터의 다양성을 제한할 수 있다. 최대 길이는 입력 데이터의 길이 제한을 정의하는데, 이를 초과하는 데이터는 잘려 정보 손실을 야기하게 된다. 그러나 너무 큰 최대 길이는 필요 이상의 메모리 사용으로 학습 효율성을 저하시킬 수 있다. 또한, 스케줄러와 웜업 단계는 학습률을 점진적으로 조정하는 중요한 요소로 작용한다. 웜업 단계에서는 학습률이 점진적으로 증가하여 초기의 불안정성을 줄이며, 스케줄러는 웜업 단계 이후 학습률의 변화를 관리한다. 이렇게 배치 크기, 최대 길이, 스케줄러 및 웜업 단계와 같은 주요 하이퍼파라미터들의 조정은 모델의 학습과정과 성능에 결정적인 영향을 미치며, 이에 대한 깊은 이해는 효과적인 모델 구현에 꼭 필요하다.



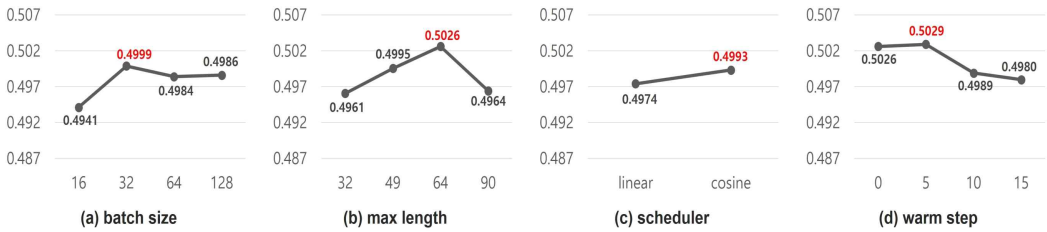
【그림 6-1】 KoBERT Hyperparameter Tuning Results



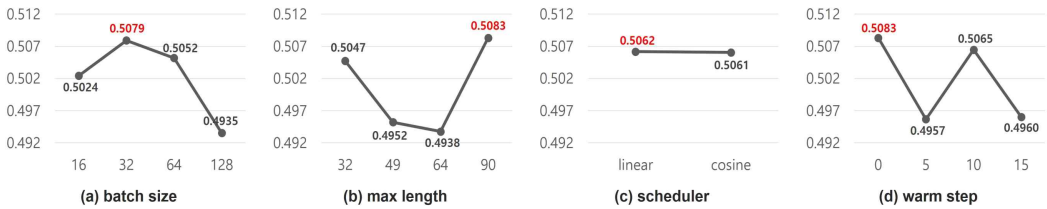
【그림 6-2】 KoBART Hyperparameter Tuning Results



【그림 6-3】 KoGPT2 Hyperparameter Tuning Results



【그림 6-4】 KcELECTRA Hyperparameter Tuning Results



【그림 6-5】 KoELECTRA Hyperparameter Tuning Results

하이퍼파라미터 실험 결과는 [그림 6-1]부터 [그림6-5]에 상세히 나와 있다. 본 실험을 통해, 우리는 각 모델별로 최적의 하이퍼파라미터를 선정하는데 성공하였다. 이런 방식으로, 우리는 각각의 모델에 대한 학습을 보다 효과적이고 정밀하게 진행할 수 있었다.

3. 모델 성능 실험 환경 및 설계

본 연구의 실험 환경은 [표 6-2]에 상세하게 기록되어 있다. 실험은 Intel i7-7700 CPU (4.2GHz), 64GB 메모리를 갖춘 시스템에서 수행되었으며, 이 시스템은 Ubuntu 16.04를 운영체제로 사용하고 있었다. 또한, 그래픽 처리 장치로는 NVIDIA GTX 1080 Ti GPU가 사용되었다. 실험에 사용된 모든 모델들은 PyTorch 프레임워크를 기반으로 구현되었다.

【표 6-2】 System Configuration

Computing Environment	workstation
Processor	Intel i7-7700, 4.2GHz CPU
Memory	64GB
Operating System	Ubuntu 16.01
Graphics Card	NVIDIA GTX 1080 Ti GPU

가장 먼저, 계층적 KoBERT 모델을 활용한 이모티콘 추천 시스템의 성능을 평가하기 위해, 비교 대상 모델을 선정하고 성능 비교 실험을 수행하였다. 이 실험에서는 계층적 구조를 가진 데이터셋을 이용해 계층적 KoBERT 모델을 학습시킨 후 추천 성능을 비교 분석하였다.

비교 대상 모델로는 텍스트 분류 분야에서 널리 사용되는 DNN, LSTM, Bi-LSTM, 그리고 GRU 모델을 선정하였으며, 모델 간 성능을 비교하기 위해 정확도(Accuracy), 정밀도(Precision), 재현율(Recall) 그리고 F1-Score를 측정하였다. 또한, 이모티콘의 사용 패턴이 사용자의 성향과 선호도에 따라 크게 달라질 수 있음을 고려하여 top-k accuracy를 추가로 적용하여 top-3, top-5 accuracy를 함께 측정하였다.

- DNN (Deep Neural Network): 가장 기본적인 신경망 중의 하나로서 여러 개의 은닉층을 가지며 비선형성을 이용하여 고차원의 복잡한 데이터셋에서 패턴을 찾아내는 데 사용된다.
- LSTM (Long Short-Term Memory): 이전 계산 결과에 관한 메모리를 도입하여 길이가 긴 시퀀스 정보를 기억하고 전파할 수 있는 신경망으로서 순차적인 데이터를 학습하는 데 장점을 가진다.
- Bi-LSTM (Bidirectional LSTM): Bi-LSTM은 양방향 LSTM으로 시간적 의존성이 있는 양방향 데이터를 처리하는 데 주로 사용된다. 순방향과 역방향의 양방향으로 입력 시퀀스를 처리하는 방식으로서 텍스트 분류 분야에서 널리 사용된다.
- GRU (Gated Recurrent Unit): 순환 신경망(RNN)의 일종으로서 LSTM과 유사하고 리셋 게이트와 업데이트 게이트의 상호작용을 통해 학습한다. LSTM보다 파라메터 수가 적어서 학습 속도가 빠른 장점을 가진다

다음으로, 스태킹 앙상블을 적용한 계층적 모델의 우수성을 입증하기 위해 한국어 전이학습 모델 기반의 단일 계층적 모델과 성능을 비교하는 실험을 진행하였다. 이에 따라, KoBERT를 포함한 KoBART, KoGPT2, KcELECTRA, 그리고 KoELECTRA를 계층적 구조로 훈련시킨 후, 이들과의 추천 성능을 비교 분석하였다.

이 과정에서는 계층적 구조를 가진 이모티콘 추천 데이터셋이 활용되었다. 특히, 가장 높은 성능을 보였던 단일 한국어 전이학습 모델의 서브모델을 앙상블 모델에 연결하여 효율적인 계층적 모델을 구축하였다.

이상적인 메타 모델을 선정하기 위해, Logistic Regression, XGB, 그리고 LGBM 모델의 성능을 대조적으로 평가하였다. 스태킹 앙상블의 효과를 입증하기 위해 소프트 보팅(Softvoting) 방법론을 활용하여 학습을 진행하고

그 결과를 비교 분석하였다. 그리고, 모델의 성능을 정밀하게 측정하기 위해 정확도(Accuracy), 정밀도(Precision), 재현율(Recall), F1-Score 및 Top-k 정확도와 같은 다양한 지표를 사용하였다.

4. 모델 성능 실험 결과 및 분석

4.1 계층적 KoBERT 모델 성능 평가

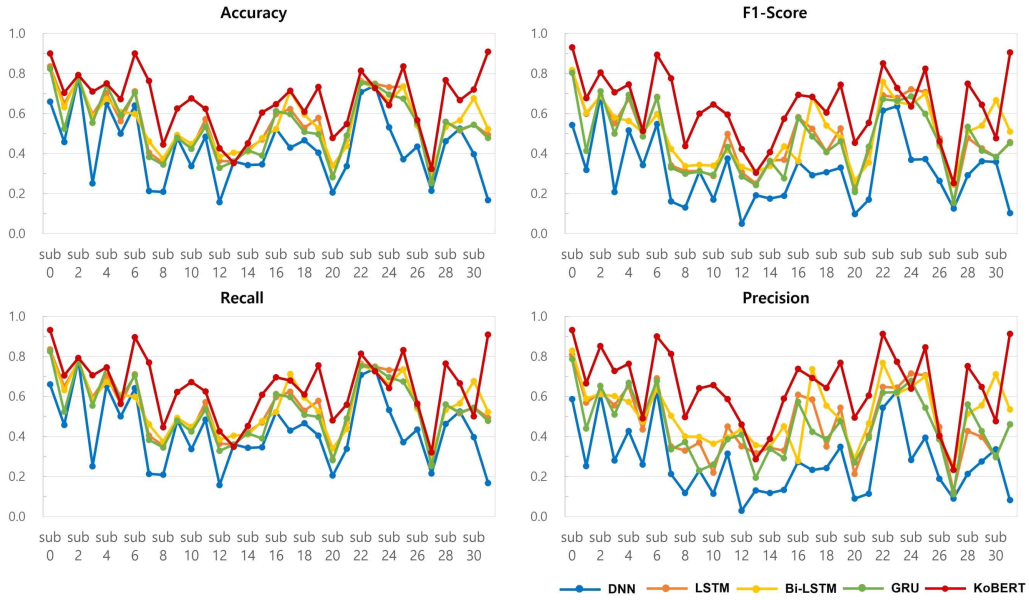
첫 번째 실험은 계층적 KoBERT 모델에서 메인모델의 성능을 측정한다. 메인 KoBERT 모델은 SNS 게시글을 이모티콘 메인 카테고리 32개 중 하나로 분류한다. 이를 위해 메인 KoBERT 모델과 비교군 모델인 DNN, LSTM, Bi-LSTM, GRU 4개 모델의 정확도, 정밀도, 재현율, 그리고 F1-Score를 [표 6-3]과 같이 비교하였다. 메인 KoBERT 모델의 정확도는 0.501의 성능을 보여주고 있으며, DNN 대비 0.286, LSTM 대비 0.259, Bi-LSTM 대비 0.24, 그리고 GRU 대비 0.263 정도 성능이 향상되었다. 메인모델의 정밀도는 0.526, 재현율은 0.501, F1-Score는 0.489로 모든 성능지표에서 다른 모델과 비교하여 높은 성능을 보여주고 있다.

또한, DNN, LSTM, Bi-LSTM, GRU와 같은 비교군 모델들이 상대적으로 간단한 아키텍처를 가지고 있어 동일한 양의 데이터를 학습하는데 있어서 더 적은 계산을 요구하므로 빠른 학습 시간을 보이는 반면, KoBERT 모델은 수백만 개의 매개변수를 가지고 있어 계산량이 크게 증가하며 이로 인해 학습 시간이 상대적으로 증가한 모습을 보이고 있다. 그러나 빠른 학습 시간이 반드시 우수한 성능을 보장하지는 않는다. 대량의 데이터로 사전 학습된 KoBERT는 풍부한 언어적 지식을 포착하여 복잡한 작업에 효과적으로 적용하였기 때문에 이러한 성능 향상이 가능하였다.

【표 6-3】 Comparison of Main model Performances
(DNN, LSTM, Bi-LSTM, GRU, KoBERT)

	Accuracy	Precision	Recall	F1-Score	Training-Time (sec)
DNN	0.218	0.090	0.218	0.101	284
LSTM	0.242	0.204	0.242	0.172	461
Bi-LSTM	0.261	0.178	0.245	0.173	713
GRU	0.238	0.200	0.238	0.161	445
KoBERT	**0.501**	**0.526**	**0.501**	**0.489**	**22,946**

두 번째 실험에서는 서브 KoBERT 모델의 성능을 평가하였다. 서브 KoBERT 모델은 메인 KoBERT 모델 하위에 있는 32개의 모델로서 서브 카테고리의 분류를 수행한다. [그림 6-6]는 32개의 서브모델별로 KoBERT 모델과 비교군 4개 모델(DNN, LSTM, Bi-LSTM, 그리고 GRU의 성능을 비교한 결과를 보여주고 있다. 각 서브 KoBERT 모델마다 훈련 데이터셋의 크기와 클래스 개수가 달라서 성능도 각기 다른 것을 확인할 수 있다. 중요한 점은 다른 모델들과 비교하여 서브 KoBERT 모델이 정확도를 포함한 모든 성능지표에서 높은 수치를 보였다.



【그림 6-6】 Comparison of Sub model Performances
(DNN, LSTM, Bi-LSTM, GRU, KoBERT)

세 번째 실험에서는 계층적 KoBERT 모델의 성능을 검증하기 위하여 메인모델과 서브모델을 연결하고 전체 테스트 데이터에 대해서 최종 이모티콘 추천 성능을 [표 6-4]와 같이 비교 분석하였다. 5개 모델의 최종 연결 정확도를 비교한 결과, KoBERT 모델의 Top-5 정확도 0.550의 성능을 보여주고 있으며, DNN 대비 0.395, LSTM 대비 0.324, Bi-LSTM 대비 0.316, 그리고 GRU 대비 0.374 정도 성능이 향상되었다. 정밀도는 0.581, 재현율은 0.451, F1-Score는 0.477로 모든 성능지표에서 다른 모델과 비교하여 높은 성능을 보여주고 있다. [표 6-5]는 LSTM 계열 모델과 KoBERT를 기반으로 한 계층적 모델들이 100개의 문장에 대해 이모티콘 추천을 실행하는데 필요한 평균 시간을 비교 분석한 결과를 보여준다. KoBERT 모델은 DNN에 비해 약 68ms 더 오래 걸리는 것으로 관찰되었으나, LSTM, Bi-LSTM, GRU와 비교했을 때는 약 10ms의 차이만 나타났다. 이러한 결과는 성능 향상을 고려

하면, 추론 시간의 상대적인 증가는 큰 부담이 아님을 보여준다. 특히, KoBERT 모델의 효율적인 연산 능력을 고려할 때, 이 모델이 가져오는 성능 향상이 추론 시간에 대한 심각한 추가 부담을 일으키지 않음을 알 수 있다.

【표 6-4】 Performance of Hierarchical KoBERT Model compared to different models

	Accuracy @T1	Accuracy @T3	Accuracy @T5	Precision	Recall	F1-Score
DNN	0.086	0.134	0.155	0.060	0.180	0.080
LSTM	0.104	0.180	0.226	0.169	0.184	0.133
Bi-LSTM	0.107	0.182	0.234	0.136	0.197	0.136
GRU	0.089	0.147	0.176	0.072	0.119	0.082
KoBERT	0.415	0.502	0.550	0.581	0.451	0.477

【표 6-5】 Comparison of Recommendation Time per Sentence between the Hierarchical KoBERT Model and Different Models

	Recommendation Time (milli-sec)
DNN	172.30
LSTM	226.06
Bi-LSTM	227.89
GRU	227.89
KoBERT	239.78

4.2 스태킹 앙상블을 적용한 계층적 모델 성능 평가

첫 번째 실험은 계층적 모델 내의 메인모델의 성능을 비교 분석한다. 이 메인모델의 역할은 인스타그램 게시글을 32개의 메인 카테고리 중 하나로 분류하는 것으로 KoBERT, KoBART, KoGPT2, KcELECTRA, KoELECTRA 기반의 메인모델의 성능을 점검하였다.

한국어 전이학습 모델을 기반으로 한 5개 메인모델의 정확도, 정밀도, 재현율, 그리고 F1-Score를 비교 분석하였다([표 6-6] 참조). 그중에서도 0.514로 정확도가 가장 높았던 KoBART 모델은 기존의 KoBERT와 KoGPT2의 한계를 해소하기 위해 두 모델의 아키텍처를 혁신적으로 결합한 결과로, 모든 성능지표에서 탁월한 성과를 보여주었다. 또한, 비정형화된 데이터인 뉴스 기사 댓글을 활용하고, 잘못된 단어를 올바른 단어로 치환하는 치환 학습방식을 적용한 KcELECTRA 모델은 0.496으로 F1-Score 지표에서 뛰어난 결과를 보여주었다. 반면, 단방향 모델인 KoGPT2는 다음 단어를 예측하는 방식을 기반으로 학습하므로, 텍스트 분류 태스크에서 다른 전이학습 모델들보다 상대적으로 부족한 성능을 보여주었다. 또한, 수백만 개에서 수십억 개에 이르는 매개변수를 가지고 있는 한국어 전이학습 모델은 계산량이 크게 증가하였다. 이에 따라, 모델의 학습 시간이 3시간 이상으로 상당히 증가한 모습을 보인다. 특히, KoBERT 모델은 약 6시간의 긴 학습 시간을 보여주는데 이는 하이퍼파라미터 설정에서 Warm Step이 15로 선정되어 초반에 학습률이 느리게 상승하는 특성 때문이다.

【표 6-6】 Comparison of Main model Performances
(KoBERT, KoBART, KoGPT2, KcELECTRA, KoELECTA)

	Accuracy	Precision	Recall	F1-Score	Training-Time (sec)
KoBERT	0.501	0.526	0.501	0.489	**22,946**
KoBART	**0.514**	0.529	**0.514**	0.495	12,751
KoGPT2	0.432	0.443	0.432	0.390	12,494
KcELECTRA	0.503	**0.534**	0.503	**0.496**	15,099
KoELECTRA	0.508	0.516	0.508	0.484	15,919

한국어 전이학습 모델에 소프트 보팅과 스태킹 앙상블을 적용하여 성능을 비교 분석하였으며, 그 결과는 [표 6-7]에서 확인할 수 있다. LGBM 모델을 이용한 스태킹 앙상블 모델은 가장 높은 정확도를 보인 KoBART 모델과 소프트 보팅 모델과 비교했을 때 0.15 이상의 정확도 향상을 보였다. 또한,

F1-Score 성능이 좋았던 KcELECTRA에 비해 약 0.18의 F1-Score를 상승시키는 결과를 보였다. 반면에, 선형 분류 모델인 Logistic Regression을 메타 모델로 사용한 스태킹 앙상블은 성능이 떨어졌다. 이는 이모티콘 추천과 같은 복잡한 문맥 분석이 필요한 문제에는 비선형 문제 해결 능력이 뛰어난 모델이 더 적합하다는 사실을 잘 보여주고 있다. 이러한 결과는 메타 모델 선택의 중요성을 강조한다. 복잡한 문맥 정보를 잘 반영할 수 있는 모델이 메타 모델로 선정될 때, 스태킹 앙상블 방법이 더 높은 성능을 보일 수 있다는 것이다. 이에 따라 본 연구에서는 최종적으로 LGBM을 메타 모델로 선정하였다.

학습 시간에 대해 논의할 때, 소프트 보팅 방식은 별도의 추가 모델 학습을 요구하지 않지만, 스태킹 앙상블의 경우 메타 모델을 새로운 데이터 세트로 학습하는 데 추가 시간이 필요하다는 점을 고려해야 한다. 본 표에서 제시된 학습 시간은 5개의 기본 모델을 학습시킨 후에 추가적으로 소요된 시간을 의미한다. 가장 높은 성능을 보인 LGBM 모델은 XGB에 비해 적은 학습 시간을 소비하면서도 큰 성능 차이를 나타내지 않는다. 또한, Logistic Regression과 비교하여도 비슷한 학습 시간을 보이면서도 성능 차이가 크지 않다. 이러한 결과는 각 모델의 학습 효율성에 대해 통찰력을 제공한다.

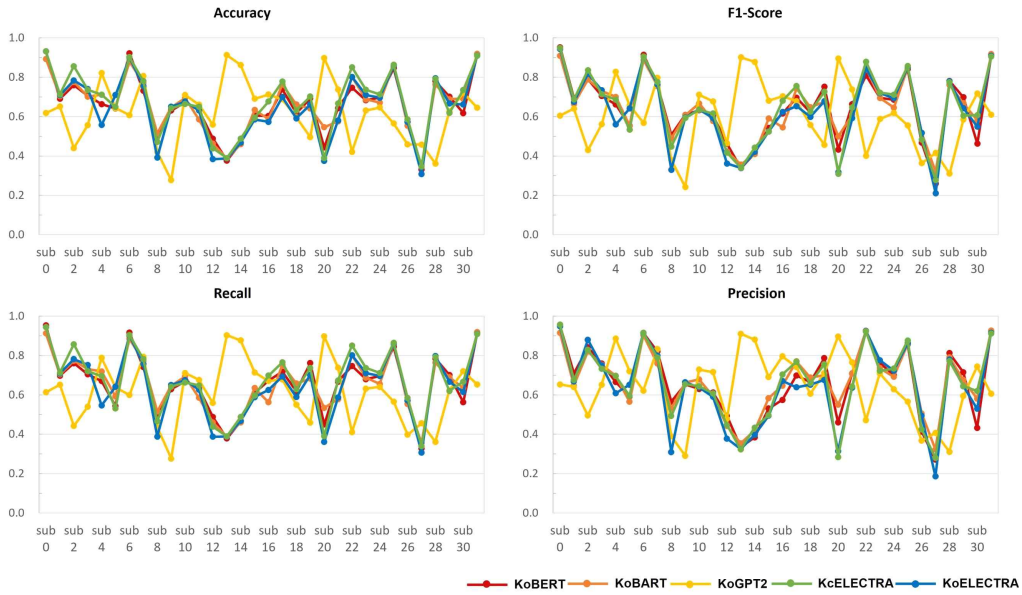
【표 6-7】 Comparison of Main model Performances
(Softvoting, Stacking-LR, Stacking-XGB, Stacking-LGBM)

	Accuracy	Precision	Recall	F1-Score	Training-Time (sec)
SoftVoting	0.521	0.579	0.521	0.510	-
Stacking (meta : lr)	0.315	0.146	0.315	0.183	953
Stacking (meta : xgb)	0.658	0.694	0.658	0.675	971
Stacking (meta : lgbm)	**0.674**	**0.706**	**0.674**	**0.689**	958

두 번째 실험에서는 서브모델들의 성능을 비교 분석하였다. 서브모델들은

메인모델 하위에서 서브 카테고리로 재분류하는 역할을 담당한다. 본 실험은 KoBERT, KoBART, KoGPT2, KcELECTRA, 그리고 KoELECTRA 기반의 서브모델들의 성능을 조사하며, 이를 통해 어떤 모델이 앙상블 모델과 가장 적합한 연결을 형성할 수 있는지를 평가하였다.

서브모델들의 성능은 정확도, 정밀도, 재현율, 그리고 F1-Score와 같은 기준을 통해 비교 분석되었다. [그림 6-7]를 참조하면, 32개의 서브모델들은 각각의 훈련 데이터셋 크기와 클래스 개수가 다르므로, 각자의 성능이 다양한 패턴을 보이는 것을 확인할 수 있다. 이러한 패턴에도 불구하고, 대부분의 서브모델들이 메인모델과 유사한 성능을 보였으며, 특히 KoBART 모델과 KcELECTRA 모델은 뛰어난 성능을 보였다. KoGPT2 모델의 성능 분포에 대해서는 별도로 주목할 필요가 있다. KoGPT2 모델은 생성적인 학습방식을 통해 사전 훈련된 모델이기 때문에, 다른 모델들과는 약간 다른 성능 패턴을 보이는 것으로 관찰되었다. 이러한 차이는 모델의 학습방식과 구조의 차이로 인해 발생하며, 이는 우리의 연구에서 중요한 관찰 결과로 해석된다. 이를 종합하여, KoBART, KcELECTRA, KoGPT2를 기반으로 한 서브모델을 앙상블 모델과 연결하였다.



【그림 6-7】 Comparison of Sub model Performances
(KoBERT, KoBART, KoGPT2, KcELECTRA, KoELECTA)

세 번째 실험은 계층적 모델의 성능을 철저히 비교 분석하는 것이다. 이를 위해, 앞선 실험에서 훈련된 메인모델과 서브모델들을 조합하여 실제 상황에서의 성능을 살펴본다. 전체 테스트 데이터를 통해 이러한 계층적 모델의 성능을 평가한 결과는 [표 6-8]에서 확인할 수 있다. 분석 결과를 살펴보면, 단일 계층적 모델에 비해 LGBM(Light Gradient Boosting Machine)을 메타 모델로 적용한 스태킹 앙상블 기법이 도입된 계층적 모델이 더욱 탁월한 성능을 보여주었다는 결론을 도출할 수 있었다. 이는 스태킹 앙상블이 메인 카테고리의 고유한 특징을 분명하게 구분해내는 능력을 향상시켜, 기존의 넓은 범위의 카테고리 구분 문제를 해결한 결과이다. 스태킹 앙상블을 적용한 계층적 모델 중에서는 KoBART 기반의 서브모델이 가장 뛰어난 성능을 보였다. [표 6-9]는 [표 6-8]에서 비교한 다양한 계층적 모델들이 100개의 문장에 대해 이모티콘 추천을 수행하는 데 필요한 평균 시간을 비교 분석한 결과를 제시한다. 주목할 결과 중 하나는, 앙상블 기법을 적용하였을

때 높은 성능 향상을 기록하였으나, 이에 따른 평균 추론 시간은 약 3배 이상 증가하였다는 것이다. 이 결과는 앙상블 기법이 성능 향상에 기여하는 동시에, 그로 인해 추가적인 계산 비용이 요구됨을 보여준다.

【표 6-8】 Performance of Ensemble-Applied Hierarchical Model compared to different models

	Accuracy @T1	Accuracy @T3	Accuracy @T5	F1 -Score	Precision	Recall
KoBERT	0.415	0.502	0.550	0.477	0.581	0.451
KoBART	0.430	0.520	0.564	0.503	0.644	0.457
KoGPT2	0.355	0.445	0.492	0.420	0.565	0.386
KcELECTRA	0.420	0.509	0.555	0.486	0.588	0.453
KoELECTRA	0.417	0.506	0.549	0.484	0.619	0.454
Stacking(meta:lgbm) + KoBART Sub	**0.502**	**0.584**	**0.616**	**0.556**	**0.654**	**0.524**
Stacking(meta:lgbm) + KcELECTRA Sub	0.499	0.584	0.615	0.554	0.653	0.521
Stacking(meta:lgbm) + KoGPT2 Sub	0.486	0.564	0.594	0.539	0.632	0.508

【표 6-9】 Comparison of Recommendation Time per Sentence between the Ensemble-Applied Hierarchical Model and Different Models

	Recommendation Time (milli-sec)
KoBERT	239.78
KoBART	269.98
KoGPT2	200.71
KcELECTRA	242.08
KoELECTRA	245.06
Stacking(meta:lgbm) + KoBART Sub	703.15
Stacking(meta:lgbm) + KcELECTRA Sub	676.73
Stacking(meta:lgbm) + KoGPT2 Sub	661.42

Ⅷ. 결론

본 논문은 인스타그램 SNS 플랫폼에서 사용자가 작성한 게시글의 문맥을 분석하여 게시글의 내용을 보완하거나 강조하는 이모티콘을 자동 추천하는 시스템을 제안하였다. 인스타그램 게시글을 수집하여 172,149개의 자체 데이터셋을 구성하고 게시글에서 추출한 616여 개의 방대하고 종류도 매우 다양한 이모티콘을 대상으로 추천한 부분에 의의가 있다. 이 문제는 게시글의 문맥을 분석하여 문맥 기반으로 이모티콘을 추천하는 것이고, 이 연구에서는 이 문제를 이모티콘 카테고리를 기준으로 게시글을 분류하는 문제로 정형화하였다. 특히, 616개나 되는 대량의 이모티콘을 대상으로 추천이 가능한 원인은 클래스가 많은 텍스트 분류 문제를 해결하기 위해서 계층적 방법으로 접근하였기 때문이다. 계층적 접근 방법은 다양한 감정과 주제를 포함하며 상/하위 카테고리가 존재하는 유니코드 이모티콘의 특징을 그대로 반영한다는 점에서 의미가 크다. 314개의 이모티콘 카테고리를 계층적으로 생성하고 계층적 KoBERT 모델을 도입, 학습시켜 SNS 게시글을 32개의 메인 카테고리와 314개의 서브 카테고리로 분류하였다. 해당 방법은 대량의 이모티콘을 대상으로 추천하는 데 적합하며 게시글의 문맥이나 내용을 반영하는 다양한 이모티콘을 추천하는 데 유용하게 활용될 수 있다.

본 연구에서 구축한 계층적 KoBERT 모델은 Top-5 정확도 0.550의 이모티콘 추천 성능을 제시하고 있다. 계층적 KoBERT 모델의 성능 검증을 위하여 DNN, LSTM, Bi-LSTM, GRU의 4개 모델을 비교군으로 선정하여 성능 비교 실험을 진행하였다. 메인모델, 서브모델, 그리고 계층적 모델에 대해 정확도, 정밀도, 재현율, F1-Score를 각각 측정하였다. 실험 결과, 다른 비교군 모델 대비, 계층적 KoBERT 모델은 모든 성능지표에서 높은 성능을

보여주고 있다.

또한, [그림 7-1]과 같이 임의의 게시글을 입력으로 이모티콘을 추천하는 추론 실험을 진행하고 임의의 게시글에 추천된 이모티콘을 살펴본 결과, 계층적 KoBERT 모델이 LSTM 모델보다 텍스트 문맥 분석 및 파악 능력이 우수한 것을 확인할 수 있었다. 이는 BERT 모델이 양방향으로 자연어를 처리하는 언어모델이고, 특히 KoBERT 모델의 경우 한국어의 불규칙한 언어 변화의 특성을 반영하기 위해 개선된 모델로서 기존 LSTM 계열의 모델보다 텍스트 문맥이나 내용 분석에 유리하기 때문이다.

이어서, KoBART, KoGPT2, KcELECTRA, KoELECTRA와 같은 한국어 전이학습 모델을 추가 도입하여 학습을 진행하였다. KoBART, KcELECTRA, KoELECTRA 모델이 KoBERT 모델에 비해 정확도 기준에서 약 0.01의 성능 향상을 보였음을 확인하였다. 이를 통해, 인스타그램과 같은 단문 텍스트의 문맥을 파악하고 계층적으로 분류하는 작업에 대하여 KoBART, KoELECTRA, KcELECTRA 모델이 더 효과적임으로 입증되었다. 본 연구에서는 메인모델의 상대적으로 낮은 성능을 개선하기 위해 스태킹 앙상블 방법을 적용하였다. 다양한 메타 모델을 활용하고, 소프트 보팅과 비교하기 위한 실험을 수행하였다. LGBM을 메타 모델로 활용한 스태킹 앙상블 모델은 기존 단일 메인모델의 정확도인 0.5를 능가하는 0.674의 정확도를 보여주었다. 이는 확실히 메인모델의 성능을 향상시킨 결과라 할 수 있다. 또한, 우수한 성능을 보인 KoBART, KoGPT2, KcELECTRA 모델을 연결하여 구성한 스태킹 앙상블을 적용한 계층적 모델은 Top-5 정확도 0.616을 달성하였다. 그러나, 모델 추천 시간이 3배 이상 증가하는 한계가 존재했다. 이러한 결과는 스태킹 앙상블의 효율적인 구현과 계산 비용 간의 균형을 찾아야 함을 나타낸다. 이에 따라, 향후 연구에서는 이러한 한계를 극복하고 성능을 더욱 향상시키는 방법에 대한 탐색이 필요함을 보여준다.

(a) LSTM 추천 결과 **(b) KoBERT 추천 결과**

【그림 7-1】 Inference Experiment for Emoji Recommendation

기존의 이모티콘 추천 연구들에서 30~100개의 소규모 이모티콘 카테고리를 추천하는 성능이 0.4을 다소 상회하는 정도인 것을 기준으로 볼 때, 본 논문에서 제안하는 시스템은 이모티콘 추천 성능을 향상시켰다고 할 수 있다. 현재, 이모티콘 추천 정확도는 0.616 정도의 성능을 보이고 있는데, 이는 이모티콘 사용 형태가 개인의 연령, 취향이나 감성에 의존적이기 때문이다. 이에 관한 향후 연구 방향으로는, 사용자의 이모티콘 사용 패턴과 특성에 대한 깊은 분석을 통해 개별 사용자의 성향을 더 잘 반영하는 추천 알고리즘 개발이 필요하다. 또한, 계산 효율성과 성능 사이의 균형을 잘 유지하는 방식으로 추천 시스템을 최적화하여 성능 향상을 추구할 계획이다.

참 고 문 헌

1) Choi, Ji-Eun, "The Influence of Motivation and Usage Patterns of Emoticons on Social Capital Formation in SNS," Management Studies, Vol.32, No.3, pp.1-20, 2017

2) Henning Pohl, Dennis Stanke, and Michael Rohs, “EmojiZoom: emoji entry via large overview maps,” Association for Computing Machinery, pp.510 - 517, 2016.

3) Eun Ji Lee, “Motivations for the Using Emoticon : Exploring the Effect of Motivations and Intimacies between Users on the Attitude and Behaviors of Using Emoticon,” Journal of the HCI Society of Korea, Vol. 12, No. 2, pp. 5-12, 2017.

4) Young Il Hong, Sun Kyung Yim, “The Effect of Emoticon Expression Type on User Satisfaction Factors,” The Treatise on The Plastic Media, Vol. 25, No.1, pp. 33-41, 2022.

5) V. N. Durga, Pavithra Kollipara, V. N. Hemanth Kollipara, and M. Prakash, “Emoji Prediction from Twitter Data using Deep Learning Approach,” Asian Conference on Innovation in Technology (ASIANCON), 2021.

6) Xuanzhi Zheng, Guoshuai Zhao,Li Zhu, Xueming Qian, “PERD: Personalized Emoji Recommendation with Dynamic User Preference,” Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 1922–1926, July 2022.

7) Seongmin Lee, Eunseo Lee, and Daeyoung Park, “Emoticon Recommendation using Emotional Analysis,” Proceedings of Symposium of the Korean Institute of communications and Information Sciences, pp. 864-865. 2021.
8) Luda hao and Connie Zeng, “Using Neural Networks to Predict Emoji Usage from Twitter Data,” Computer Science, 2017.
9) Kazuyuki Matsumoto, Minoru Yoshida, and K. Kita, “Classification of Emoji Categories from Tweet Based on Deep Neural Networks,” Proceedings of the 2nd International Conference on Natural Language Processing and Information Retrieval(NLPIR), pp 17-25, Sep. 2018.
10) J Shobana, S Amudha, and S Kumar, “Emoji Anticipation and Prediction Using Deep Neural Network Model,” International Conference on Power, Energy, Control and Transmission Systems (ICPECTS), 2022.
11) Ruobing Xie, Zhiyuan Liu, Rui Yan and Maosong Sun, “Neural Emoji Recommendation in Dialogue Systems,” 2016.
12) Joon Gyum Kim, Tae Sik Gong, Bo Goan Kim, Jae Yeon Park, Woo Jeong Kim, Evey Huang, Kyung Sik Han, Ju Ho Kim, Jeong Gil Ko, and Sung Ju Lee, “No More One Liners: Bringing Context into Emoji Recommendations,” ACM Transactions on Social Computing, Vol. 3. pp. 1-25, 2020.
13) Gael Guibon, Magalie Ochs, and Patrice Bellot, “Emoji Recommendation in Private Instant Messages,” Proceedings of the 33rd Annual ACM Symposium on Applied Computing(SAC), pp. 1821-1823. 2018.

14) B. Felbo, A. Mislove, A. Søgaard, I. Rahwan, and S. Lehmann, “Using Millions of Emoji Occurrences to Learn Any-Domain Representations for Detecting Sentiment, Emotion and Sarcasm,” in Proceedings of the Conference on Empirical Methods in Natural Language Processing, pp. 1615–1625, 2017.
15) Tomihira T., Otsuka A., Yamashita A., and Satoh T., “Multilingual Emoji Prediction using BERT for Sentiment Analysis,” International Journal of Web Information System, Vol. 16, No. 3, pp. 265-280, 2020.
16) Peijun Zhao, Jia Jia, Yongsheng An, Jie Liang, Lexing Xie, and Jiebo Luo, “Analyzing and Predicting Emoji Usages in Social Media,” WWW '18: Companion Proceedings of the The Web Conference, pp. 327-334, 2018.
17) Guoshuai Zhao, Zhidan Liu, Yulu Chao, and Xueming Qian, “CAPER: Context-Aware Personalized Emoji Recommendation,” IEEE Transactions on Knowledge and Data Engineering, pp. 1-1, 2020.
18) JeeHyun Kim, YeRim Kim, HaeWon Byun, “SNS Context-based Emoji Recommendation using Hierarchical KoBERT,” Journal of Digital Contents Society,24(6), 2023.6.
19) Vaswani, Ashish, et al, "Attention is all you need," Advances in neural information processing systems 30, 2017.
20) Devlin, Jacob, et al, "Bert: Pre-training of deep bidirectional transformers for language understanding," 2018.
21) SKTBrain KoBERT, GitHub repository, https://github.com/SKTBrain/KoBERT, 2019.

22) AIOPEN ETRI, AI API·DATA, https://aiopen.etri.re.kr/bertModel, 2019.
23) Radford, Alec, et al, "Language models are unsupervised multitask learners," 2019.
24) Radford, Alec, et al, "Improving language understanding by generative pre-training," 2018.
25) Perez, Luis, Lizi Ottens, and Sudharshan Viswanathan, "Automatic Code Generation using Pre-Trained Language Models," 2021.
26) SKT-AI KoGPT2, Github repository, https://github.com/SKT-AI/KoGPT2, 2020.
27) Lewis, Mike, et al, "Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension," 2019.
28) SKT-AI KoBART, Github repository, https://github.com/SKT-AI/KoBART, 2020.
29) Clark, Kevin, et al, "Electra: Pre-training text encoders as discriminators rather than generators," 2020.
30) monologg KoELECTRA, GitHub repository, https://github.com/monologg/KoELECTRA, 2022.
31) Beomi KcELECTRA, GitHub repository, https://github.com/Beomi/KcELECTRA, 2022.
32) Ji Hye Heo, Su Bin I, Won Hyuk Yang, Dong Hoon Lim, “Transfer learning-based ensemble deep learning for image classification of COVID-19 patients,” Journal of the Korean Data And Information Science Society, 32(6), pp. 1219-1235, 2021.

33) Lee Shin Haeng, “Machine Learning for Detecting Malicious Comments on YouTube: Focusing on the Application of Stacking Ensemble Model,” Journal of The Korean Data Analysis Society, 24(4), pp. 1583-1598, 2022.
34) Min-Ki Kim, “Automatic Fruit Grading Using Stacking Ensemble Model Based on Visual and Physical Features,” Journal of Korea Multimedia Society, 25(10), pp. 1386-1394, 2022.
35) Myung-woo Nam, Young-Jin Choi, Hoe-Ryeon Choi, Hong-Chul Lee, “Parallel Network Model of Abnormal Respiratory Sound Classification with Stacking Ensemble,“ Journal of the Korea Society of Computer and Information, 26(11), pp. 21-31, 2021.
36) Reimers, Nils and Iryna Gurevych, “Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks,” Conference on Empirical Methods in Natural Language Processing, 2019.

ABSTRACT

Context aware emoji recommendation for SNS based on emsemble pre-training model

Kim Jee-Hyun
Department of
Convergence Technology Engineering
Graduate School of
Sungshin University

Emoticon recommendation is a critical task that assists users in easily finding the appropriate emoticon among thousands available. Traditional emoticon recommendation methods target chat platforms and primarily recommend emoticons that users frequently use to express emotions. However, on social media platforms like Instagram, emoticons are often used more to supplement or emphasize the content of short posts rather than to convey emotions.

This study proposes a new methodology for recommending emoticons by understanding the context of Korean posts on social media platforms.

We introduce hierarchical KoBERT into the emoticon recommendation problem to understand the context of Korean posts and recommend a variety of suitable emoticons. Recommending 616 emoticons from 314 categories proves useful in conveying the implicit meanings of short social media posts more accurately.

We construct a dataset that reflects the real world by collecting Instagram posts and build a hierarchical KoBERT model to learn the hierarchical categories of emoticons inserted in each text. Experimental results have validated that the hierarchical KoBERT model outperforms DNN, LSTM, Bi-LSTM, and GRU models in emoticon recommendation.

Additionally, for performance improvement, we introduce additional Korean transfer learning models such as KoBART, KoGPT2, KoELECTRA, KcELECTRA, and compare their performances and apply stacking ensemble techniques.