



저작자표시-비영리-변경금지 2.0 대한민국

이용자는 아래의 조건을 따르는 경우에 한하여 자유롭게

- 이 저작물을 복제, 배포, 전송, 전시, 공연 및 방송할 수 있습니다.

다음과 같은 조건을 따라야 합니다:



저작자표시. 귀하는 원저작자를 표시하여야 합니다.



비영리. 귀하는 이 저작물을 영리 목적으로 이용할 수 없습니다.



변경금지. 귀하는 이 저작물을 개작, 변형 또는 가공할 수 없습니다.

- 귀하는, 이 저작물의 재이용이나 배포의 경우, 이 저작물에 적용된 이용허락조건을 명확하게 나타내어야 합니다.
- 저작권자로부터 별도의 허가를 받으면 이러한 조건들은 적용되지 않습니다.

저작권법에 따른 이용자의 권리는 위의 내용에 의하여 영향을 받지 않습니다.

이것은 [이용허락규약\(Legal Code\)](#)을 이해하기 쉽게 요약한 것입니다.

[Disclaimer](#)

2022년 8월
박사학위 논문

대화 문맥의 연관성을 적용한 멀티 레이블 감정인식 모델

조선대학교 대학원

컴퓨터공학과

임 명 진

대화 문맥의 연관성을 적용한 멀티 레이블 감정인식 모델

Multi-Label Emotion Recognition Model applying
Correlation of Conversation Context

2022년 8월 26일

조선대학교 대학원

컴퓨터공학과

임 명 진

대화 문맥의 연관성을 적용한 멀티 레이블 감정인식 모델

지도교수 김 판 구
공동지도교수 신 주 현

이 논문을 박사학위신청 논문으로 제출함.

2022년 4월

조선대학교 대학원

컴퓨터공학과

임 명 진

임명진의 박사학위논문을 인준함

위원장	조선대학교	교수	양희덕 (인)
위원	조선대학교	교수	김판구 (인)
위원	조선대학교	교수	신주현 (인)
위원	조선대학교	교수	최준호 (인)
위원	U S T	교수	황명권 (인)

2022년 6월

조선대학교 대학원

목 차

ABSTRACT

I. 서론	1
A. 연구 배경 및 목적	1
B. 연구 내용 및 구성	5
II. 관련 연구	7
A. 감정인식	7
B. 비정형 대화 텍스트	12
C. 텍스트 감정인식	15
D. 멀티 레이블 감정인식	18
III. 멀티 레이블 감정인식 모델	21
A. 모델 구성도	21
B. 데이터 전처리	23
1. 비정형 대화 텍스트 변환	23
2. 형태소 분석	26
C. 멀티 레이블 감정인식	28
1. 문장 임베딩	28
2. 멀티 레이블 감정인식	32

IV. 대화 문맥 연관성을 적용한 MLER 모델	42
A. 모델 구성도	42
B. 대화문 멀티 레이블 감정인식	44
1. 대화문 분류	44
2. 대화문 감정 예측	51
C. 대화 문맥 연관성	53
1. 대화 문맥 연관성 분석	53
2. 대화 문맥 연관성 적용	57
V. 실험 및 평가	61
A. 실험 데이터	61
B. 실험 평가 및 분석	64
1. 실험 평가 방법	64
2. 실험 결과 분석	67
VI. 결론 및 제언	77
참고문헌	79

표 목 차

[표 2-1] POMS(Profile of mood States) 이론	8
[표 2-2] Ekman의 감정을 나타내는 표정	10
[표 2-3] 본 논문에서 사용한 7가지 감정	11
[표 2-4] 비정형 대화 텍스트의 유형	13
[표 2-5] 비정형 대화 텍스트 분석 기법	14
[표 3-1] 비정형 대화 텍스트 변환 방법	23
[표 3-2] Algorithm 1 : Unstructured conversation text processing	25
[표 3-3] 형태소 분석기 비교	26
[표 3-4] Algorithm 2 : Word2vec embedding	29
[표 3-5] KoBERT embedding	31
[표 3-6] LSTM 모델 구조	32
[표 3-7] CNN-LSTM 모델 구조	33
[표 3-8] Algorithm 3 : Attention model	35
[표 3-9] Attention 모델 구조	36
[표 3-10] Transformer 모델 구조	38
[표 3-11] 입력 문장 예시	39
[표 3-12] 멀티 레이블 감정인식 결과	39
[표 4-1] 대화 분류 과정	44
[표 4-2] Algorithm 4 : Conversation classification : SL	45
[표 4-3] Algorithm 5 : Conversation classification : 1-turn CL	46
[표 4-4] Algorithm 6 : Conversation classification : multi-turn CL	47
[표 4-5] 대화문 분류 과정	48
[표 4-6] 대화 문장 예시	49
[표 4-7] 대화문 분류 결과	49
[표 4-8] 멀티 레이블 감정인식	51
[표 4-9] 대화문의 멀티 레이블 감정인식	51
[표 4-10] 상관계수에 따른 상관도	53
[표 4-11] 상관관계 분석 과정	54
[표 4-12] 강한 상관관계 순위	56

[표 4-13] 전체 대화 문맥의 연관성	57
[표 4-14] 대화 문맥 가중치	58
[표 4-15] Algorithm 7 : Apply <i>corrCC</i>	60
[표 5-1] 개발 환경	61
[표 5-2] 감정 클래스별 데이터의 수	63
[표 5-3] Confusion Matrix of Emotion Label e	64
[표 5-4] Confusion Matrix 용어 설명	65
[표 5-5] Epoch에 따른 MLER 성능	68
[표 5-6] 임베딩&데이터에 따른 MLER 성능	69
[표 5-7] 모델별 MLER 성능	70
[표 5-8] 감정 클래스별 MLER 성능	71
[표 5-9] 모델별 멀티 레이블 분류 비교	73
[표 5-10] 감정과 문장의 유사도 비교	74
[표 5-11] 대화 문장 예시	76
[표 5-12] 감정인식 결과 비교	76

그림 목 차

[그림 1-1] 전체 구성도	5
[그림 2-1] Plutchik의 감정 모델	8
[그림 2-2] Thayer 모델	9
[그림 2-3] 데이터 속성에 따른 특징	20
[그림 3-1] 멀티 레이블 감정인식 모델 구성도	21
[그림 3-2] Attention 모델	37
[그림 3-3] ‘Neutral’ 문장의 멀티 레이블 감정인식 예시	40
[그림 3-4] ‘Surprise’ 문장의 멀티 레이블 감정인식 예시	41
[그림 4-1] 대화 문맥의 연관성을 적용한 MLER 모델 구성도	42
[그림 4-2] 대화 데이터 예시	48
[그림 4-3] 대화문의 멀티 레이블 감정인식 그래프	52
[그림 4-4] 상관관계 Heat map	55
[그림 4-5] 양의 상관관계와 음의 상관관계	56
[그림 4-6] 문장 순서별 대화 문맥 가중치 그래프	58
[그림 5-1] Single Dataset	62
[그림 5-2] Continuous Dataset	62
[그림 5-3] 데이터 셋에 따른 감정 클래스의 비율	63
[그림 5-4] Epoch에 따른 MLER 성능 그래프	68
[그림 5-5] 임베딩&데이터에 따른 MLER 성능 그래프	69
[그림 5-6] 모델별 MLER 성능 그래프	71
[그림 5-7] 감정 클래스별 MLER 성능 그래프	72
[그림 5-8] 모델별 멀티 레이블 분류 비교 그래프	73
[그림 5-9] 감정과 문장의 유사도 비교 그래프	75
[그림 5-10] 대화 예시	75

ABSTRACT

Multi-Label Emotion Recognition Model applying Correlation of Conversation Context

MyungJin Lim

Advisor : Prof. PanKoo Kim, Ph.D.

Co-Advisor : Prof. JuHyun Shin, Ph.D.

Department of Computer Engineering,
Graduate School of Chosun University

Recently, due to COVID-19 (Coronavirus Disease 2019), non-face-to-face services have rapidly spread. As a result, the number of users communicating through messengers or SNS (Social Networking Service) increases. Research on analyzing user information or opinions and recognizing emotions using a large amount of data generated at this time is being actively conducted.

Conversation texts such as SNS, which users freely write, have unstructured features without a structured format. Due to these characteristics, analysis using AI (Artificial Intelligence) is difficult and the performance of emotion recognition techniques is degraded. Therefore, a processing method suitable for the characteristics of the unstructured conversation text is required. Among unstructured conversation texts, most Korean emotion recognition recognizes a single emotion by analyzing emotion keywords or vocabulary. However, a study on multi-label emotion recognition is needed because multiple emotions exist in a complex way in one sentence.

In addition, since the conversation proceeds by exchanging the conversation between the speakers, one's emotions may change depending on the words of others, and the emotions once generated may last for a certain period of time. And emotions are not only revealed through vocabulary, but also indirectly through conversation content. Therefore, it is necessary to effectively reflect the various factors that induce emotions and the previous sentences in the conversation context.

Therefore, in this paper, we analyze and process the characteristics of unstructured conversation text for more accurate emotion recognition. And implement a multi-label emotion recognition (MLER) model that understands the meaning of conversation and recognizes inherent and complex emotions. Conversation sentences are classified according to their characteristics, and the MLER model is applied to the conversation to derive the conversation context correlation and apply it.

As a method to verify the usefulness of the proposed technique, we experiment and evaluate the unstructured conversation text processing method, the multi-label emotion recognition (MLER) model, and the MLER model that applies the correlation of the conversation context. As a result, performance improved when processing unstructured conversation text and balancing emotion classes. Also, when using KoBERT sentence embedding and Attention model, Accuracy showed the best performance with 78.4%. In addition, as a result of evaluating multi-label performance, Micro-F1 of the proposed MLER model had the best performance at 74.8%. Finally, as a result of applying the conversation context correlation to the proposed model, it was confirmed that the emotion similarity was higher. Therefore, using the proposed multi-label emotion recognition model to which conversation context correlation is applied can contribute to improving the accuracy and performance of conversation emotion recognition.

I. 서론

A. 연구 배경 및 목적

최근 COVID-19(Coronavirus Disease 2019)로 인한 비대면 서비스의 증가로 트위터(Twitter), 페이스북(Facebook) 등 소셜 네트워크 서비스(Social Network Service : SNS)를 통한 정보의 생성과 공유가 활발하게 이루어지고 있고, 이를 통해 비대면으로 소통하는 경우가 점점 늘어나고 있다. 이때 사용자에게 의해 발생한 대량의 데이터를 활용하여 의미 있는 정보를 추출하고 사용자 정보나 의견을 분석하는 연구가 활발하게 진행되고 있다[1,2]. 그중에서도 감정인식은 사회적 이슈나 다양한 개인적인 의견을 분석하여 감정을 인식하는 연구 분야이다[3,4]. 또한 감성 컴퓨팅(Affective Computing)은 사람의 감정을 컴퓨터를 통해 이해하기 위한 분야로써 새로운 방법을 사용하여 감정을 소통하고 이해하며 응답하는 학문으로 많은 연구가 진행되고 있다.

감정은 어떤 사건이나 일에 대한 느낌이나 마음으로, 관련된 다양한 연구가 수행되었다. 감정인식은 사람의 사고방식과 행동양식을 구분하기 위한 중요한 요소로써 비즈니스(Business), 정치, 사회 복지 등 다양한 분야에 활용되고 있다. 또한 감정인식의 품질과 정확률(Accuracy)을 높이기 위한 감정을 분류하는 방법에 관련된 많은 연구가 수행되어왔다[5]. 비대면 대화에서 텍스트 데이터(Text data)는 인간이 경험하는 다양한 감정을 더욱 정교하게 표현해주는 가장 발달된 감정표현 수단이다. 텍스트로 표현하여 어떤 감정을 전달하려는 가를 인식하는 것은 기계가 인간의 감정을 읽어내는 데 꼭 필요하다. 인공지능(Artificial Intelligence : AI)이 텍스트 표현에 포함된 감정 반응을 정확하게 파악하기 위해서는 인간의 감정표현 언어에 관한 풍부하고 정교한 데이터베이스(Database)를 갖는 것이 필수적이다. 그리고 인간이 사회적 상황에서 경험하는 감정들은 기본 감정보다 훨씬 복잡한 감정이 존재하기 때문에 더 다양한 감정 분류 방법이 필요하다[6]. 텍스트 데이터에서 사용자의 감정을 인식하는 것은 다양한 방법으로 사용자의 의도를 분석할 수 있게 하고 관점을 이해하는 데 도움이 된다[7].

인터넷에서 사용되는 대부분의 비대면 대화 데이터는 정해진 형식이 없는 형태인 비정형 텍스트(Unstructured text)로 구성된다. 이러한 비정형 대화 텍스트의 특징을 살펴보면 많은 신조어와 비속어가 빈번하게 사용되고 있고 단어를 자유자재로 변형한 줄임말의 사용도 증가하고 있다[8]. 이러한 비정형 대화 텍스트의 사용은 주 사용층 외의 사용자들은 단어와 의미를 이해하는 데 어려움이 있어서 의사소통 문제가 발생하기도 한다. 또한 사전적인 의미를 지니지 않아 감정인식 알고리즘(Algorithm)의 성능 저하와 제약상황이 발생한다[9]. 따라서 신조어, 비속어, 줄임말과 같은 단어가 포함된 비정형 대화 텍스트의 특징을 잘 파악하여 처리해야 한다. 그중에서도 특히 한국어는 영어와 달라 분석하는 데 어려움이 있고 단어의 순서와 전체 문장에 대한 의미 파악이 중요하다. 어절만 사용하면 형태론적 변형이 다양하게 발생하는 한국어 대화의 특징에 따라 데이터 희소 문제가 발생할 수 있다. 따라서 언어에 맞는 비정형 대화 텍스트의 특징을 잘 이해하고 이에 특화된 처리 방법이 필요하다.

텍스트를 기반으로 한 감정 분류 연구를 살펴보면, 주로 어휘를 통해 감정을 인식하거나 감정 사전 또는 다량의 Uni-gram을 활용하여 단어나 문장 단위의 감정을 분석하고 있다. 그러나 이러한 분류에서는 양질의 학습 데이터와 충분한 학습 데이터의 양, 그리고 데이터의 균형을 맞추는 것이 중요하다[10]. 또한 딥러닝(Deep Learning) 모델을 사용한 감정 분류 연구로는 합성곱 신경망(Convolutional Neural Networks : CNN) 모델과 장단기 메모리(Long Short-Term Memory : LSTM) 모델, 그리고 두 모델을 앙상블(Ensemble) 한 CNN-LSTM을 사용하여 감정 분류 성능을 향상하려는 시도가 계속되었다[5,11,12]. 그러나 모델을 개선하는 방법보다는 데이터를 개선하여 품질을 향상하는 연구가 성능 면에서 더 효과적이라는 연구 결과가 있었다[13].

복합적인 특징을 가지는 감정은 한 문장 내에서도 다양한 감정을 포함할 수 있으므로, 단일 레이블(Single-Label) 데이터 셋보다는 멀티 레이블(Multi-Label) 데이터 셋이 더 효과적이라는 주장이 제기되었다[5]. 대화 데이터의 멀티 레이블 감정인식은 사용자의 다양한 감정을 이해하는 기능을 제공하는데 중요한 작업이다. 멀티 레이블과 관련된 연구로는 MEISD(Multimodal Multi-Label Emotion, Intensity and Sentiment Dialogue dataset)를 활용하여 순환 신경망(Recurrent

Neural Network : RNN)을 변형한 DialogueRNN(Attentive RNN in Conversations)을 기반으로 다른 입력 보다 텍스트의 멀티 레이블 감정인식의 결과가 더 정확한 것을 증명하였다[14]. 또한 SemEval-2018(Semantic Evaluation-2018)을 활용하여 AttnConvnet(Attention-based Convolutional Neural Networks)을 기반으로 문장의 멀티 레이블 감정을 인식하였다[15]. 그리고 트윗에 콘텐츠 기반 방법을 적용하여 멀티 레이블 감정을 분류하였다[16]. 다양한 방법으로 멀티 레이블 감정을 인식하려는 시도가 계속되고 있으나 대부분 영어 데이터 셋을 활용한 연구로 진행되었다. 한국어 대화 데이터 셋은 대부분 단일 레이블로 분류되어 있고 멀티 레이블에 관한 데이터 셋과 연구는 부족한 실정이다. 따라서 한국어 대화 텍스트에 내재한 감정을 인식하는 멀티 레이블 감정 분류가 필요하다. 또한 한국어의 특성을 파악하여 대화의 의미를 이해하고 문장에 포함된 다양한 감정을 인식하는 방법이 필요하다.

대화 데이터는 일반 텍스트와 다르게 화자(Speaker) 간에 대화문을 주고받으며 이야기가 진행된다. 비대면 대화의 경우에는 보내는 사람과 받는 사람 간에 다른 의미로 해석되는 경우가 있어 뜻하지 않은 오해가 생길 수 있다. 그리고 보낸 사람이 의도한 상황이 받는 사람에게는 제대로 전달되지 않을 때도 있다. 화자 간에 오해 없는 대화를 위해서 투명한 감정 교류는 중요한 요소이다. 화자들의 감정은 다른 사람의 말에 따라 감정이 바뀌기도 하고, 한번 발생한 감정이 일정 시간 동안 지속되기도 한다. 그리고 직접적인 어휘를 통해 감정을 드러내기도 하지만 간접적으로 대화 내용을 통해 나타나기도 한다[5]. 따라서 대화의 감정인식은 다양한 감정 어휘뿐 아니라, 대화 문맥 속에서 나타나는 감정을 유발하는 다양한 요소들이 반영되어야 하며 현재 문장뿐 아니라 이전 문장들을 효과적으로 반영해야 한다.

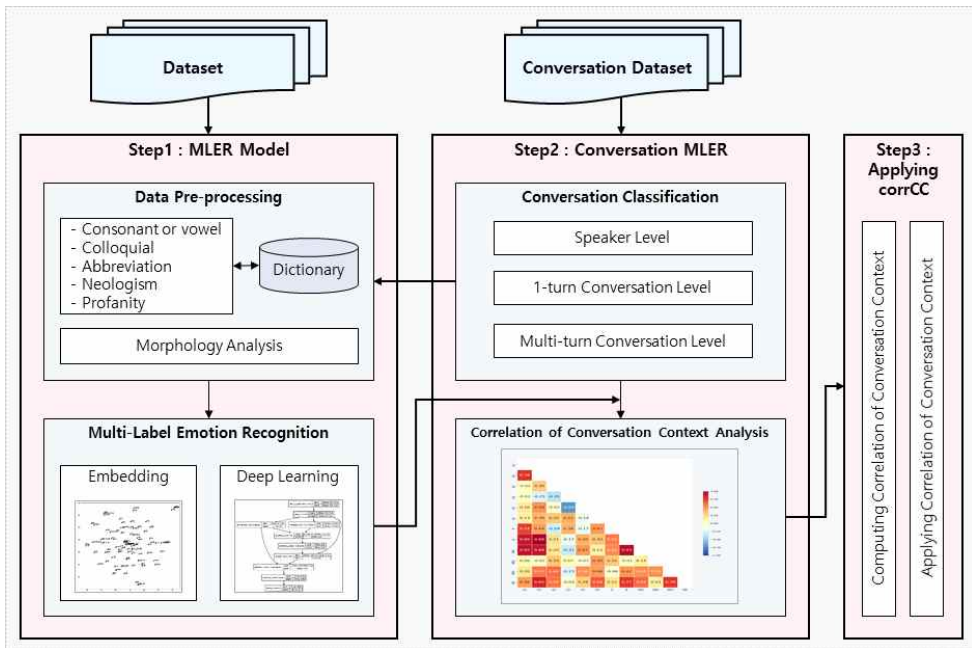
대화형 인터페이스를 위한 감정인식 또한 텍스트로 구성된 문장만으로 감정을 예측하는 것은 어려우므로 화자의 의도를 파악하기 위한 다양한 분석이 시도되었다. 화자는 사전적 의미를 대화에 포함하여 의도를 전달하고 현재 감정 상태에 따라서 다른 의도를 표현하거나 같은 의미가 있는 문장에서도 다양한 의도를 표현한다. 예를 들면, 같은 의미를 나타내는 문장이어도 대화의 흐름에 따라 다른 감정을 나타내는 경우도 있고, 비꼬는 방식의 문장인 경우에는 의미에 따라 나타나는 감정과 상반되는 감정을 나타낸다. 따라서 화자의 이전 감정과 상대방 화자의 감정 간

에 연관성을 분석하고 대화 문맥을 잘 파악하도록 하는 방법이 필요하다. 대화의 전후 상황을 이해하고 감정 상태가 어떻게 변화하는지 인식해야 한다. 대화에서 화자의 감정을 파악하는 것은 다양한 방향으로 의도를 분석할 수 있게 한다.

본 논문에서는 비대면 대화에서 비정형 대화 텍스트의 사용으로 인한 감정인식의 어려움과 단일 감정인식으로 인한 문제를 해결하기 위한 다음과 같은 방법을 제안한다. 대화에서 감정인식의 성능향상을 위해 비정형 대화 텍스트를 처리하고 내재한 다양한 감정을 인식하는 멀티 레이블 감정인식(Multi-Label Emotion Recognition : MLER) 모델을 학습한 다음, 대화문에 MLER 모델을 적용하여 인식한 감정을 통해 대화문 간의 감정 연관성을 도출하고 이를 적용하는 대화 문맥의 연관성을 적용한 멀티 레이블 감정인식 모델을 제안하고자 한다.

B. 연구 내용 및 구성

본 논문에서는 비대면 대화에서 감정인식의 어려움과 단일 감정인식에 대한 한계를 개선하기 위해 비정형 대화 텍스트를 처리하고 멀티 레이블 감정인식(Multi-Label Emotion Recognition : MLER) 모델을 적용하여 문장에 내재한 다양한 감정을 인식한다. 그리고 대화 데이터의 특성에 맞게 SL(Speaker Level), 1-turn CL(Conversation Level), Multi-turn CL 대화문으로 분류하고 대화문에 MLER 모델을 적용하여 멀티 레이블 감정을 인식한다. 감정인식의 결과 분석을 통해 대화 문맥의 연관성을 도출하고 이를 적용한 멀티 레이블 감정인식 모델을 제안한다. [그림 1-1]은 본 연구의 전체 구성도를 나타낸다.



[그림 1-1] 전체 구성도

[그림 1-1]과 같이 본 연구는 대화형 인터페이스에서 감정인식의 성능향상을 위한 기초 연구로써 비정형 대화 텍스트를 처리한 다음 문장의 멀티 레이블 감정인식 단계와 이를 적용한 대화문의 멀티 레이블 감정인식 단계, 그리고 감정인식 결과 분석을 통한 대화 문맥의 연관성 도출 단계와 이를 적용하는 단계로 구성된다.

본 논문의 구성은 다음과 같다.

본 장인 서론에 이어 2장에서는 본 연구의 이론적 배경인 감정인식, 비정형 대화 텍스트의 개념과 이와 관련된 연구를 자세히 살펴본다. 그리고 연구 진행에 필요한 텍스트 감정인식에 관한 연구와 멀티 레이블 감정인식에 관한 연구를 제시하여 본 논문에서 제안하는 연구 내용의 이해를 돕는다.

3장에서는 데이터를 전처리하는 방법과 멀티 레이블 감정인식(MLER) 모델을 제안한다. 비정형 대화 텍스트의 유형을 파악하고 처리하는 과정을 기술한다. 또한 더 정확한 감정인식을 위해 문장을 임베딩하고 4가지 딥러닝 모델을 활용하여 문장에 내재한 다양한 감정을 인식하는 MLER 모델에 관해 기술한다.

4장에서는 대화 문맥의 연관성을 적용한 멀티 레이블 감정인식(MLER) 모델을 제안한다. 대화의 감정인식 성능향상을 위해 대화 데이터의 특성에 맞게 SL, 1-turn CL, Multi-turn CL 대화문으로 분류하고 MLER 모델을 적용하여 대화문의 멀티 레이블 감정을 인식한다. 대화문 간의 감정인식 결과 분석을 통해 대화 문맥의 연관성을 도출하고 이를 적용한 MLER 모델에 관해 기술한다.

5장에서는 실험 데이터를 이용하여 본 논문에서 제안하는 멀티 레이블 감정인식(MLER) 모델과 대화 문맥의 연관성을 적용한 MLER 모델을 실험 및 평가하고, 마지막으로 6장에서는 결론과 향후 연구에 관해 기술하고 마무리한다.

II. 관련 연구

본 장에서는 제안하는 연구의 이론적 배경이 되는 감정인식 모델과 비정형 대화 텍스트에 관련된 연구를 기술한다. 그리고 연구 진행에 필요한 텍스트 감정인식과 멀티 레이블 감정인식에 관한 연구에 관해 기술한다.

A. 감정인식

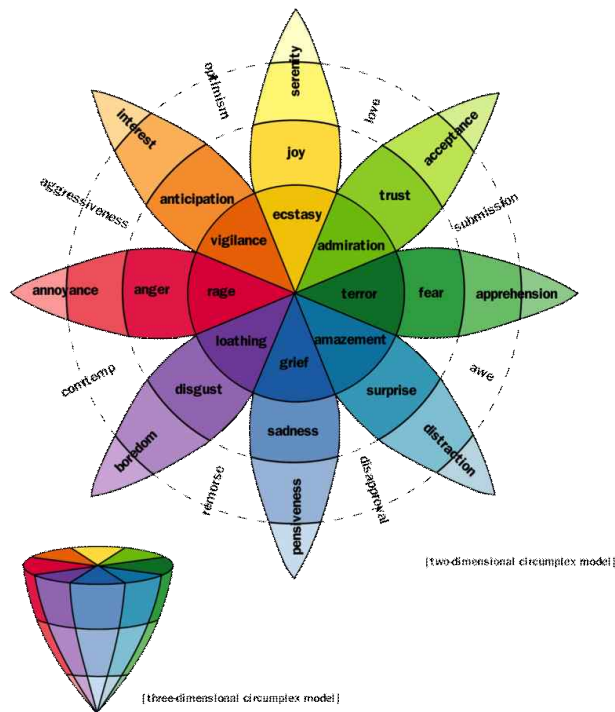
영어로 감정을 뜻하는 ‘Emotion’은 움직임(motion)과 밖으로(e)라는 라틴어에서 비롯되었다. 즉 움직여 나아가는 경향이라고 볼 수 있으므로 감정은 동작과 관련이 있다. 학자들은 감정에 대해 ‘감정이란 사물에 대한 적합성을 판단하여 접근하거나 회피하려는 느낌의 경향으로서 그 느낌은 신체적 반응으로 보강된다. 감정은 주변 환경에서 벌어지는 사건에 대한 심리적, 생리적 반응이다.’라고 정의하기도 한다 [17,18]. 감정은 사람과 사람 사이의 대화에 있어서 매우 중요한 역할을 한다. 감정이 없으면 인간은 의사결정을 내리는 데 문제가 생기며, 이는 인간의 이성과 감정이 분리되는 게 아니라는 의미이다. 사람의 감정을 컴퓨터를 통해 이해하기 위한 분야로써 감성 컴퓨팅(Affective Computing)이 연구되고 있다. 감성 컴퓨팅은 새로운 방법을 사용하여 감정을 소통하고 이해하며 응답하는 학문이다.

감정인식은 문서를 구성하는 문장에서 사용자들의 감정을 인식하는 연구이다. 기존 감정인식 연구에서는 주로 감정 단어를 긍정과 부정으로 분류하는 극성 분류 중심이었지만, 감정 단어마다 극성을 부여하여 다양한 감정으로 분류하는 연구가 활발하게 진행되고 있다. 심리학에서 정의하는 감정에 대한 분류 체계로 다양한 이론[19-21]이 있다. [19]의 기분상태척도(Profile of mood States : POMS) 이론은 총 65개의 형용사로 구성된 단어를 바탕으로 요인을 분석하고 설문을 통해 평가하는 방법이다. [표 2-1]은 POMS 이론의 요인 분석을 나타낸다.

[표 2-1] POMS(Profile of mood States) 이론[22]

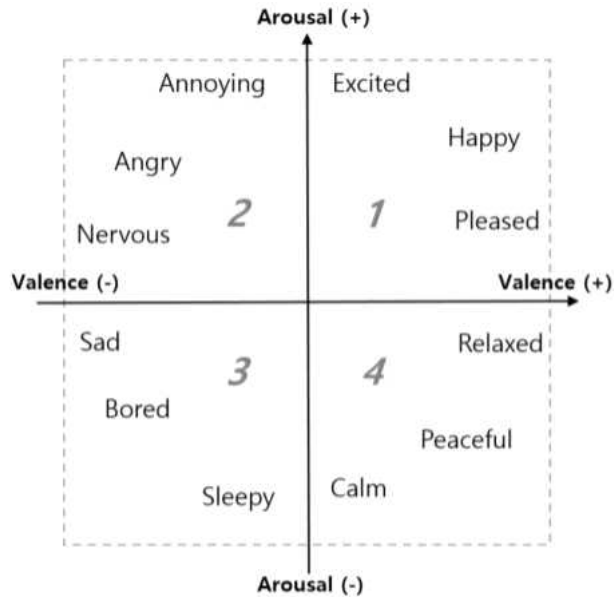
POMS factor analysis	
tesion - anxiety	fatigue - inertia
confusion - bewilderment	angry - hostility
vigor - activity	depression - dejection

Plutchik은 인간과 동물에게 있어 나타나는 공통적인 기본 감정을 8가지(분노, 공포, 슬픔, 혐오, 놀람, 기대, 신뢰, 기쁨)로 정의하고 인간의 감정을 원반상에 감정 바퀴라고 표현했다[20]. [그림 2-1]은 Plutchik의 감정 모델을 나타낸다. 감정의 세기에 따라서 원반상에서 안쪽으로 갈수록 강한 감정들이 배열되고 바깥쪽으로 갈수록 약한 감정들이 배열된다. 또한 마주 보고 있는 감정은 서로 반대되는 감정을 나타내고 강도에 따라 세분화하여 나누어지며 기본 감정들이 혼합된 여러 조합 감정을 이룬다.



[그림 2-1] Plutchik 감정 모델

Thayer는 감정을 생물 심리학적인 관점으로 분류한 모델을 제안하였다. [그림 2-2]는 Thayer 모델을 나타낸다. 2차원의 공간에 가로축은 긍정과 부정의 정도에 따른 Valence 축이고, 세로축은 감성의 강도를 나타내는 Arousal 축을 나타내고 총 12가지 감정들이 분류되어 있다[21]. 또한 감정을 연구하는 분야 외에 다양한 연구 분야에서도 채택하여 사용하고 있다[23].



[그림 2-2] Thayer 모델

Ekman은 인간의 얼굴 표정과 감정에 관한 연구를 통해 9가지 특징을 분석하여 인간의 보편적인 6가지(분노, 공포, 슬픔, 기쁨, 혐오, 놀람) 감정을 분류하였다[24]. 이후 확장 연구를 통해 인간의 감정을 7가지 감정(분노, 공포, 슬픔, 행복, 혐오, 놀람, 경멸)으로 분류할 수 있다고 제안했다[25]. Harmon-Jones 등은 이산 설문 조사를 통해 8가지 감정(분노, 혐오, 공포, 불안, 슬픔, 행복, 편안함, 욕망)을 포착했으며[26], Cowen과 Keltner는 27가지 감정을 분류하였다[27]. [표 2-2]는 Ekman이 제안한 6가지 감정을 나타내는 미세한 얼굴 표정을 정리한 것이다.

[표 2-2] Ekman의 감정을 나타내는 표정[28]

Emotion	Expression
행복(happy)	광대가 올라가고 입술 모서리는 뒤로 당겨 올라간다. 아랫눈꺼풀 아래 피부의 주름이 지고, 코와 윗입술 사이와 눈가 바깥 부분에도 주름이 진다.
혐오(dislike)	윗입술이 들린다. 일반적으로 비대칭의 코끝과 윗입술 부근의 주름 이마에 주름이 잡힌다. 광대가 올라가 아래 눈꺼풀에 주름이 생긴다.
분노(angry)	광대가 올라가고 입술 모서리는 뒤로 당겨 올라간다. 아랫눈꺼풀 아래 피부의 주름이 지고, 코와 윗입술 사이와 눈가 바깥 부분에도 주름이 진다.
공포(fear)	눈썹 높이가 높아지고 수축하며 윗눈꺼풀과 아랫눈꺼풀이 올라간다. 입술이 가늘다. 때로는 입이 벌어지기도 한다.
놀람(surprise)	눈썹 높이가 올라가고, 동그란 모양으로 바뀐다. 눈썹 아래 피부가 퍼진다. 눈이 확장된다(위는 올라가고 아래는 내려간다). 턱이 벌어진다.
슬픔(sad)	눈의 바깥쪽이 아래로 비스듬히 기울어진다. 삼각형 모양의 눈썹 피부 입술이 아래로 향하면 입술이 흔들릴 수도 있다.

이처럼 감정을 어떻게 분류할 것인지에 관한 연구가 계속되었고, 다양한 감정인식 방법이 제안되었다. 본 논문에서는 감정인식 방법 중에서 Ekman이 제안한 6가지 감정인 ‘행복(Happy)’, ‘분노(Angry)’, ‘공포(Fear)’, ‘놀람(Surprise)’, ‘슬픔(Sad)’, ‘혐오(Dislike)’를 사용한다. 또한 ‘중립(Neutral)’을 포함하여 7가지 감정을 인식하는 모델을 구현하였다. [표 2-3]은 본 논문에서 사용한 7가지 감정의 설명과 예시 문장을 나타낸 것이다.

[표 2-3] 본 논문에서 사용한 7가지 감정

Emotion	Explanation	Example
중립(Neutral)	감정이 드러나지 않음.	이거 한번 볼래?
놀람(Surprise)	놀라운 느낌.	믿을 수가 없어!
분노(Angry)	분개하여 몹시 성을 냄. 또는 그렇게 내는 성.	여기는 동네 단골을 이런 식으로 대하나?
슬픔(Sad)	슬픈 마음이나 느낌.	나 진짜 너무 속상해.
행복(Happy)	생활에서 충분한 만족과 기쁨을 느끼고 흐뭇함. 또는 그러한 상태	오늘 정말 기분이 좋아.
혐오(Dislike)	증오, 불결함 등의 이유로 싫어하거나 기피하는 감정으로, 불쾌, 기피함, 싫어함 등의 감정이 복합적으로 이루어진 비교적 강한 감정.	아이고, 잘도 그러겠다.
공포(Fear)	두렵고 무서움.	이러다가 영영 안 되는 거 아닐까 걱정되네요.

B. 비정형 대화 텍스트

비정형(Unstructured)이란 비구조화 또는 비구조적이라고도 하며 미리 정의된 모델이 없거나 미리 정의된 방식으로 정리되지 않은 형식을 말한다. 1998년 메릴린치는 잠재적으로 이용 가능한 모든 비즈니스 정보 중 약 80~90% 정도가 비정형에서 기원한 것으로 보는 ‘경험 법칙’을 언급하였다. 이 법칙은 1차 연구나 양적 연구에 근간을 두지 않지만, 일부는 받아들여지고 있다[29]. 정해진 형식이 없는 비정형 데이터는 대부분 텍스트 중심으로 되어 있으나 날짜, 숫자, 사실과 같은 데이터도 포함할 수 있다. 이로써 변칙과 모호함이 발생하므로 데이터베이스의 칸 형식(Form)에 저장되거나 문서에 주석화된(의미에 따라 태그된) 데이터에 비해 전통적인 프로그램을 사용하여 이해하는 것을 불가능하게 만든다[30].

비정형 데이터 중 비정형 대화 텍스트(Unstructured conversation text)는 데이터의 폭발적인 증가에 대응하기 위한 처리 기술의 발전과 함께 새롭게 조명되고 있다. 인터넷에서 대화에 사용되는 텍스트 데이터는 정해진 형식이 없고 새로운 용어는 인터넷 내에서는 물론이고 상당수가 현실에서도 사용된다. 이러한 텍스트들은 대부분 유행에 민감하고 빠르게 확산하였다가 식어버리는 특징이 존재하기 때문에 단어의 의미 외에 구조나 탄생에 대해서는 의의가 필요하지 않다. 따라서 일반적인 형태소 분석기로는 새롭게 등장한 단어 즉 신조어에 대해서 형태소 분석이 이루어지지 않기 때문에 신조어와 그 의미를 매칭(Matching)하는 사전을 구축하여 그 뜻을 파악한 이후에 감정을 분류해야 한다.

비정형 대화 텍스트 중 신조어는 품사로써 의미가 없는 단어이고 형태가 분명치 않기 때문에 기존에 사용하는 형태소 분석으로는 신조어를 추출할 수 없다. 따라서 문장 내에서 공백이나 키워드를 기준으로 나누기 위해 토큰나이징(Tokenizing) 기법을 이용하고, 품사 태깅을 진행하지 않는다. 이후 신조어를 판별하기 위해 의미가 담긴 품사와 뒤에 붙는 (보)조사, 숫자, 특수문자를 제거한다. 다음으로 불용어 제거가 완료된 텍스트 데이터들을 구축한 사전에서 의미를 추출하고 학습 및 감정 분석을 진행한다.

비정형 대화 텍스트는 기호만 사용하거나 신조어, 비속어 등의 언어 파괴적인 특징이 있다. 규격화되지 않았으므로 분석할 수 없고 의미를 읽어낼 수 없는 데이터로 판단할 수 있다. 이로 인해 문장을 품사별, 최소 단위로 분해하는 형태소 분석기가 올바르게 작동하지 못하게 되고 감정인식에 혼란을 가져와 모델의 성능이 저하된다. 따라서 비정형 대화 텍스트에 대한 유형을 파악하고 별도의 처리 작업이 필요하다. [표 2-4]는 비정형 대화 텍스트의 유형을 나타낸다.

[표 2-4] 비정형 대화 텍스트의 유형[31]

구분	의미	예문
기호 사용	자음·모음, 이모티콘, 숫자의 패턴을 사용한 의미 전달	πππ, ^^;;, 8282
음운 변이	음운을 추가하거나 교체	알취(알지), 시러(싫어)
형태 통사	문장에 형태 통사적 변화를 가미함	여러분 진심으로 환영(합니다)
신조어	사회 현상을 반영해 새로 생긴 말	역세권(역으로 인하여 그 세력이 미치는 권역)
비속어	통속적으로 쓰는 저속한 말	쌩까다(모른척하고 무시)
의성어·의태어	소리나 모양을 나타내는 말	헤헤(쓱스러운 듯이 웃는 모습)

비정형 대화 텍스트의 분석 기법은 정형 데이터 분석 기법인 데이터 마이닝(Data Mining)의 한계를 극복하기 위해 꾸준히 발전해 왔다. 데이터 마이닝은 정제된 데이터베이스에서 의미를 찾아낸다는 점에서 합리적인 분석 기법으로 보이지만, 다음과 같은 한계가 있다[32]. 먼저 데이터베이스의 정보는 데이터 관리자에 따라 정제된 데이터이므로, 상식적인 예측을 벗어나지는 못한다. 이는 작업자의 일정한 판단 기준이 데이터의 정제 과정에서 작용한 결과로, 전수 조사할 수 없는 데이터베이스의 한계로도 볼 수 있다. 또한 데이터 마이닝으로 분석한 데이터는 숨은 의도를 파악하지 못할 가능성이 크다. 이는 데이터베이스를 구축할 때 불가피하게 맥락(Context)이 제거되면서 데이터를 분석할 때 왜곡이 발생할 수도 있음을 의미한다[33]. 데이터 마이닝의 한계를 극복하기 위해 발전된 다양한 마이닝 기법들이 있다. [표 2-5]는 비정형 대화 텍스트를 분석하는 4가지 기법을 정리한 것이다.

[표 2-5] 비정형 대화 텍스트 분석 기법[33]

Technique	Concept
텍스트 마이닝 (Text Mining)	<p>텍스트(text) 형태의 비정형 데이터들을 자연어 처리(Natural Language Processing : NLP) 방식을 이용하여 정보를 추출하는 기법. 텍스트 정보에서 문맥을 파악하고 텍스트 간 연계를 분석하는 등 비정형화된 문서에서 정보를 얻을 수 있다는 장점이 있음.</p>
오피니언 마이닝 (Opinion Mining)	<p>소셜 미디어 등에서 여론의 향방을 긍정(Positive), 부정(Negative), 중립(Neutral)으로 구분하여 선호도를 판별하는 기술. 분석 대상인 키워드와 긍정, 부정, 중립을 뜻하는 단어가 자주 출현하는 빈도수를 측정하여 해당 키워드에 대한 여론을 측정.</p>
소셜 네트워크 분석 (Social Network Analytics)	<p>소셜 네트워크 서비스(SNS)에 내포된 정보들과 시스템을 분석하는 기법. 실시간으로 수집되는 대화 정보를 통해 신제품 반응 관찰, 기업 및 브랜드와 상품에 대한 평가, 불만 모니터링, 상품 및 서비스에 대해 소비자들이 드러낸 행적을 토대로 추출한 성향 파악과 숨은 니즈(Needs) 등의 다양한 정보를 얻을 수 있음.</p>
군집 5분석 (Cluster Analytics)	<p>변화가 많은 대상 집단을 일정한 군집으로 나누어 특성을 분석하고, 각 데이터 간의 거리를 분석하여 비슷한 유형에 대한 성향을 파악하거나 타 집단과의 차이점을 관측하기 위한 목적으로 활용되는 기법.</p>

C. 텍스트 감정인식

텍스트 감정인식은 인간의 감정을 기계가 인식하는데 있어서 현재 입력 데이터에 대한 감정 분류를 해준다. 또한 지능적으로 과거 기억이나 감정 주체, 성격이나 성향 등에 따라 더욱 정확한 감정인식을 가능하게 하는 데 필요하다[34]. 과거에는 대부분 감정 키워드를 추출하여 감정을 판단하였다. 그러나 이러한 방법은 문장에 들어있는 다양한 구문 정보나 의미 정보를 잃어버리기 때문에 감정과 같은 인간의 복잡한 생각을 기계가 인식하는 데는 한계가 있다.

텍스트 기반 감정인식을 위한 다양한 방법이 연구되고 있다. 텍스트 감정인식은 문장에서 추출된 감정 키워드 데이터를 바탕으로 감정을 판단하는 기술로서, 개인에 따라 텍스트에서 사용되는 언어적 표현은 차이가 생길 수 있고 감정의 직접적인 지표로는 볼 수 없다. 하지만 발달된 감정표현 방식으로서 다양하고 섬세하게 내적 감정 상태를 표현할 수 있다. 기존의 텍스트 기반 감정인식 기술은 실제 사용자로부터 직접적으로 데이터를 수집해야 하는 불편함이 존재하고, 대량의 실험 데이터를 수집하고 분석한 후에는 분류하는 작업이 선행되어야 한다는 어려움이 있었다. 또한, 특정 키워드에 관련된 감정을 정의하는 것을 다양한 구문 정보나 의미 정보 없이 판단했기 때문에 다른 감정인식 기술에 비하여 복잡한 인간의 생각을 인식하는 데에 한계를 지니고 있었다. 그러나 최근에는 감정을 나타내는 키워드만으로 감정을 판단하는 것이 아니라 자연어 처리 기술의 발달로 키워드와 구문 간의 연관 관계도를 포함한 감정을 분석하고 있다[35].

텍스트 감정 분류는 주로 어휘를 통한 감정인식 방법이 연구된다. [36]의 연구에서는 Ekman의 6가지 감정 클래스에 해당하는 감정 단어와 감정 이모티콘 사전을 구축하였다. 이를 기반으로 블로그 텍스트에서 SVM 분류기를 활용하여 감정을 인식하고자 하였다. [37]의 연구에서는 분석 대상으로 대표적인 소셜 네트워크 서비스인 Twitter 데이터를 사용하였다. 감정 분류에 감정 사전을 구축하여 활용하는 대신 다량의 유니 그램(Uni-gram)을 사용하였다. 그러나 사전이나 어휘 자질을 사용한 지도 학습 방식의 분류에서 가장 문제가 되는 점은 데이터 부족 문제이다[5]. 감정 클래스가 다양하고 클래스별로 사용되는 어휘 또한 다양하므로 학습이 가능

할 만큼 적절한 수준의 충분한 학습 데이터를 구축하는 것이 중요한 문제이다[38]. 따라서 양질의 학습 데이터와 학습 데이터의 양, 그리고 데이터의 균형을 맞추는 것이 매우 중요하다.

딥러닝을 사용한 감정 분류 연구로 [11]의 연구에서는 CNN(Convolutional Neural Networks) 모델은 단순하게 Convolution과 Pooling만으로도 문장과 문서 분류에서는 성능이 좋았다. 또한 [12]의 연구에서는 LSTM(Long Short-Term Memory) 모델은 연속적인 입력 데이터에 대해 순서를 고려한 학습을 진행하였다. 생성 측면에서도 기계번역에서뿐만 아니라 다양한 문제에서 좋은 성능을 보여주고 있다. [5]의 연구에서는 문맥을 반영하고 대량의 어휘를 추출하여 자동 학습이 가능하도록 분류 모델 내에서 대화문의 감정 분류에 두 모델을 앙상블(Ensemble) 한 CNN-LSTM을 사용하였다. 기존 연구 모두 감정 분류 성능은 향상되었으나 하나의 문장이 한가지 단일 감정으로 분류된다는 한계가 있었다.

한국어 데이터를 기반으로 한 감정 분석 연구로 [39]의 연구에서는 한국어를 영어와 같이 단어를 중심으로 감정 단어 사전을 사용하여 감정을 분석하였다. 이때 하나의 단어가 두 가지 이상의 의미로 해석되는 중의성 문제에 빠질 수 있다는 단점이 있다. 이를 고려해서 [40]의 연구에서는 품사 태깅이나 감정 단어 사전을 사용하지 않고 단어의 순서에 초점을 맞춰 감정을 학습했다. 단어 순서의 패턴과 입력된 문장을 비교하여 분석하는 한국어에 높은 Accuracy를 가진 알고리즘을 제시했다. 하지만 모호한 표현이 아닌 명확한 감정 단어를 가진 경우에는 오히려 Accuracy가 떨어지는 결과를 보였다. 따라서 한국어의 감정 분석에서도 단어의 순서뿐만 아니라 감정 단어 사전의 사용도 중요하다. 그리고 중의성 문제를 해결하기 위해 문장 전체를 감정 분석에 활용해야 한다.

대화 감정인식 연구로 [41,42]의 연구에서는 대화의 감정 분류가 문맥에 의존할 수 있는 문제를 지적하고 이를 해결하기 위해 발화 내 어휘 자질과 대화 문맥을 이용하였다. 그러나 문맥을 확장할수록 어휘 자질이 과도하게 많아지고, 대용량의 학습 데이터가 없는 한 심각한 데이터 부족(Data sparseness) 문제가 대두된다. [43]은 데이터 부족 문제 해결을 위해 대량의 원시 코퍼스(Corpus)를 사용한 단어 임베딩(Word Embedding) 모델을 감정 분류를 위한 자질에 활용하였다. 이러한 방

법을 통해 문제가 어느 정도 해소되었으나, 구조적으로 분류 모델에서 대화 히스토리(History)를 반영한 것이 아니기 때문에 여러 번 대화 문맥을 포함한 자질 튜닝 (Feature Engineering) 과정을 거쳐야 했다.

[11]의 CNN 모델은 Convolution과 Pooling의 단순한 과정만으로 문장 및 문서 분류에서 좋은 성능을 보였다. 또한 [44]의 LSTM 모델은 순차적(Sequential)인 입력 데이터에 대해 순서를 고려한 자질의 학습과 생성이라는 점에서 기계번역 영역에서뿐 아니라 다양한 문제에서 좋은 성능을 보인다. [45]의 모델은 텍스트 중에서도 특히 길이가 짧고, 연속적인(문서 속 문장들, 대화 속 발화) 텍스트를 RNN(Recurrent Neural Network)과 CNN을 사용해 분류에 사용하여 화행(Dialogue act) 분류에서 좋은 성능을 보여주었다. 이처럼 텍스트에서 감정을 인식하려는 다양한 연구가 제안되고 있으나 한가지 감정으로만 인식하여 정확한 감정 인식이 어려운 문제를 개선하기 위해서는 문맥을 파악하고 의미를 분석하여 내재한 다양한 감정까지 인식하는 방법이 필요하다.

D. 멀티 레이블 감정인식

멀티 레이블(Multi-Label) 분류 방법은 객체에 단일 값이 아닌 n 개의 레이블이 있는 자료를 분류한다. 그리고 해당 n 개의 레이블은 벡터로 처리할 수 있다. 대개 해당 레이블은 이진값으로 표기된다. 단일 레이블 분류 방법과 달리 멀티 레이블 분류는 n 개의 여러 레이블을 처리할 수가 있어서 감정 분석 연구에 필요하다. 이진 분류 데이터 셋은 멀티 레이블 분류의 입장에서 가장 기본적인 데이터 구조이다. 단일 레이블 멀티 클래스(Multi-Class) 역시 멀티 레이블 분류 데이터 셋과는 차이가 있음을 알 수 있다. 단일 레이블 멀티 클래스 구조를 가진 데이터 셋의 경우 이를 적절하게 사용하면 멀티 레이블 문제를 일부 해결할 수 있는 것처럼 보인다. 하지만 레이블 수가 증가할 때마다 멀티 레이블에서 해당 레이블 변수는 $2n$ 개의 출력값으로 표현된다. 따라서, 멀티 클래스 분류는 단일 레이블 분류 방법의 레이블 수를 전부 반영할 수 없다[46].

감정을 멀티 레이블로 인식하는 방법은 감정을 분류하는 기준이 명확하게 없는 것에서부터 시작되었다. 한 문장을 하나의 감정으로 단일 레이블링하는 대신 다양한 감정으로 멀티 레이블링하는 방법이 제안되었다[5]. [48]의 연구에서는 단일 레이블 감정 분류 Accuracy 향상은 더욱 어려워지고, 한 문장이 단일 레이블링 되면 여러 감정 클래스 중 하나의 클래스에만 속하는 것으로 간주하지만 실제로는 다양한 감정을 포함하고 있다고 주장하였다. 이를 증명하기 위해 뉴스 기사에서 독자의 감정을 예측하는 멀티 레이블 감정 분류 기법을 제안하였다. [49]의 연구에서는 감정 분류 연구가 대부분 단일 레이블 감정 분류에 집중되어 있어서 다른 여러 감정의 공존을 무시한다고 지적하면서, 딥러닝 기반 접근 방식을 통해 트위터 데이터의 멀티 레이블 감정 분류를 제안하였다.

또 다른 멀티 레이블 감정인식에 관한 연구로 [14]의 연구에서는 MEISD (Multimodal Multi-Label Emotion, Intensity and Sentiment Dialogue dataset)을 활용하여 순환 신경망(Recurrent Neural Network, RNN)을 변형한 DialogueRNN(Attentive RNN in Conversations)을 기반으로 멀티 레이블 감정을 인식하였다. 오디오와 비디오 보다 텍스트의 멀티 레이블 감정인식의 결과가 더 정

확한 것을 증명하였다. [15]의 연구에서는 SemEval-2018(Semantic Evaluation-2018)을 활용하여 Attention과 Convolution을 결합한 AttnConvnet(Attention-based Convolutional Neural Networks)을 기반으로 문장의 멀티 레이블 감정을 인식하였다. 인간이 문장을 분석할 때 사용하는 2단계 절차인 먼저 문장의 의미를 이해하고 감정을 분류하는 절차를 활용하였다. [16]의 연구에서는 트윗에 콘텐츠 기반 방법(단어 및 문자 n-gram)을 적용하여 멀티 레이블 감정을 분류하였다. 콘텐츠 기반 단어 Uni-gram이 다른 방법보다 능가함을 보여주었다. 하지만 대부분 영어 데이터를 활용한 연구들이었고 최근에는 한국어 대화 데이터 셋을 활용한 멀티 레이블 분류 연구[49]도 진행되고 있으나 더 다양한 연구가 필요하다. 따라서 한국어 대화의 특성을 파악하여 대화의 의미를 이해하고 내재한 감정을 인식하여 문장에 포함된 다양한 감정을 인식하는 방법이 필요하다.

멀티 레이블 분류 알고리즘은 단일 레이블 분류 방법과 매우 유사하다. 그렇지만 단일 레이블 분류 방법의 알고리즘을 멀티 레이블 분류 방법에 적용하기 위해서는 알고리즘의 변용이 필요하다. BR(Binary Relevance)는 레이블 값이 이진값으로 되어 있는 간단한 형태의 멀티 레이블 문제를 각각의 레이블마다 독립적으로 평가하는 방식이다. CC(Classifier Chain)는 이진 분류기의 가장 잘 알려진 멀티 레이블 앙상블로, 데이터에서 많은 레이블이 무작위 순서로 선택되어 k개의 모델을 훈련해 평가하는 방식이다[50]. LP(Label Powerset)는 레이블 조합을 기반으로 하는 가장 기본적인 변환 방법이다[51]. LP는 일반적으로 레이블 종속성 정보를 고려하므로 BR과 비교하여 평가한다. RAKEL(Random-kLabelsets)는 레이블의 무작위 부분집합에 대해 다 계층 분류기를 학습하는 방법이다. CLR(Calibrated Label Ranking)은 이진 분류기 앙상블이다[52]. 레이블 쌍의 관련성을 구별하는 방식인 OVO(One Versus One) 접근 방식으로 가상 레이블을 만든 다음 최종 분류기를 보정하고 관련성이 없는 레이블과 있는 레이블을 분리하여 평가한다[53]. BP-MLL(BackPropagation)은 인공신경망 모델을 기반으로 만들어진 알고리즘이며, ML-kNN은 k-최근접 이웃(kNN) 방법을 전환하여 멀티 레이블에 적용한 것이다[54].

[그림 2-3]은 이진 분류 데이터 셋, 단일 레이블 멀티 클래스 데이터 셋, 그리고 멀티 레이블 데이터 셋을 알기 쉽게 표현하여 데이터 속성에 따른 특징을 나타낸 것이다. 따라서 대화에 내재한 다양한 감정을 인식하여 감정인식의 성능을 향상하기 위해서는 대화 문장에 포함된 멀티 레이블 감정을 인식하는 방법이 필요하다.

Binary Dataset						Multi-Class Dataset					
X_1	X_2	...	X_{f-1}	X_f	class	X_1	X_2	...	X_{f-1}	X_f	class
X_{11}	X_{12}	...	X_{1f-1}	X_{1f}	1	X_{11}	X_{12}	...	X_{1f-1}	X_{1f}	C7
X_{21}	X_{22}	...	X_{2f-1}	X_{2f}	0	X_{21}	X_{22}	...	X_{2f-1}	X_{2f}	C1
...					1	...					C6
X_{n1}	X_{n2}	...	X_{nf-1}	X_{nf}	0	X_{n1}	X_{n2}	...	X_{nf-1}	X_{nf}	C3

Multi-Label Dataset								
X_1	X_2	...	X_{f-1}	X_f	Y_1	...	Y_k	
X_{11}	X_{12}	...	X_{1f-1}	X_{1f}	0	1	0	
X_{21}	X_{22}	...	X_{2f-1}	X_{2f}	1	0	1	
...					0	0	1	
X_{n1}	X_{n2}	...	X_{nf-1}	X_{nf}	1	1	0	

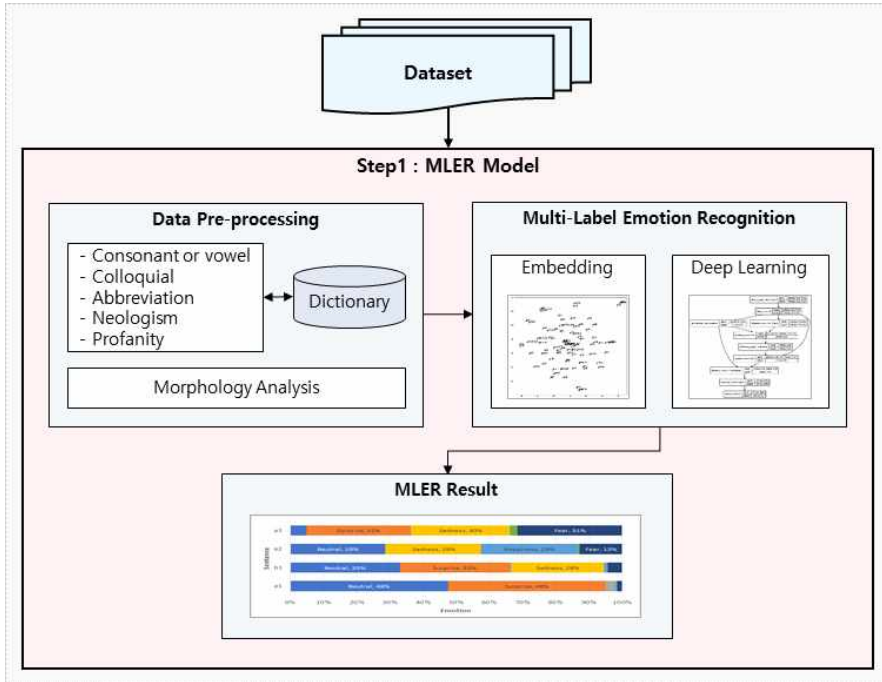
[그림 2-3] 데이터 속성에 따른 특징[46]

Ⅲ. 멀티 레이블 감정인식 모델

본 장에서는 더 정확한 감정인식을 위해 비정형 대화 텍스트를 처리하는 방법과 멀티 레이블 감정인식(Multi-Label Emotion Recognition : MLER) 모델을 제안한다. 비정형 대화 텍스트의 유형을 파악하고 처리하기 위해 표준어로 변환하는 과정을 기술하고, 문장을 임베딩(Embedding) 한 다음 문장에 내재한 다양한 감정을 인식하는 멀티 레이블 감정인식 모델에 대해 기술한다.

A. 모델 구성도

본 절에서는 비정형 대화 텍스트를 처리한 다음 멀티 레이블 감정인식 모델의 프로세스에 대해 기술한다. [그림 3-1]은 제안하는 모델의 구성도를 나타낸다.



[그림 3-1] 멀티 레이블 감정인식 모델 구성도

[그림 3-1]과 같이 멀티 레이블 감정인식 모델에는 데이터 전처리 단계와 문장 임베딩 단계, 모델 구현 단계가 있다. 데이터 전처리 단계에서는 초성체, 중성체, 구어체, 신조어, 줄임말, 비속어를 비정형 대화 텍스트로 정의하고 표준어로 변환한 다음 데이터에 적합한 형태소 분석기를 활용하여 형태소를 분석한다. 문장 임베딩 단계에서는 전처리한 문장을 Word2vec과 KoBERT로 벡터화를 진행한다. 모델 구현 단계에서는 학습된 벡터를 LSTM, CNN-LSTM, Attention, Transformer 모델로 학습하여 멀티 레이블 감정인식 모델을 구현한다.

B. 데이터 전처리

본 절에서는 감정인식 모델의 성능을 향상하기 위한 데이터의 전처리 방법으로 비정형 대화 텍스트의 유형을 파악하고 표준어로 변환하는 방법과 형태소 분석 방법에 대해 기술한다.

1. 비정형 대화 텍스트 변환

감정인식 모델의 더 나은 성능을 위해 비정형 대화 텍스트를 표준어로 변환하는 과정이 필요하다. 본 논문에서는 초성체(Consonant), 중성체(Vowel), 구어체(Colloquial), 신조어(Neologism), 줄임말(Abbreviation), 비속어(Profanity)를 비정형 대화 텍스트로 정의한다. [표 3-1]은 비정형 대화 텍스트의 변환 방법을 나타낸다.

[표 3-1] 비정형 대화 텍스트 변환 방법

Type	Original → Process
초성체(Consonant)	ㅇㅈ → 어쩔, ㅇㅊ → 인정, ㅈㅈ → 죄송 ㄹㅇ → 레알(real), ㄹ → 리플레이(replay)
중성체(Vowel)	ㅓ → 우, ㅕ → 유, ㅡ → 으
구어체(Colloquial)	~당 → ~다, ~욘 → ~요, ~넝 → ~네
신조어(Neologism)	롭곡옘눈 → 폭풍눈물, IdH → 배, 땡땡이 → 멍멍이
줄임말(Abbreviation)	마상 → 마음의 상처, 팬아저 → 팬이 아니어도 저장 TMI → 너무 과한 정보(Too much information)
비속어(Profanity)	개(이득) → 매우(이득), 킹스 → 혐오스럽다 존버하자 → 존X버티자(어떤 상황이든 버티고 보자)

[표 3-1]과 같이 자음으로만 사용한 경우 초성체(Consonant)라고 정의한다. ‘ㅇ, ㅈ, ㅇㅈ’과 같이 특정한 단어의 자음만 사용한 경우에는 해당하는 단어인 ‘어쩔, 인정’으로 변환한다. 또한 ‘ㄹㅇ, ㄹ’ 등 특정한 영어 단어의 한글 초성으로만 사용한 경우에는 해당하는 단어인 ‘레알, 리플레이’로 변환한 다음 다시 해당하는 의미의 한글로 변환이 필요하다. 모음으로만 사용한 경우에는 중성체(Vowel)라고 정의한다. ‘ㅏ, ㅓ, ㅡ’ 과 같이 모음만 사용한 경우에는 ‘우, 유, 으’와 같이 초성+중성+(중성)의 형태로 변환한다.

‘~넹’ 등 온라인상에서 자주 사용되는 구어체(Colloquial)는 ‘~네’와 같이 표준어로 변환한다. 신조어(Neologism)는 Naver 오픈 사전 PRO에서 제공하는 신조어 사전 [55]과 위키백과에서 제공하는 신조어 목록[56]을 참고하여 표준어로 변환한다. 줄임말(Abbreviation)은 나무위키의 줄임말 목록[57]을 참고하여 표준어로 변환한다. 비속어(Profanity)는 국립국어원의 표준국어대사전[58]을 참고하여 표준어로 변환한다. [표 3-1]의 예시와 같이 비정형 대화 텍스트에 적합한 표준어가 1:1로 매칭되는 경우에는 이러한 처리 규칙에 따라 해당하는 표준어로 변환한다.

그러나 1:1로 매칭되지 않는 경우에는 이러한 고정적인 변화가 오히려 일반화 저하의 문제를 가져올 수 있다. 예를 들어 ‘ㅋ’는 1회부터 수십 회 반복하여 감정을 표현하는 경우가 많고 연속적으로 5회 이상 쓰이는 경우가 특히 많은 것으로 분석되었다[59]. 이런 경우에는 짧게 표현했을 때의 의미와 구분되는 의미의 변화가 있거나 의미상의 강조를 나타내고 있음을 예상할 수 있다. 따라서 1:1로 매칭되지 않는 경우에는 문장의 의미를 파악하고 그에 맞는 추가적인 변환 방법이 필요하다. [표 3-2]의 Algorithm 1은 본 논문에서 사용한 비정형 대화 텍스트 처리 알고리즘을 나타낸다.

[표 3-2] Algorithm 1 : Unstructured conversation text processing[60]

Algorithm 1. Unstructured conversation text processing

Input: ODL:OriginalDataList

Output: PDL:ProcessDataList

Definition: sen - sentence, sw - standard word, uw - unstructured word

Dic - sw definition for uw

[consonant, vowel, colloquial, neologism, abbreviation, profanity]

```

1  READ ODL;
2  for each sen in ODL do
3      for each word in sen do
4          if uw in Dic then
5              uw ← sw;
6          end if
7      end for
8  end for
9  PDL ← ODL;
10 return PDL;

```

[표 3-2]의 Algorithm 1과 같이 원본 데이터 셋(OriginalDataList)을 입력으로 한 문장(sen)을 추출하고, 추출된 문장에서 단어(word)를 추출한다. 다음으로 단어가 사전(Dic)에 포함된 비정형 단어(uw)인지 확인한다. 사전에 있으면 매칭된 표준어(sw)로 변환하고 표준어로 변환된 원본 데이터는 처리 데이터(ProcessDataList)에 저장한다.

2. 형태소 분석

비정형 대화 텍스트를 처리한 데이터는 형태소 분석(Morphological Analysis)을 진행한다. 형태소 분석은 텍스트 데이터에서 문자열을 분석하여 형태소 단위로 분해하는 과정이다. 형태소란 더는 분해할 수 없는 의미를 갖는 요소로서는 최소 단위이고 구분된 형태소들을 통해 문장의 의미를 파악한다. 형태소 분석기에 따라 차이가 있으므로 데이터 셋에 적합한 방법을 선택해야 한다. [표 3-3]은 4가지 형태소 분석기를 비교한 것이다.

[표 3-3] 형태소 분석기 비교[61]

형태소 분석기	내용
원문장	사랑하고싶게하는마음속온감정을헤집어놓는영화예요정말최고
Mecab	[[‘사랑/NNG’, ‘하/XSV’, ‘고/EC’, ‘싶/VX’, ‘게/EC’, ‘하/VV’, ‘는/ETM’, ‘마음속/NNG’, ‘온/VV+ETM’, ‘감정/NNG’, ‘을/JKO’, ‘헤집/VV’, ‘어/EC’, ‘놓/VX’, ‘는/ETM’, ‘영화/NNG’, ‘예요/VCP+EF’, ‘정말/MAG’, ‘최고/NNG’]]
Komorran	[[‘사랑/NNG’, ‘하/XSV’, ‘고/EC’, ‘싶/VX’, ‘게/EC’, ‘하/VX’, ‘는/ETM’, ‘마음속/NNG’, ‘오/VV’, ‘ㄴ/ETM’, ‘감정/NNP’, ‘을/JKO’, ‘헤집/VV’, ‘어/EC’, ‘놓/VX’, ‘는/ETM’, ‘영화/NNP’, ‘예/NNG’, ‘요정/NNP’, ‘말/NNG’, ‘최고/NNP’]]
Kkma	[[‘사랑/NNG’, ‘하/XSV’, ‘고/ECE’], [‘싶/VXA’, ‘게/ECD’], [‘하/VV’, ‘는/ETD’], [‘마음속/NNG’], [‘오/VV’, ‘ㄴ/ETD’], [‘감정/NNG’, ‘을/JKO’], [‘헤집/VV’, ‘어/ECD’], [‘놓/VXV’, ‘는/ETD’], [‘영화/NNG’, ‘이/VCP’, ‘예요/EFN’], [‘정말/MAG’], [‘최고/NNG’]]
Okt	[‘사랑/Noun’, ‘하다/Verb’, ‘싶다/Verb’, ‘마음속/Noun’, ‘온/Noun’, ‘감정/Noun’, ‘을/Josa’, ‘헤집다/Verb’, ‘영화/Noun’, ‘예요/Josa’, ‘정말/Noun’, ‘최고/Noun’]

[표 3-3]과 같이 비교한 결과 Kkma와 Komoran은 다른 분석기보다 더 구체적으로 형태소에 따라 단어를 쪼개는 것을 볼 수 있다. Okt의 경우 다른 분석기들과 미세하게 결과가 다른 것을 볼 수 있다. Okt는 stem=True, norm=True의 파라미터가 존재하므로, 단어들의 정규화와 오타 수정도 가능하다. ‘사랑하고’는 ‘사랑하다’로, ‘싫게’는 ‘싫다’로, ‘헤집어놓는’은 ‘헤집다’로 변경된 것을 볼 수 있다[61]. 본 논문에서 사용한 데이터 셋에는 띄어쓰기가 무시된 문장이 많았다. 따라서 한국어 대화 데이터에서 띄어쓰기가 안 된 문장의 분석에 좋은 Okt(Open Korean Text) 형태소 분석기를 사용하였다.

C. 멀티 레이블 감정인식

본 절에서는 더 정확한 감정인식을 위해 전처리한 데이터를 활용하여 문장을 임베딩하고 내재한 다양한 감정을 인식하는 멀티 레이블 감정인식(MLER) 모델에 관해 기술한다. 문장 임베딩 방법으로는 Word2Vec과 KoBERT를 활용하고 그 결과를 비교하여 성능이 가장 좋은 모델을 선정한다. 또한 멀티 레이블 감정인식 방법으로는 LSTM, CNN-LSTM, Attention, Transformer 모델을 활용하여 감정인식 모델을 구축하고 그 결과를 비교하여 Accuracy가 가장 좋은 모델을 선정한다.

1. 문장 임베딩

a. Static embedding model

Static embedding model로는 학습을 완료한 뒤 고정된 벡터값을 반환하는 Word2vec 모델을 활용하여 학습한다. 비정형 대화 텍스트 중 한국어 데이터는 특히 단어의존도가 높은 만큼 단어의 순서와 전체 문장에 대한 의미 파악이 중요하다. 따라서 앞뒤 문맥을 고려하여 의미를 폭넓게 파악할 수 있는 Word2vec 모델을 사용하였다. 단어의 순서에 맞는 의미 파악을 위해 데이터의 문장 전체를 Word2vec 모델로 학습하여 단어를 벡터화한다.

[표 3-4]의 Algorithm 2는 본 논문에서 사용한 Word2vec 임베딩 모델의 알고리즘을 나타낸다. Word2vec 모델에서 사용한 학습 방식은 center-word에서 context-word를 예측하는 Skip-Gram을 사용하였다. 하이퍼 파라미터(Hyper parameter)로는 Learning Rate는 0.05, Dimension(Vector space)은 512, Window Size는 2, Min Count는 5로 설정한다[60]. Word2vec 모델로 학습하여 임베딩 한 결과 데이터 셋에서 의미가 유사한 단어들이 근접한 벡터 공간에 위치하였고 단어의 문맥적 의미가 보존되었다[62]. 따라서 임베딩을 통해 데이터 셋에서 의미적으로 유사한 단어의 추출이 가능하고 벡터값을 딥러닝 모델의 입력으로 사용하여 학습할 수 있다.

[표 3-4] Algorithm 2 : Word2vec embedding[60]

Algorithm 2. Word2vec embedding

Input: d:dataset

Output: Matrix $W_{(512,113)}$ of one-hot vectors for each possible byte value (0-255)

```

1  let f be a list of tuples (byte_value, frequency);
2  for i := 0 to 511 do
3    freq ← 0;
4    for each item j in d do
5      freq ← freq + frequencyOfOccurence(i,j);
6    end for
7    append (i,freq) tuple to f;
8  end for
9  f ← sort f based on frequencies;
10 W ← word2vec(f,113);
11 return W;

```

b. Pre-trained language model

Pre-trained language model로는 사전 학습된 임베딩 모델 자체를 불러와 적용하는 BERT 모델을 활용한다. BERT(Bidirectional Encoder Representations from Transformers)는 2018년 11월 구글이 공개한 인공지능 언어 모델로 전이 학습(Transfer Learning), 트랜스포머(Transformer) 구조를 활용해 구현되었으며, 위키 피디아와 BooksCorpus와 같은 레이블이 없는 텍스트 데이터로 훈련되었다. BERT는 파라미터를 재조정하여 추가 훈련인 파인튜닝(fine-tuning)을 할 수 있고, 파인튜닝 시 성능이 높게 나온다. 이를 통해 각종 자연어 처리 태스크(Task)를 해결한다[63].

BERT의 사전훈련 방법은 마스크드(Masked) 언어 모델과 다음 문장을 예측하는 두 가지 방법이 있다. 마스크드 언어 모델은 입력 텍스트 중 15%의 단어를 랜덤

(Random)으로 마스킹한 다음 마스킹 된 단어들의 80%는 [MASK]로 변경한다. 또한 10%는 랜덤으로 단어를 변경하고 나머지 10%는 동일하게 유지한다. 그리고 인공신경망을 통해 가려진 단어(Masked words)를 예측한다. 다음 문장 예측 방법은 두 개의 문장을 준 다음 이 문장이 이어지는 문장인지 아닌지를 맞히는 방식으로 훈련한다. 이를 위해 50:50 비율로 실제 이어지는 두 개의 문장과 랜덤으로 이어 붙는 두 개의 문장을 주고 훈련한다.

본 논문에서는 BERT에서 파생된 많은 모델 중에서 한국어 버전(Version)의 자연어 처리 모델(SKT Brain KoBERT Model : KoBERT)을 활용하여 문장 임베딩을 학습한다. 입력 문장을 토큰라이저(Tokenizer) 방법의 하나인 BertTokenizer를 통해 토큰화하고 bag-of-words와 같은 방법으로 단어를 input ids로 변환한다. 사전 학습된 KoBERT의 출력 vector size는 768이고 학습에 사용된 말뭉치의 크기는 문장 5M, 단어 54M, 단어의 개수는 8,002이다. 최대 입력을 512개의 토큰까지 받으며, 출력의 크기는 768이다[64]. [표 3-5]는 본 논문에서 사용한 KoBERT embedding 방법을 나타낸다.

Word2Vec과 KoBERT를 활용하여 문장 임베딩을 구성한 다음 멀티 레이블 감정 인식 모델을 구축한다. 구축된 모델을 통한 감정인식 결과를 비교하여 가장 성능이 좋은 문장 임베딩 방법을 선정한다.

[표 3-5] KoBERT embedding

```

def BERT_token_embedding(text, model):
    marked_text = "[CLS] " + text + " [SEP]"

    tokenized_text = tokenizer.tokenize(marked_text)
    indexed_tokens = tokenizer.convert_tokens_to_ids(tokenized_text)

    segments_ids = [1] * len(tokenized_text)
    tokens_tensor = torch.tensor([indexed_tokens])
    segments_tensors = torch.tensor([segments_ids])

    with torch.no_grad():
        outputs = model(tokens_tensor, segments_tensors)
        hidden_states = outputs[2]

    batch_i = 0
    token_i = 0
    layer_i = 0
    vec = hidden_states[layer_i][batch_i][token_i]

    token_embeddings = torch.stack(hidden_states, dim=0)
    token_embeddings = torch.squeeze(token_embeddings, dim=1)
    token_embeddings = token_embeddings.permute(1,0,2)

    token_vecs_sum = []

    for token in token_embeddings:
        sum_vec = torch.sum(token[-4:], dim=0)
        token_vecs_sum.append(sum_vec)

    token_vecs = hidden_states[-2][0]
    return np.array(token_vecs)
    
```

2. 멀티 레이블 감정인식

멀티 레이블 감정인식을 위해 문장 임베딩으로 학습된 벡터와 감정 클래스를 딥러닝 모델을 활용하여 학습한다. 주어진 문장의 감정을 학습하고 복합적인 감정인 멀티 레이블 감정을 예측하기 위해 LSTM, CNN-LSTM, Attention, Transformer 모델을 활용한다.

a. LSTM 모델

LSTM(Long Short-Term Memory)은 RNN(Recurrent Neural Network)의 한 종류로서 배열(Sequence)의 형태를 갖는 데이터에서 패턴을 인식하는 인공지능망이다. [표 3-6]은 본 논문에서 사용한 LSTM 모델의 구조를 나타낸다.

[표 3-6] LSTM 모델 구조

```

model = Sequential()
model.add(LSTM(32, input_shape=(X.shape[1], X.shape[2])))#
model.add(Dense(64, activation='relu'))
model.add(Dense(Y.shape[1], activation='softmax'))

model.compile(loss='categorical_crossentropy', optimizer='adam',
              metrics=['accuracy'])
print(model.summary())

history = model.fit(X_train, Y_train, epochs=30,
                   validation_data=(X_test, Y_test), verbose=2)

```

[표 3-6]과 같이 LSTM 모델 학습을 위해 문장 임베딩 된 3차원 벡터를 X, 숫자 0~6으로 이루어진 7가지 감정 레이블(Label)을 Y로 지정한다. LSTM 모델의 하이퍼 파라미터(Hyper Parameter)로 Units는 32, Time Step은 단어의 개수인 113, 특

성(Feature)은 문장 임베딩 학습할 때 사용된 차원(Dimension)의 개수인 512로 지정하였다. 신경망 층인 Dense는 은닉층과 출력층으로 구성하고, 은닉층의 하이퍼 파라미터 Units는 64, 출력층의 하이퍼 파라미터는 7가지 멀티 레이블 감정을 예측을 위해 Units 7, 활성화 함수(Activation)는 Softmax로 설정하였다.

b. CNN-LSTM 모델

CNN(Convolution Neural Networks)은 일반적인 인공신경망 계층과 하나 또는 여러 개의 합성곱 계층들로 이루어져 있는 인공신경망이다. CNN-LSTM 모델은 CNN과 LSTM 모델을 앙상블(Ensemble) 한 모델이다. [표 3-7]은 본 논문에서 사용한 CNN-LSTM 모델을 나타낸다.

[표 3-7] CNN-LSTM 모델 구조

```

model = Sequential()
model.add(Conv1D(filters=32, kernel_size=5, padding="same",
                 activation="relu", input_shape=[X.shape[1], X.shape[2]]))
model.add(LSTM(16))#, input_shape=(X.shape[1], X.shape[2]))
model.add(Dense(16, activation='relu'))
model.add(Dense(Y.shape[1], activation='softmax'))#softmax

model.compile(loss='categorical_crossentropy', optimizer='adam',
              metrics=['accuracy'])

history = model.fit(X_train, Y_train, epochs=30,
                    validation_data=(X_test, Y_test),
                    verbose=2) #callbacks=[checkpoint]

```

[표 3-7]과 같이 CNN-LSTM 모델은 입력층과 은닉층, 출력층으로 구성된다. 입력층의 데이터는 2차원으로 구성되어 있으며, 전체 데이터 중 문장 기준 최대 단어

의 개수인 113과 문장 임베딩 벡터 차원 개수인 512의 입력을 가진다. 입력층 CNN 모델은 5 크기의 필터가 32개 존재하며 은닉층은 16개의 유닛을 가진 LSTM을 2층 구조로 쌓고, Dense층은 32개의 노드로 구성된다. 마지막 출력층에서는 7가지 멀티 레이블 감정을 출력으로 놓고 활성화 함수는 Softmax로 구성하여 감정 분류를 진행한다. 분석 결과는 정확한 성능 평가를 위해 Accuracy를 확인하였다.

c. Attention 모델

Attention은 decoder에서 출력 단어를 예측하는 매시점마다 encoder의 입력 시퀀스를 다시 참고하는 모델이다. 이때 입력 시퀀스를 예측 단어와 관련이 있는 입력 단어로 더욱 치중해서 보기 때문에 Attention value가 추가로 필요하다. 식 (1)~(3)은 본 논문에서 사용한 Attention value를 구하는 식을 나타낸다.

$$score(s_t, h_i) = s_t^T h_i \quad (1)$$

$$\alpha^t = softmax(e^t) \quad (2)$$

$$a_t = \sum_{i=1}^N \alpha_i^t h_i \quad (3)$$

식 (1)은 decoder의 time step에서 hidden state와 encoder의 time step에서 hidden state의 Attention score를 구하는 방법을 나타낸다. 식 (2)는 식 (1)에서 얻은 Attention score에 softmax 함수를 적용하여 모든 값의 합이 1이 되는 확률 분포 Attention Distribution을 얻는 방법을 나타낸다. Attention Distribution은 [0.1, 0.4, 0.1, 0.3]의 형태를 갖는 벡터를 얻게 되며 각각의 값을 Attention Weight라 한다. 식 (3)은 식 (2)에서 구한 Attention Weight와 각 hidden state를 통해 Attention value를 얻는 방법을 나타낸다. 최종적으로는 Attention value와 decoder의 시점의 hidden state를 연결한다. [표 3-8]의 Algorithm 3은 본 논문에서 사용한 Attention 모델의 알고리즘을 나타내고, [표 3-9]는 Attention 모델의 구조를 나타낸다.

[표 3-8] Algorithm 3 : Attention model[60]

Algorithm 3. Attention model

Input: d :dataset, l :dataset true lables, W :word2vec matrix

Output: MLE of Attention trained model on test dataset

Definition: MLE(multi-label emotion)

```

1   let f be the featureset 3d matrix;
2   for i in dataset do
3       let  $f_i$  be the featureset matrix of sample i;
4       for j in i do
5            $v_j \leftarrow \text{vectorize}_{(j,w)}$ 
6           append  $v_j$  to  $f_i$ 
7           append  $f_i$  to f
8       end for
9   end for
10   $f_{\text{train}}, f_{\text{test}}, l_{\text{train}}, l_{\text{test}} \leftarrow \text{split feature set and lables into train}$ 
                                     subset and test subset
11   $M \leftarrow \text{Attention}(f_{\text{train}}, l_{\text{train}})$ 
12   $\text{MLE} \leftarrow \text{evaluate}(f_{\text{test}}, l_{\text{test}}, M)$ 
13  return MLE

```

[표 3-9] Attention 모델 구조

```

time_steps, input_dim, output_dim = X_train.shape[1], X_train.shape[2],
                                     Y_train.shape[1]

model = Sequential()
model.add(LSTM(64,
              input_shape=(time_steps,input_dim),return_sequences=True))
model.add(Attention(units=128, name='attention_weight'))
model.add(Dense(output_dim, activation='sigmoid'))
model.compile(loss='categorical_crossentropy',
              optimizer='adam',
              metrics=['accuracy'])
print(model.summary())

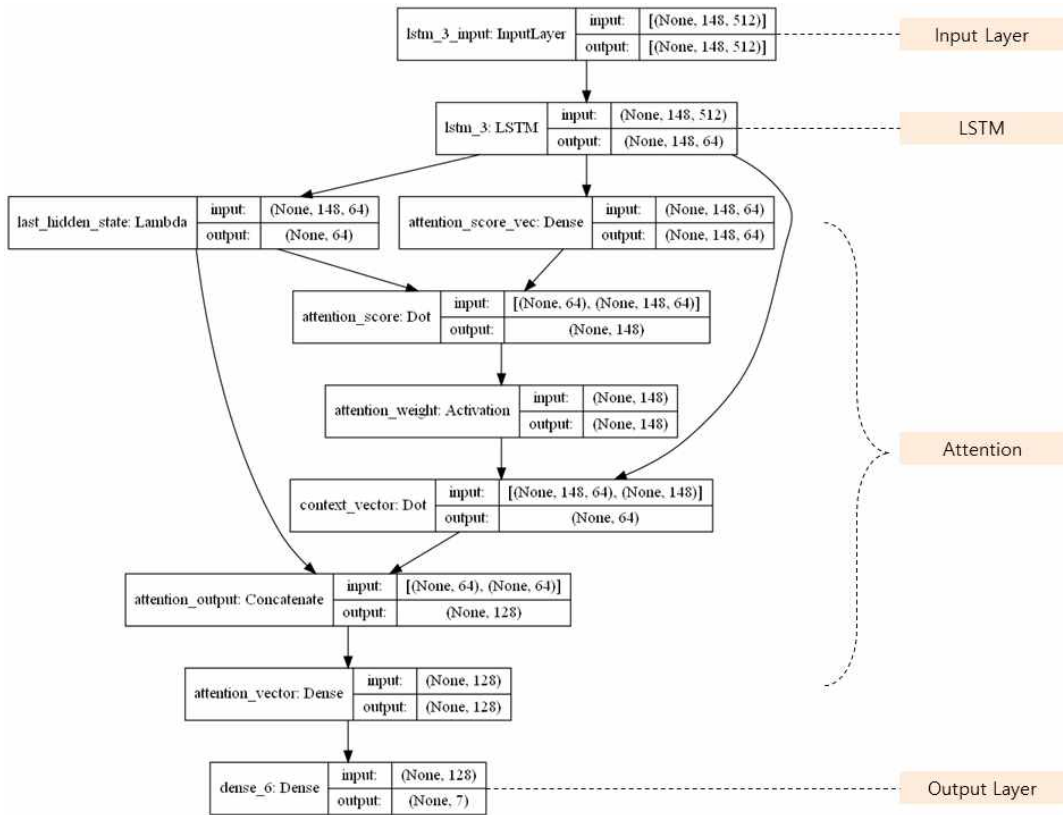
history = model.fit(X_train, Y_train, epochs=30,
                   validation_data=(X_test, Y_test), verbose=2)

model.save('./model_save/attention.hdf5')

```

[표 3-9]와 같이 설계한 모델은 LSTM의 단점인 Vanishing Gradient 문제를 해결하기 위해 Attention으로 구성하였다. 입력 데이터는 수집된 문장의 개수와 한 문장당 최대 단어 개수인 113개이다. 단어를 학습한 문장 임베딩 한 512차원으로 3차원(문장 개수, 단어 길이, 임베딩 벡터)으로 구성된다. Attention은 LSTM의 출력 부분에 위치하여 문장의 길이가 길어도 잘 예측할 수 있도록 하며 유닛은 128개로 구성하였다. 출력층에서는 멀티 레이블 감정을 예측하기 위해 감정 개수에 맞게 Dense의 출력을 설정하고 활성화 함수는 softmax를 사용하였다. [그림 3-2]는 본 논문에서 사용한 Attention 모델을 나타낸다.

Attention 모델을 활용하면 문장의 의미를 이해할 수 있어서 단어들 사이의 의존성을 더욱 잘 파악할 수 있다. 의미 정보 추출을 통해 문장에 내재한 감정을 인식하고 복합적인 감정을 인식하는 것이 가능하게 된다.



[그림 3-2] Attention 모델

d. Transformer 모델

트랜스포머(Transformer)는 어텐션 메커니즘(Attention Mechanism)을 기반으로 하는데 어텐션 메커니즘은 기계번역을 위한 시퀀스 투 시퀀스(Sequence-to-Sequence) 모델에 처음 도입되었다. 번역하고자 하는 언어를 입력으로 해서 이를 벡터로 만드는 앞부분을 인코더(Encoder)라고 하고, 인코더가 출력한 벡터를 입력으로 해서 타겟 언어를 출력하는 뒷부분을 디코더(Decoder)라고 한다. 그러나 입력으로 주어지는 문장의 길이가 길어질수록 모델의 성능이 나빠져 Accuracy가 떨어지게 된다. 이를 개선하기 위해 모델이 중요한 부분에 집중하게 만드는 것이 어텐션 메커니즘의 핵심이라고 할 수 있다. [표 3-10]은 Transformer 모델의 구조를 나타낸다.

[표 3-10] Transformer 모델 구조

```

vocab_size = sentences.max()+1
embed_dim = 32
num_heads = 8
ff_dim = 32

inputs = layers.Input(shape=(maxlen,))
embedding_layer = TokenAndPositionEmbedding(maxlen, vocab_size,
                                              embed_dim)
x = embedding_layer(inputs)
transformer_block = TransformerBlock(embed_dim, num_heads, ff_dim)
x = transformer_block(x)
x = layers.GlobalAveragePooling1D()(x)
x = layers.Dropout(0.1)(x)
x = layers.Dense(20, activation="relu")(x)
x = layers.Dropout(0.1)(x)

outputs = layers.Dense(Y_train.shape[1], activation="softmax")(x)

model = keras.Model(inputs=inputs, outputs=outputs)

model.compile("adam", "categorical_crossentropy", metrics=["accuracy"])
#categorical_crossentropy, sparse_categorical_crossentropy
history = model.fit(X_train, Y_train, batch_size=32, epochs=20,
                    validation_data=(X_test, Y_test)
)
  
```

[표 3-10]과 같이 설계한 Transformer 모델은 입력 시퀀스를 인코더에서 입력받고, 출력 시퀀스를 디코더에서 출력하는 인코더-디코더 구조를 유지한다. Transformer는 인코더에서 단순히 각 단어의 임베딩 벡터를 입력받는 것이 아니라 임베딩 벡터에서 조정된 값을 입력받는다. Transformer의 핵심은 순차적으로 입력받아 처리하는 RNN 특성으로 인해 위치 정보(position information)를 가질 수 있다는 것이다.

e. 멀티 레이블 감정인식 결과 분석

앞에서 설명한 LSTM, CNN-LSTM, Attention, Transformer 모델을 활용하여 멀티 레이블 감정인식 모델을 구축한다. 본 논문에서는 감정인식 결과를 비교하여 Accuracy가 가장 좋은 멀티 레이블 감정인식 모델을 선정한다. 단일 감정으로 분류된 데이터라도 제안한 모델을 활용하면 문장 내에 있는 여러 가지 감정을 인식할 수 있다. [표 3-11]은 입력 문장의 예시를 나타내고, [표 3-12]는 입력 문장에 대한 멀티 레이블 감정인식 결과를 나타낸다.

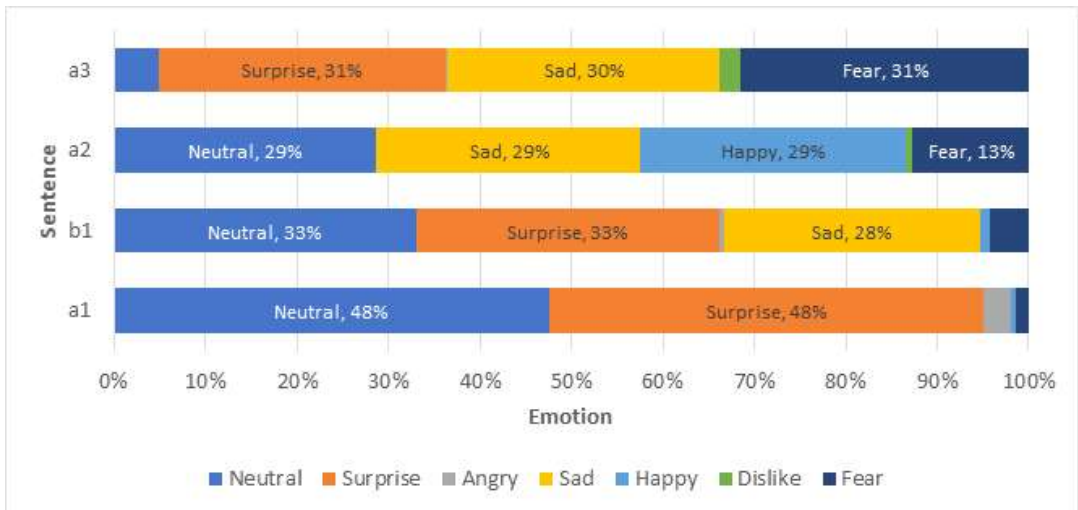
[표 3-11] 입력 문장 예시

Sentence	Emotion	Contents
a1	Neutral	물고기 키워 본 적 있어요?
b1	Neutral	없어요. 먹어본 적 밖에는.
a2	Neutral	온도는 내가 맞춰 봤으니까 내일 오전에 한번만 먹이 주면 되요.
b2	Dislike	이제 저 물고기까지 봐야 합니까?
a3	Neutral	제가 오늘부터 다음주 월요일까지 출장이라서요.
b3	Angry	그러니까 그 사정을 제가 왜 일일이 다 봐줘야 하나구요.

[표 3-12] 멀티 레이블 감정인식 결과

Sentence	Neutral	Surprise	Angry	Sad	Happy	Dislike	Fear
a1	1.00000	0.99899	0.06302	0.00073	0.00988	0.00056	0.02869
b1	0.99746	0.99805	0.01268	0.84546	0.03370	0.00090	0.12589
a2	0.99139	0.00004	0.00007	0.99990	0.99963	0.02837	0.43851
b2	0.78417	0.00762	0.99702	0.12438	0.00474	0.99999	0.00014
a3	0.15622	0.99807	0.00106	0.95076	0.00000	0.07260	1.00000
b3	0.99995	0.05140	0.98325	0.99985	0.00000	0.00002	0.00491

[표 3-11]의 입력 문장을 멀티 레이블 감정인식 모델을 적용한 결과 [표 3-12]와 같은 결과를 얻을 수 있었다. 입력 문장은 단일 감정으로 분류되어 있지만, 문장에 내재하여 있는 7가지 감정 정도를 알 수 있다. 또한 입력 문장 중 ‘b2, b3’를 제외한 다른 문장들은 ‘Neutral’로 분류되어 있어 어떤 감정인지 분류하기 어렵지만, 멀티 레이블 감정인식 결과 다른 감정들의 포함 여부를 알 수 있다. [그림 3-3]은 ‘Neutral’ 감정으로 분류된 문장의 멀티 레이블 감정인식 결과를 그래프로 나타낸 것이다.



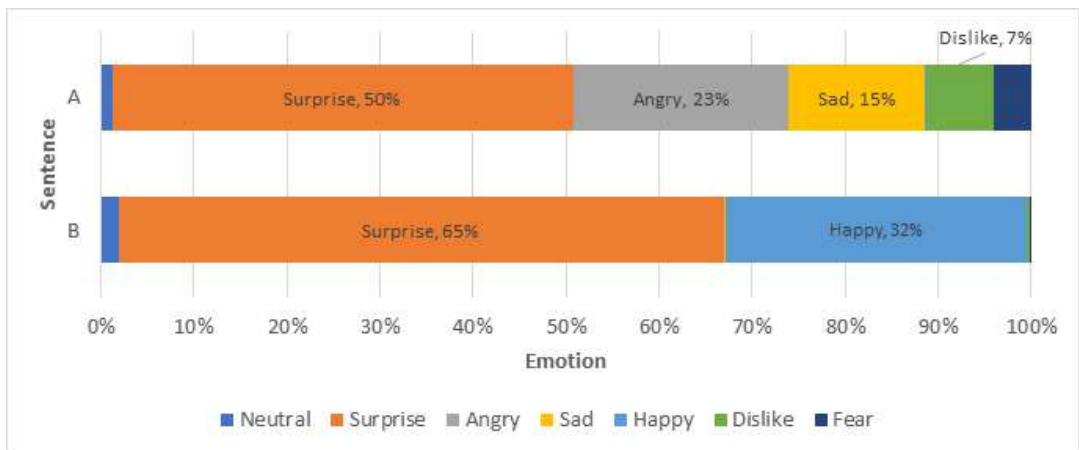
[그림 3-3] ‘Neutral’ 문장의 멀티 레이블 감정인식 예시

[그림 3-3]과 같이 ‘Neutral’로 분류된 문장의 멀티 레이블 감정인식 결과 다른 감정들의 포함 여부를 알 수 있다. ‘a1’ 문장은 ‘Surprise’ 48%, ‘b1’ 문장은 ‘Surprise’ 33%와 ‘Sad’ 28%, ‘a2’ 문장은 ‘Sad’ 29%와 ‘Happy’ 29%의 감정이 내재해 있다. ‘a3’ 문장은 ‘Neutral’보다 ‘Fear, Surprise, Sad’ 감정이 더 많이 내재한 것을 볼 수 있다. 따라서 ‘Neutral’로 분류된 문장이라도 멀티 레이블 감정인식 모델을 적용하면 문장에 내재한 세밀한 감정을 인식할 수 있어서 더 정확한 감정 분류가 가능해진다.

다음으로는 ‘Surprise’로 분류된 문장 예시를 통해 멀티 레이블 감정인식의 결과를 분석한다. 감정은 단독으로 표현되기보다는 다른 감정과 함께 표현되는 경우가

많다. 그리고 대부분 감정은 긍정과 부정으로 분류할 수 있다. ‘Happy’는 긍정으로 ‘Angry, Sad, Dislike, Fear’는 부정으로 분류된다. 하지만 ‘Neutral’은 분류가 안 되는 감정이고 ‘Surprise’는 긍·부정으로 분석하기 모호한 감정이다. 긍정적 의미의 놀람인 ‘Positive-Surprise’가 있고 부정적 의미의 놀람인 ‘Negative-Surprise’도 있다. [그림 3-4]는 ‘Surprise’로 분류된 두 문장의 멀티 레이블 감정인식 결과를 나타낸다.

A : 어떻게 사람을 이런 식으로 내치냐.
B : 오늘 아름다운 연기를 펼쳐주었군요.



[그림 3-4] ‘Surprise’ 문장의 멀티 레이블 감정인식 예시

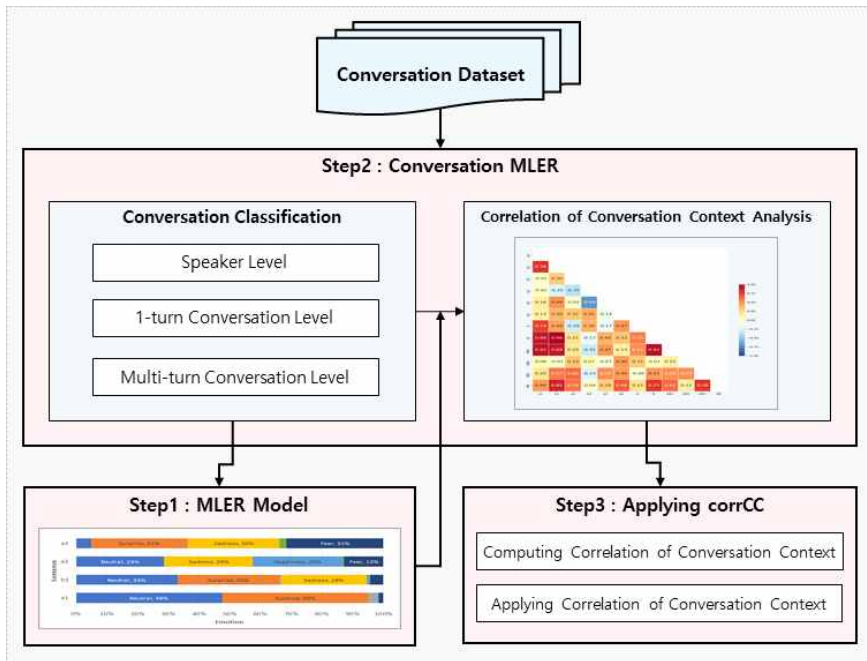
[그림 3-4]와 같이 데이터 셋에서 ‘Surprise’라는 동일한 감정으로 분류된 두 문장이지만 의미적으로 분석해보면 ‘A’ 문장(어떻게 사람을 이런 식으로 내치냐.)은 ‘Negative-Surprise’이고, ‘B’ 문장(오늘 아름다운 연기를 펼쳐주었군요.)은 ‘Positive-Surprise’이다. 두 문장의 멀티 레이블 감정인식 결과 ‘A’ 문장에는 ‘Surprise’ 50%, ‘Angry’ 23%, ‘Sad’ 15%, ‘Dislike’ 7% 등 ‘Negative’ 감정이 포함되어 있다. 반면, ‘B’ 문장에는 ‘Surprise’ 65%와 ‘Happy’ 32% 등 ‘Positive’ 감정이 포함된 것을 확인할 수 있다. 따라서 멀티 레이블 감정인식 방법을 활용하면 대화의 의미를 이해하고 내재한 감정과 복합적인 감정을 인식할 수 있어서 더 정확한 감정인식과 성능향상이 가능하다.

IV. 대화 문맥 연관성을 적용한 MLER 모델

본 장에서는 대화 문맥의 연관성을 적용한 멀티 레이블 감정인식(MLER) 모델을 제안한다. 대화의 감정인식 성능향상을 위해 두 화자(Speaker) 간 대화 데이터를 대화의 특성에 맞게 화자를 기준으로 화자별(Speaker Level : SL), 1-turn CL(Conversation Level), multi-turn CL 대화문으로 분류하고 MLER 모델을 적용하여 대화문의 감정을 인식한다. 문장과 대화문 간의 감정 분석을 통해 대화 문맥의 연관성을 도출하고 이를 적용한 MLER 모델에 관해 기술한다.

A. 모델 구성도

본 절에서는 제안하는 대화 문맥의 연관성을 적용한 MLER 모델의 프로세스에 관해 기술한다. [그림 4-1]은 제안하는 방법의 구성도를 나타낸다.



[그림 4-1] 대화 문맥의 연관성을 적용한 MLER 모델 구성도

[그림 4-1]과 같이 대화 문맥의 연관성을 적용한 MLER 모델에는 대화문 분류 단계와 멀티 레이블 감정인식 단계, 대화 문맥 연관성 적용 단계가 있다. 대화문 분류 단계에서는 두 화자(Speaker) 간 대화 데이터를 대화의 특성에 맞게 화자를 기준으로 SL, 1-turn CL, multi-turn CL 대화문으로 분류한다. 멀티 레이블 감정인식 단계에서는 MLER 모델을 적용하여 대화문의 멀티 레이블 감정을 인식한다. 대화 문맥 연관성 적용 단계에서는 문장과 대화문 간의 멀티 레이블 감정 분석을 통해 대화 문맥의 연관성을 도출하고 이를 적용한 MLER 모델을 구현한다.

B. 대화문 멀티 레이블 감정인식

본 절에서는 대화 데이터의 특성에 맞게 SL, 1-turn CL, Multi-turn CL의 세 가지 형태의 대화문으로 분류하고 멀티 레이블 감정인식(MLER) 모델을 활용하여 대화에 내재한 다양한 감정을 인식하는 방법에 대해 기술한다.

1. 대화문 분류

두 화자(Speaker) 간 대화 데이터를 대화의 특성에 맞게 화자를 기준으로 구분해 보면 세 가지로 분류할 수 있다. 첫 번째 단일 화자로 구성된 대화문(Speaker Level : SL), 두 번째 화자 간 1-turn으로 주고받는 형태의 대화문(1-turn Conversation Level : 1-turn CL), 세 번째 화자 간 연속적으로 주고받는 형태의 대화문(Multi-turn Conversation Level : Multi-turn CL)으로 분류할 수 있다. 본 논문에서는 대화 데이터 셋을 SL, 1-turn CL, Multi-turn CL의 세 가지 형태의 대화문으로 분류하여 연구에 활용한다. [표 4-1]은 대화 데이터 셋의 분류 과정을 나타낸다.

[표 4-1] 대화 분류 과정

Speaker	Sentence		Classification	Sentence	Conversation
A	a1	⇒	SL	a1,a2,a3	A
B	b1			b1,b2,b3	B
A	a2		1-turn CL	a1,b1	AB1
B	b2			a2,b2	AB2
A	a3			a3,b3	AB3
B	b3		Multi-turn CL	a1~b3	AB

[표 4-1]과 같이 A, B 두 화자 간 대화 데이터에서 SL은 A, B 화자별 대화문으로 분류하고, 1-turn CL은 A, B가 주고받은 1회의 대화문들로 분류한다.

Multi-turn CL은 A, B가 주고받은 전체 대화문으로 분류한다. [표 4-2]의 Algorithm 4는 대화 데이터를 SL 대화문으로 분류하는 알고리즘을 나타낸다.

[표 4-2] Algorithm 4 : Conversation classification : SL

Algorithm 4. Conversation classification : SL

Input: CDL:ConversationDataList

Output: SLL:SpeakerLevelList

```

1  READ CDL;
2  for each standard_s in CDL do
3    if standard_s[0] = 'S' then
4      append [A, B] to SLL;
5      number ← 1;
6      A ← standard_s[1]+' ';
7      B ← "";
8    else
9      if number%2 = 0 then
10       B ← B + standard_s[1]+' ';
11     else if number%2 = 1 then
12       A ← A + standard_s[1]+' ';
13     end if
14   end if
15   number ← number + 1;
16 end for
17 return SLL;

```

[표 4-2]의 Algorithm 4와 같이 대화 형식의 데이터는 현재 하나의 Column에 나열되어있다. 이를 SL 대화문으로 분류하기 위해 알고리즘을 적용하였다. 하나의 대화는 'S'라는 구분자를 가지고 있어서 데이터에 'S'가 있으면 새로운 대화의 시작을 의미한다. 대화를 시작하면 첫 번째 발화문은 A, 두 번째 발화문은 B, 세 번째 발

화문은 A, 네 번째 발화문은 B로 저장한 다음 각 A와 B에 해당하는 문장들을 하나의 문자열로 병합한다. 이 결과 1개의 대화는 A와 B로 구성된 SL 대화문으로 데이터가 구성된다. [표 4-3]의 Algorithm 5는 대화 데이터를 1-turn CL 대화문으로 분류하는 알고리즘을 나타낸다.

[표 4-3] Algorithm 5 : Conversation classification : 1-turn CL

Algorithm 5. Conversation classification : 1-turn CL

Input: CDL:ConversationDataList

Output: 1tCLL:1-turnConversationLevelList

```

1  READ CDL;
2  for each standard_s in CDL do
3    if standard_s[0] = 'S' then
4      number ← 1;
5    end if
6    if number%2 = 0 then
7      AB[1] ← standard_s[1];
8      number ← 1;
9      append AB to 1tCLL;
10   AB ← [“,“];
11   else if number%2 = 1 then
12     AB[0] ← standard_s[1];
13     number ← 2;
14   end if
15 end for
16 return 1tCLL;

```

[표 4-3]의 Algorithm 5와 같이 대화 형식의 데이터는 현재 하나의 Column에 나열되어있다. 이를 1-turn 대화문으로 분류하기 위해 알고리즘을 적용하였다. 하나의 대화는 ‘S’라는 구분자를 가지고 있어서 데이터에 ‘S’가 있으면 새로운 대화의

시작을 의미한다. 대화를 시작하면 첫 번째 발화문은 1행 A, 두 번째 발화문은 1행 B, 세 번째 발화문은 2행 A, 네 번째 발화문은 2행 B로 저장한다. 하나의 대화 문장이 홀수개로 A에서 끝나면 해당 A는 삭제 후 다음 대화 문장을 이어서 진행한다. 이 결과 1개의 대화는 여러 개의 행과 A와 B column으로 구성된 1-turn 대화문으로 데이터가 구성된다. [표 4-4]의 Algorithm 6은 대화 데이터를 multi-turn CL 대화문으로 분류하는 알고리즘을 나타낸다.

[표 4-4] Algorithm 6 : Conversation classification : multi-turn CL

Algorithm 6. Conversation classification : multi-turn CL

Input: CDL:ConversationDataList

Output: mtCLL:multi-turn ConversationLevelList

```

1  READ CDL;
2  for each standard_s in CDL do
3    if standard_s[0] = 'S' then
4      append s_str to mtCLL;
5      s_str ← “;
6    end if
7    s_str ← standard_s[1];
8  end for
9  return mtCLL;

```

[표 4-4]의 Algorithm 6과 같이 대화 형식의 데이터는 현재 하나의 Column에 나열되어있다. 이를 Multi-turn 대화문으로 분류하기 위해 알고리즘을 적용하였다. 하나의 대화는 'S'라는 구분자를 가지고 있어서 데이터에 'S'가 있으면 새로운 대화의 시작을 의미한다. 대화를 시작하면 첫 번째 발화문부터 마지막 발화문을 하나의 문자열로 저장한다. 이 결과 1개의 대화는 1개의 문자열로 구성된 Multi-turn 대화문으로 데이터가 구성된다. 다음 [그림 4-2]와 [표 4-5]는 앞에서 제시한 두 화자 간 대화 데이터를 대화문으로 분류하는 과정의 예시를 나타낸다. 또한 [표 4-6]과 [표 4-7]은 실제 데이터의 예시를 나타낸다.



[그림 4-2] 대화 데이터 예시

[표 4-5] 대화문 분류 과정

Classification	Conversation	Sentence	Contents
SL	A	a1,a2,a3	물고기 키워 본 적 있어요? 온도는 내가 맞춰 놔으니까 내일 오전에 한번만 먹이 주면 되요. 제가 오늘부터 다음주 월요일까지 출장이라서요.
	B	b1,b2,b3	없어요. 먹어본 적 밖에는. 이제 저 물고기까지 봐야 합니까? 그러니까 그 사정을 제가 왜 일일이 다 봐줘야 하나구요.
1-turn CL	AB1	a1,b1	물고기 키워 본 적 있어요? 없어요. 먹어본 적 밖에는.
	AB2	a2,b2	온도는 내가 맞춰 놔으니까 내일 오전에 한번만 먹이 주면 되요. 이제 저 물고기까지 봐야 합니까?
	AB3	a3,b3	제가 오늘부터 다음주 월요일까지 출장이라서요. 그러니까 그 사정을 제가 왜 일일이 다 봐줘야 하나구요.
Multi-turn CL	AB	a1~b3	물고기 키워 본 적 있어요? 없어요. 먹어본 적 밖에는. 온도는 내가 맞춰 놔으니까 내일 오전에 한번만 먹이 주면 되요. 이제 저 물고기까지 봐야 합니까? 제가 오늘부터 다음주 월요일까지 출장이라서요. 그러니까 그 사정을 제가 왜 일일이 다 봐줘야 하나구요.

[표 4-6] 대화 문장 예시

Speaker	Sentence	Contents
A	a1	물고기 키워 본 적 있어요?
B	b1	없어요. 먹어본 적 밖에는.
A	a2	온도는 내가 맞춰 놔으니깐 내일 오전에 한번만 먹이 주면 되요.
B	b2	이제 저 물고기까지 봐야 합니까?
A	a3	제가 오늘부터 다음주 월요일까지 출장이라서요.
B	b3	그러니까 그 사정을 제가 왜 일일이 다 봐줘야 하나구요.

[표 4-7] 대화문 분류 결과

Classification	Conversation	Contents
SL	A	물고기 키워 본 적 있어요? 온도는 내가 맞춰 놔으니깐 내일 오전에 한번만 먹이 주면 되요. 제가 오늘부터 다음주 월요일까지 출장이라서요.
	B	없어요. 먹어본 적 밖에는. 이제 저 물고기까지 봐야 합니까? 그러니까 그 사정을 제가 왜 일일이 다 봐줘야 하나구요.
1-turn CL	AB1	물고기 키워 본 적 있어요? 없어요. 먹어본 적 밖에는.
	AB2	온도는 내가 맞춰 놔으니깐 내일 오전에 한번만 먹이 주면 되요. 이제 저 물고기까지 봐야 합니까?
	AB3	제가 오늘부터 다음주 월요일까지 출장이라서요. 그러니까 그 사정을 제가 왜 일일이 다 봐줘야 하나구요.
Multi-turn CL	AB	물고기 키워 본 적 있어요? 없어요. 먹어본 적 밖에는. 온도는 내가 맞춰 놔으니깐 내일 오전에 한번만 먹이 주면 되요. 이제 저 물고기까지 봐야 합니까? 제가 오늘부터 다음주 월요일까지 출장이라서요. 그러니까 그 사정을 제가 왜 일일이 다 봐줘야 하나구요.

[표 4-6]은 A, B 두 화자 간 대화의 예시 문장을 나타내고, [표 4-7]은 대화를 Algorithm 4~6을 활용하여 세 가지 대화문으로 분류한 결과를 나타낸다. 본 예시를 통해 대화 데이터를 대화문으로 분류하는 과정의 이해를 돕는다.

2. 대화문 감정 예측

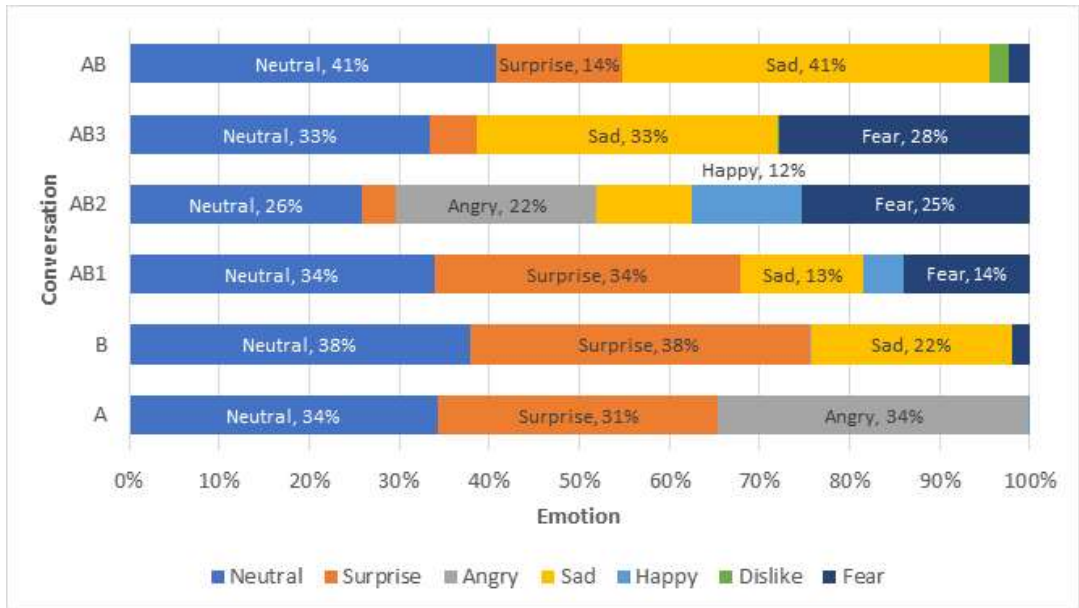
앞에서 분류한 세 가지 형태의 SL, 1-turn CL, multi-turn CL 대화문을 3장에서 제안한 MLER 모델을 적용하여 멀티 레이블 감정을 예측한다. [표 4-8]은 문장의 데이터 셋에 분류된 감정과 멀티 레이블 감정인식 결과를 나타내고 [표 4-9]는 대화문의 멀티 레이블 감정인식 결과를 나타낸다.

[표 4-8] 멀티 레이블 감정인식

Sentence	Emotion	Neutral	Surprise	Angry	Sad	Happy	Dislike	Fear
a1	Neutral	1.00000	0.99899	0.06302	0.00073	0.00988	0.00056	0.02869
b1	Neutral	0.99746	0.99805	0.01268	0.84546	0.03370	0.00090	0.12589
a2	Neutral	0.99139	0.00004	0.00007	0.99990	0.99963	0.02837	0.43851
b2	dislike	0.78417	0.00762	0.99702	0.12438	0.00474	0.99999	0.00014
a3	Neutral	0.15622	0.99807	0.00106	0.95076	0.00000	0.07260	1.00000
b3	angry	0.99995	0.05140	0.98325	0.99985	0.00000	0.00002	0.00491

[표 4-9] 대화문의 멀티 레이블 감정인식

Conversation	Neutral	Surprise	Angry	Sad	Happy	Dislike	Fear
A	1.00000	0.90451	0.99934	0.00478	0.00015	0.00168	0.00049
B	0.99999	0.99934	0.00319	0.58928	0.00003	0.00152	0.05140
AB1	0.99998	1.00000	0.00079	0.39741	0.13606	0.00000	0.41050
AB2	0.99986	0.15178	0.85837	0.41329	0.47272	0.00194	0.97955
AB3	0.99860	0.15673	0.00076	0.99999	0.00017	0.00566	0.83074
AB	0.99999	0.34481	0.00024	0.99971	0.00025	0.05494	0.05589



[그림 4-3] 대화문의 멀티 레이블 감정인식 그래프

[표 4-8]과 같이 멀티 레이블 감정인식 모델을 활용하면 문장별 단일 감정으로 분류되어 있는 데이터 셋을 입력하여 문장에 내재한 다양한 감정을 인식할 수 있고, [표 4-9]와 같이 세 가지로 분류한 대화문을 입력하여 대화문의 다양한 감정도 인식할 수 있다. 그리고 [그림 4-3]을 통해 대화문에 내재한 다양한 감정들이 얼마나 포함되어 있는지 확인할 수 있다.

C. 대화 문맥 연관성

1. 대화 문맥 연관성 분석

본 절에서는 대화 문장과 대화문을 상관관계 분석하고 추출된 상관계수를 통해 대화의 문맥 연관성을 분석하는 과정을 기술한다. 대화 문장 간에 서로 얼마나 밀접하게 선형적 관계를 맺고 있는지를 상관이라고 하며, 상관관계가 어떤 관련성을 가지고 변화하는지 연관성을 분석하는 것을 상관 분석이라고 한다[65]. 분석 방법으로는 상관 분석으로 가장 많이 사용하는 피어슨 상관계수(Pearson Correlation Coefficient)를 활용하였다. 식 (4)는 상관관계를 구하는 식을 나타낸다.

$$Correlation(x,y) = \frac{\sum(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum(x_i - \bar{x})^2 \times \sum(y_i - \bar{y})^2}} \quad (4)$$

식 (4)와 같이 상관계수는 두 변수 x, y가 서로 상관관계에 있을 때, x와 y의 공분산을 x의 분산과 y의 분산을 곱한 제곱으로 나눈다. [표 4-10]은 상관계수에 따른 상관관계 정도를 나타낸다. 상관계수는 -1~1 사이의 값을 가지며 0~0.4일 경우에는 상관관계가 약하다는 것을 뜻한다. 보통 0.4~0.7 사이의 값이면 상관관계가 있다고 간주하고 0.7~1.0 사이의 값이면 강한 상관관계로 간주한다[65].

[표 4-10] 상관계수에 따른 상관도

Correlation coefficient		Degree of correlation
-	+	
-0.9 ~ -1	0.9 ~ 1	매우 강함
-0.7 ~ -0.9	0.7 ~ 0.9	강함
-0.4 ~ -0.7	0.4 ~ 0.7	상관관계가 있음
-0.2 ~ -0.4	0.2 ~ 0.4	약함
0 ~ -0.2	0 ~ 0.2	매우 약함

[표 4-11]은 Python을 활용한 상관관계 분석 과정을 나타낸다.

[표 4-11] 상관관계 분석 과정

```

import pandas as pd
import seaborn as sb
import matplotlib.pyplot as plt

data=pd.read_csv("data/analysis_1.csv")

lst=[data.a1, data.b1, data.a2, data.b2, data.a3, data.b3, data.A, data.B,
      data.AB1, data.AB2, data.AB3, data.AB]
df=pd.DataFrame(lst).T
corr=df.corr(method='pearson') # pearson 상관 분석
data.corr() # 상관분석 결과 출력

# 그림 size 지정
fig, ax = plt.subplots( figsize=(12,12) )

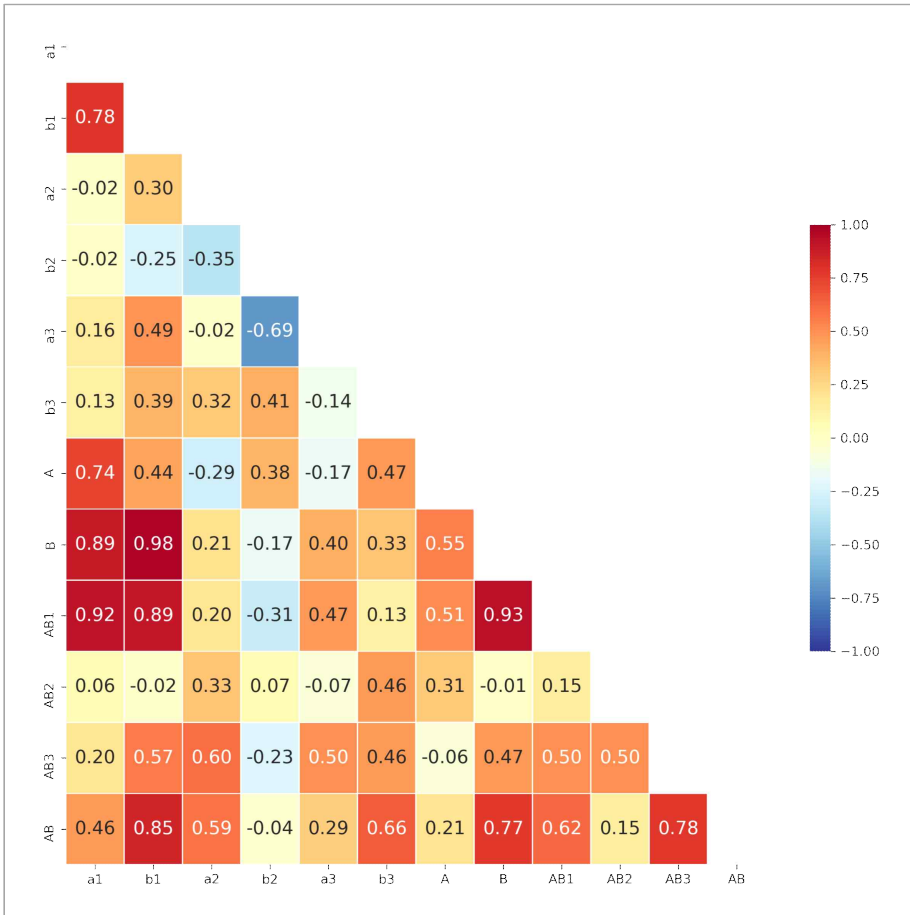
# 삼각형 마스크를 만든다(위 쪽 삼각형에 True, 아래 삼각형에 False)
mask = np.zeros_like(df, dtype=np.bool)
mask[np.triu_indices_from(mask)] = True

# 히트맵을 그린다
sb.heatmap(df,
           cmap = 'RdYlBu_r',
           annot = True, # 실제 값을 표시
           annot_kws = {"size" : 14},
           fmt='.2f',
           mask=mask, # 표시하지 않을 마스크 부분을 지정
           linewidths=.5, # 경계면을 실선으로 구분
           cbar_kws={"shrink": .5},# 컬러바 크기 절반으로 줄임
           vmin = -1,vmax = 1 # 컬러바 범위 -1 ~ 1
           )

plt.savefig('data/analysis_1.png',format='png',dpi=300) # 히트맵 저장

```

[그림 4-4]는 [표 4-11]의 상관관계 분석 과정을 통한 상관관계 Heat map을 표시한 것이다. ‘a1~b3’는 문장, ‘A~AB’는 대화문을 나타낸다. 문장과 대화문의 상관관계 결과를 분석해보면 면의 색에 따라 빨간색이 진해질수록 양의 상관관계가 강해지고 파란색이 진해질수록 음의 상관관계가 강해진다. 숫자는 상관계수를 나타낸다.



[그림 4-4] 상관관계 Heat map

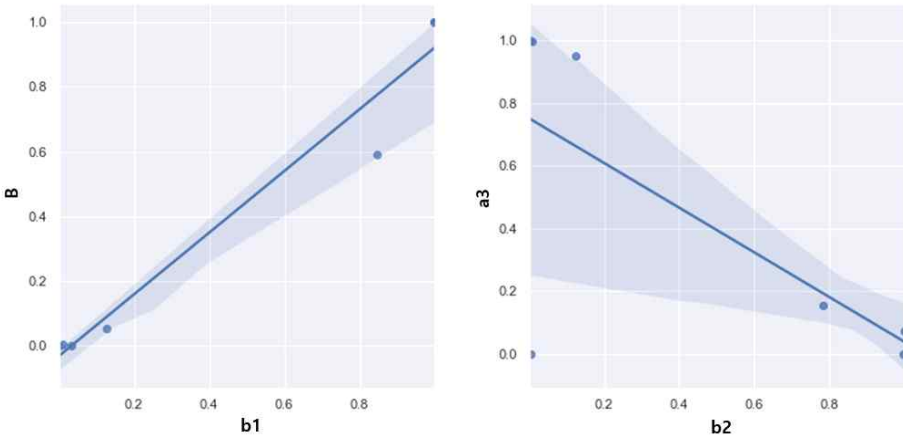
[표 4-12]는 상관관계가 강한 상관계수 0.7 이상인 문장과 대화문을 순위별로 나타낸 것이다. 상관관계 순으로 살펴보면 [b1, B]의 상관관계가 가장 높은 것은 화자 B의 첫 번째 문장이 화자의 감정에 가장 크게 영향을 미친다는 것을 알 수 있다. 다음으로 [a1, AB1]의 상관관계가 높다는 것은 첫 번째 1-turn CL에서는 화자 A의 감정이 영향을 미친다는 것을 알 수 있다. 분석 결과를 통해 대화에서는 화자

의 이전 감정이 다음 대화에 영향을 주고 화자 간 감정이 서로 교류가 되는 것을 알 수 있다.

[표 4-12] 강한 상관관계 순위

순위	Sentence & Conversation		Correlation
1	b1	B	0.98129
2	a1	AB1	0.91591
3	b1	AB1	0.89187
4	a1	B	0.88666
5	b1	AB	0.84712
6	a1	b1	0.78303
7	a1	A	0.73601

[그림 4-5]는 양의 상관관계와 음의 상관관계를 나타내고 있다. 가장 큰 양의 상관관계는 [b1, B]의 관계이고 그래프가 오른쪽 위로 올라가는 형태이다. 가장 작은 음의 상관관계는 [b2, a3]의 관계이고 그래프가 오른쪽 아래로 내려가는 형태이다. 상관관계 수치에 따라 그래프의 각도가 달라진다.



[그림 4-5] 양의 상관관계와 음의 상관관계

2. 대화 문맥 연관성 적용

본 절에서는 대화 문맥 연관성 분석 방법을 전체 대화문에 적용하여 상관관계를 통해 대화 문맥 연관성 식을 도출하고 이를 적용하는 과정에 대해 기술한다. 전체 대화 데이터 셋을 문장(Sentence)과 대화문(Conversation)으로 구분하여 대화 문맥 연관성 분석 방법을 적용한다. 문장과 대화문의 상관관계(*corr*) 중에서 강한 상관관계인 0.7 이상만 추출하고 문장의 순서에 따른 대화 문맥 가중치(Correlation Weight : *corr W*)가 0.4 이상인 상관계수만 합산하여 대화 문맥 연관성(Correlation of Conversation Context : *corr CC*)을 구한다. [표 4-13]은 전체 대화 데이터 셋의 문맥 연관성을 상관관계가 강한 순으로 1위부터 7위까지 나타낸 것이다.

[표 4-13] 전체 대화 문맥의 연관성

순위	Sentence & Conversation		Correlation of entire conversation context
1	b2	AB2	0.99833
2	a1	AB	0.95072
3	b1	AB1	0.88419
4	a2	a3	0.81706
5	b2	b3	0.80495
6	b3	AB2	0.77948
7	b2	a3	0.73910

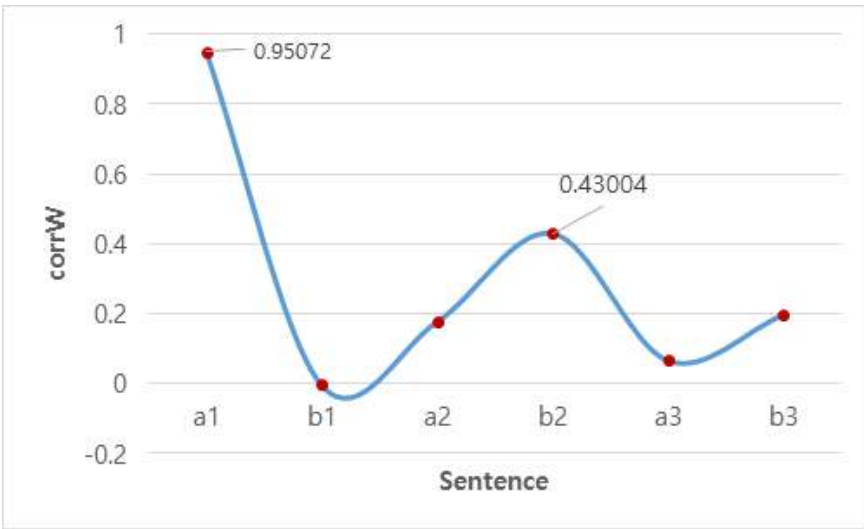
[표 4-13]과 같이 [b2, AB2]의 상관관계가 가장 높은 것으로 보아 두 번째 1-turn CL에서는 화자 B의 대화가 감정에 가장 크게 영향을 미친다는 것을 알 수 있다. 다음으로 [a1, AB]의 상관관계가 높다는 것은 화자 A의 첫 번째 대화가 전체 대화문의 감정에 큰 영향을 미친다는 것을 알 수 있다. 전체 대화 문맥의 연관성 분석을 통해 문장의 순서에 따른 대화 문맥 가중치(*corr W*)를 도출한다. [표 4-14]는 10,000개의 전체 대화 데이터 셋에서 문장 순서에 따른 대화 문맥 가중치를 도출하여 문장별 상관계수의 증위값을 순위별로 나타낸 것이다. 증위값을 사용

한 이유는 전체 데이터 중 가운데에 있는 수로 극단적인 값이 있는 경우 평균값보다 유용하기 때문이다.

[표 4-14] 대화 문맥 가중치

순위	Sentence	Correlation Weight(<i>corr W</i>)
1	a1	0.95072
2	b2	0.43004
3	b3	0.19774
4	a2	0.17732
5	a3	0.06580
6	b1	-0.00339

[표 4-14]와 같이 대화 문맥 가중치를 분석해본 결과 대화의 첫 문장이 전체 대화문의 감정과 가장 연관성이 크다는 것을 알 수 있다. 또한 동일한 화자의 감정은 다음 대화에 영향을 미치며, 상대방 화자의 감정에 따른 감정 변화가 있다는 것을 알 수 있다. [그림 4-6]은 문장 순서별 대화 문맥 가중치 그래프를 나타낸다.



[그림 4-6] 문장 순서별 대화 문맥 가중치 그래프

[그림 4-6]과 같이 전체 대화문과 ‘a1, b2’ 문장은 대화 문맥 연관성이 있음을 확인할 수 있다. ‘a1’ 문장의 대화 문맥 연관성이 가장 크고 다음으로는 ‘b2’ 문장의 대화 문맥 연관성이 있는 것을 볼 수 있다. ‘a2, a3, b3’ 문장은 전체 대화문과의 감정 연관성이 매우 약하고 ‘b1’ 문장은 음의 상관관계가 있음을 볼 수 있다. 따라서 대화 문맥 가중치는 대화 문맥 연관성이 있는 ‘a1, b2’ 문장에 적용한다.

대화 문맥 연관성은 상관계수를 활용하여 *corrCC*(Correlation of Conversation Context)로 정의한다. *corrCC*는 대화문 *i, j*가 있을 때 두 대화문의 상관계수(*corr_{ij}*)와 대화 문맥 가중치(*corr W_i*)를 합산하여 도출한다. 문장 *i*의 대화 문맥 연관성(*corrCC*)을 구하는 방법은 다음과 같다. 두 대화문 *i, j*의 상관계수(*corr_{ij}*)가 0.7 이상이면 문장 *j*의 멀티 레이블 감정(*Me_j*)에 상관계수(*corr_{ij}*)를 곱한다. 그리고 대화 문맥 가중치(*corr W_i*)가 0.4 이상이면 합산하여 *Me_i'*으로 저장한다. 문장 *i*의 멀티 레이블 감정(*Me_i*) 인식 결과와 대화 문맥 연관성(*corrCC_{ij}*)을 적용한 감정(*Me_i'*) 인식 결과를 비교하기 위해 원래 값을 변경하지 않고 추가하여 저장한다. 상관계수(*corr_{ij}*)가 0.7 이하인 경우와 대화 문맥 가중치(*corr W_i*)가 0.4 이하인 경우에는 적용하지 않으므로 0으로 설정하고 계산하지 않는다. 식 (5)는 대화 문맥 연관성을 도출하는 식을 나타낸다.

$$\begin{aligned}
 Me'_i &= Me_i + Me_j \times corr_{ij} + corr W_i \\
 corr &= \begin{cases} corr : \text{if } corr \geq 0.7 \\ 0 : \text{otherwise} \end{cases} \\
 corr W &= \begin{cases} corr W : \text{if } corr W \geq 0.4 \\ 0 : \text{otherwise} \end{cases} \\
 corrCC &= [corr, corr W]
 \end{aligned}
 \tag{5}$$

Me : Multi - Label Emotion
corr : correlation
corr W : correlation Weight
corrCC : correlation of Conversation Context

앞에서 도출한 대화 문맥 연관성을 전체 문장과 대화문에 적용한다. [표 4-15]의 Algorithm 7은 대화 문맥 연관성을 적용하는 알고리즘을 나타낸다.

[표 4-15] Algorithm 7 : Apply *corrCC*

Algorithm 7. Apply *corrCC*(Correlation of Conversation Context)

Input: SenConList:sentence & conversation multi-label emotion list

Output: corrCCList:apply *corrCC* SenConList

```

1  READ SenConList;
2  for each sentence in SenConList do
3    for each conversation in SenConList do
4      if  $corr \geq 0.7$  or  $corrW \geq 0.4$  then
5         $Me_{sen}' \leftarrow Me_{sen} + Me_{con} \times corr + corrW$ 
6      end if
7    end for
8  end for
9  corrCCList  $\leftarrow$  SenConList;
10 return corrCCList;
```

[표 4-15]의 Algorithm 7과 같이 문장과 대화문의 상관관계를 분석하여 상관계수 (*corr*)가 0.7 이상이거나 대화 문맥 가중치(*corrW*)가 0.4 이상이면 대화문의 멀티 레이블 감정인식 결과에 상관계수를 곱한 다음 대화 문맥 가중치를 합산하여 그 값을 문장의 멀티 레이블 감정에 추가하여 저장한다. 따라서 대화문의 감정인식에 상관계수와 대화 문맥 가중치를 합산한 대화 문맥 가중치를 적용하면 문맥에 따른 대화의 감정을 인식할 수 있어서 감정인식의 성능향상이 가능하다.

V. 실험 및 평가

본 장에서는 제안한 비정형 대화 텍스트 처리 방법과 멀티 레이블 감정인식 (MLER) 모델, 대화 문맥의 연관성을 적용한 MLER 모델을 실험 및 평가한다. 실험에서 사용한 데이터 셋에 대하여 설명하고 제안한 방법의 효용성을 입증하기 위해 분류성능평가지표인 정확률(Accuracy), 재현율(Recall), 정밀도(Precision), F1 점수(F1-score)를 측정하여 성능을 평가한다[66]. 또한 멀티 레이블 분류성능평가지표인 Macro-F1, Micro-F1을 측정하여 성능을 평가한다.

본 연구에서 실험에 사용한 환경은 [표 5-1]과 같다.

[표 5-1] 개발 환경[66]

Division		Contents
HW	CPU	Intel(R) Core(TM) i7-9700K CPU @ 3.60GHz
	GPU	NVIDIA GeForce RTX 2060
	RAM	64.0 GB
	HDD	2TB
SW	OS	Window 10 Pro
	Program Language	Python

A. 실험 데이터

본 절에서는 제안한 모델을 실험하기 위해 사용한 데이터 셋에 관해 기술한다. 사용한 데이터는 비정형 대화 텍스트의 감정을 인식하기 위해 AI- HUB에서 제공하는 ‘한국어 감정 정보가 포함된 단발성 대화 데이터 셋(Single)’과 ‘한국어 감정 정보가 포함된 연속적 대화 데이터 셋(Continuous)’을 사용하였다[67].

두 가지 데이터 셋은 비정형 대화 텍스트인 SNS, 댓글, 대화에 대한 웹 크롤링을 시행하고 선정한 문장을 단일 감정 클래스로 레이블링하여 구축한 것이다. Single은 ‘Sentence(문장)’와 ‘Emotion(감정)’으로 구분되어 있고 총 38,594개의 문장으로 구성되어 있다. 문장별로 ‘공포, 놀람, 분노, 슬픔, 중립, 행복, 혐오’의 7가지 감정 클래스로 분류되어 있다. Continuous는 ‘dialog(대화)’와 ‘발화’, ‘감정’으로 구분되어 있다. 총 10,000개의 대화로 이루어진 55,600개의 문장으로 구성되어 있고 7가지 감정 클래스로 분류되어 있다. [그림 5-1]과 [그림 5-2]는 본 논문에서 사용한 데이터 셋을 나타낸다.

	Sentence	Emotion
0	언니 동생으로 부르는게 맞는 일인가요..??	공포
1	그냥 내 느낌일뿐겠지?	공포
2	아직너무조기라서 그런거죠?	공포
3	유치원버스 사고 났다던데	공포
4	근데 원래이런거맞나요	공포

[그림 5-1] Single Dataset

dialog #	발화	감정
0	S 아 진짜! 사무실에서 피지 말라니깐! 간접흡연이 얼마나 안좋은데!	분노
1	NaN 그럼 직접흡연하는 난 얼마나 안좋겠니? 안그래? 보면 꼭... 지 생각만 하고	혐오
2	NaN 손님 왔어요.	중립
3	NaN 손님? 누구?	중립
4	NaN 몰라요. 팀장님 친구래요.	중립
5	NaN 내 친구? 친구 누구?	중립
6	NaN 그걸 내가 어떻게 알아요!	분노
7	S 그래서... 무슨 일 해?	중립
8	NaN 그냥 방송일 조금.	중립

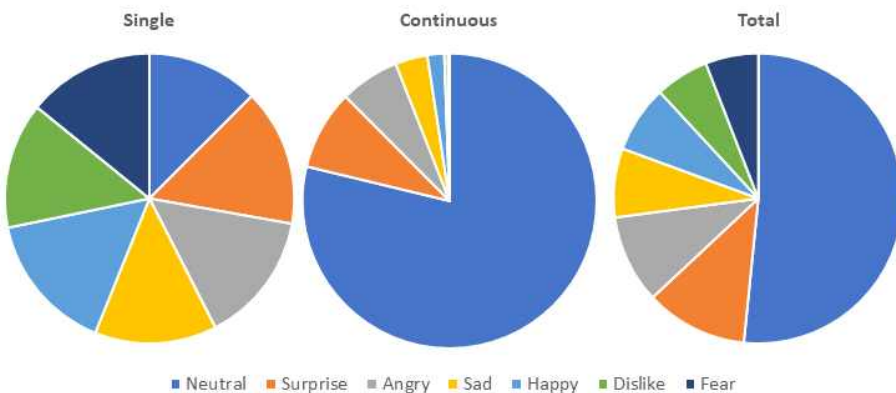
[그림 5-2] Continuous Dataset

[표 5-2]는 감정 클래스별 데이터의 수를 나타낸다. Single보다 Continuous의 데이터 불균형이 더 심한 것을 확인할 수 있다. 본 논문에서는 감정인식을 학습하는데 충분한 학습 데이터를 구축하기 위해 두 가지 데이터 셋을 병합하여 사용한다.

[표 5-2] 감정 클래스별 데이터의 수

Emotion Dataset	Neutral	Surprise	Angry	Sad	Happy	Dislike	Fear	Total
Single (%)	4,830 (12.5)	5,898 (15.3)	5,665 (14.7)	5,267 (13.6)	6,037 (15.6)	5,429 (14.1)	5,468 (14.2)	38,594 (100)
Continuous (%)	43,786 (78.8)	4,866 (8.8)	3,628 (6.5)	1,972 (3.5)	1,030 (1.9)	220 (0.4)	98 (0.2)	55,600 (100)
Total (%)	48,616 (51.6)	10,764 (11.4)	9,293 (9.9)	7,239 (7.7)	7,067 (7.5)	5,649 (6.0)	5,566 (5.9)	94,194 (100)

[표 5-2]와 같이 두 데이터 셋을 합한 Total 역시 감정 클래스별 데이터의 수가 불균형 데이터로 구성되어 있다. 감정 클래스별 데이터의 비율을 보면 가장 많은 감정은 ‘Neutral’ 클래스로 48,616개이며 51.6%를 차지하고 있다. 가장 적은 감정은 ‘Fear’ 클래스로 5,566개이며 5.9%를 차지하고 있다. 본 논문에서는 데이터 불균형 문제를 해결하기 위해 데이터의 수가 가장 적은 ‘Fear’ 클래스에 맞추어 모든 감정 클래스의 데이터를 5,500개로 균형을 맞추었다. 7가지 감정별 5,500개의 데이터를 랜덤으로 추출한 총 38,500개의 데이터를 처리 데이터(Process)로 사용하였다. [그림 5-3]은 데이터 셋에 따른 감정 클래스의 비율을 나타낸다. Single은 균형 데이터에 가깝고 Continuous는 불균형 데이터라는 것을 한눈에 파악할 수 있다.



[그림 5-3] 데이터 셋에 따른 감정 클래스의 비율

B. 실험 평가 및 분석

본 절에서는 제안한 방법의 우수성을 평가하고 결과를 분석하기 위해 멀티 레이블 감정인식 모델과 대화 문맥의 연관성 적용 방법을 실험 및 성능 평가하고, 활용한 모델과 결과 분석을 통한 성능 평가와 비교 평가에 관하여 기술한다.

1. 실험 평가 방법

제안한 방법의 평가 방법으로 분류성능평가지표인 정확률(Accuracy), 재현율(Recall), 정밀도(Precision), F1 점수(F1-score)를 사용하고, 멀티 레이블의 분류 평가 방법으로는 행과 클래스의 F1-score의 평균을 비교하는 Macro-F1과 Micro-F1을 사용하여 성능을 비교한다.

본 논문에서 제안한 방법의 성능 평가 방법으로는 원본 데이터(Original)와 처리(Process) 데이터의 성능을 평가하고, Word2Vec과 KoBERT의 문장 임베딩 성능을 평가한다. 또한 LSTM, CNN-LSTM, Attention, Transformer 모델로 구축한 멀티 레이블 감정인식의 성능을 평가하고, 감정 클래스별 성능을 평가한다. 비교 평가 방법으로는 멀티 레이블 데이터 셋을 활용하여 멀티 레이블 분류 결과를 비교 평가하고, 대화 문맥 연관성(*corrCC*)을 적용한 멀티 레이블 감정인식(MLER) 모델의 결과를 비교 평가한다. [표 5-3]은 분류성능평가지표로 성능을 확인하기 위해 모델이 예측한 레이블과 실제 레이블을 기반으로 구성된 레이블 e 의 혼동행렬(Confusion Matrix)을 나타낸다.

[표 5-3] Confusion Matrix of Emotion Label e

Emotion e		Model Prediction	
		Positive	Negative
True Label	Positive	TP_e	FN_e
	Negative	FP_e	TN_e

[표 5-4]는 Confusion Matrix에 표시된 TP_e , FN_e , FP_e , TN_e 용어에 관한 설명을 나타낸다. 식 (6)은 4가지 분류성능평가지표를 계산하기 위한 식을 나타낸다.

[표 5-4] Confusion Matrix 용어 설명[66]

TP_e	True Positive	실제 Positive를 Positive로 예측(정답)
FN_e	False Negative	실제 Positive를 Negative로 예측(오답)
FP_e	False Positive	실제 Negative를 Positive로 예측(오답)
TN_e	True Negative	실제 Negative를 Negative로 예측(정답)

$$Accuracy_e = \frac{TP_e + TN_e}{TP_e + TN_e + FP_e + FN_e}$$

$$Recall_e = \frac{TP_e}{TP_e + FN_e}$$

(6)

$$Precision_e = \frac{TP_e}{TP_e + FP_e}$$

$$F1-score_e = 2 \times \frac{Precision_e \times Recall_e}{Precision_e + Recall_e}$$

식 (6)과 같이 Accuracy는 실제 데이터와 예측 데이터가 얼마나 같은지를 판단하는 지표로서 직관적인 모델 예측 성능을 나타내는 평가지표이다. 재현율(Recall)은 ‘실제 정답을 얼마나 맞혔는가’를 나타내고 정밀도(Precision)는 ‘예측값이 얼마나 정확한가’를 나타낸다. F1 점수(F1-score)는 정밀도와 재현율의 조화평균으로 주로 분류 클래스 간의 데이터 불균형이 심각할 때 사용한다.

또한 멀티 레이블의 분류 평가 방법은 데이터가 포함되었다고 예측되는 클래스의 레이블 또는 데이터가 각 클래스에 포함될 확률을 0~1 사이의 수치로 계산하여

평가하고 이진 분류를 클래스 수만큼 반복한다. Mean-F1은 행 데이터 단위로 F1-score를 계산한 평균값을 비교하는 방법이고, Macro-F1은 행 데이터 단위로 F1-score를 계산한 평균값으로 클래스의 빈도수와 관계없이 똑같은 가중치를 주고 평균을 내는 방법이다. 그리고 Micro-F1은 행 데이터*클래스의 각 쌍에 대해 TP, TN, FP, FN 중 어디에 해당하는지 카운트하고 이에 근거해 F 점수를 계산하여 클래스별 불균형 문제를 해결하기 위해 빈도수에 따라 가중치를 두고 평균을 내는 방법이다. 본 논문에서는 멀티 레이블의 평가 방법으로 Macro-F1과 Micro-F1을 사용한다. 식 (7)은 Macro-F1과 Micro-F1을 계산하기 위한 식을 나타낸다.

$$Macro-F1_e = \frac{1}{N} \sum_{i=0}^N F1_e \tag{7}$$

$$Micro-F1_e = 2 \times \frac{Micro-precision_e \times Micro-recall_e}{Micro-precision_e + Micro-recall_e}$$

2. 실험 결과 분석

실험 결과 분석으로는 본 논문에서 제안한 데이터 전처리와 문장 임베딩, 멀티레이블 감정인식 모델(MLER)의 성능을 평가하고, MLER 모델과 대화 문맥 연관성을 적용한 감정인식 모델을 비교 평가한 결과를 기술한다.

데이터 셋의 전처리 과정으로 비정형 대화 텍스트를 처리하고 형태소 분석하였다. 실험 결과 성능이 가장 좋은 방법으로 문장 임베딩은 KoBERT를 활용하여 문장을 벡터화하였고, 생성된 벡터와 감정 클래스로 Attention 모델을 통해 MLER 모델을 구축하였다. 감정 클래스별 5,500개의 처리 데이터(Process)는 8:2의 비율로 train에 4,400개, test에 1,100개를 사용하였다. 원본 데이터(Original)의 경우 감정 클래스별 5,500개 미만인 감정 클래스는 그대로 사용하였다. 또한 MLER 모델을 대화문에 적용하여 대화 문맥의 연관성을 도출하고 이를 적용한 대화문의 멀티레이블 감정을 인식하였다.

a. 성능 평가

제안한 방법의 성능 평가는 총 4단계로 구성된다. 1단계는 데이터 전처리와 모델의 학습 횟수에 따른 성능을 평가하고 2단계는 Word2Vec과 KoBERT의 문장 임베딩 성능을 평가한다. 다음으로 3단계는 LSTM, CNN-LSTM, Attention, Transformer 모델로 구축한 감정인식 성능을 평가하고 마지막으로 4단계는 감정 클래스별 감정인식 성능을 평가한다.

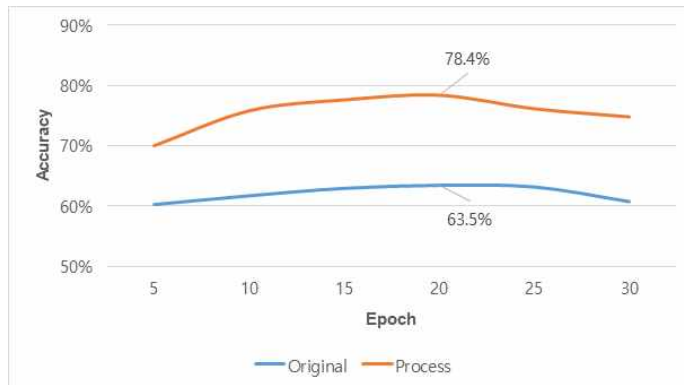
먼저 1단계로 데이터 전처리와 모델의 학습 횟수에 따른 성능을 평가하였다. 원본 데이터(Original, 감정 클래스별 데이터 수가 불균형)로 학습한 결과와 처리 데이터(Process, 감정 클래스별 데이터 수가 같고 전처리를 수행함)로 학습한 결과를 비교하였다. KoBERT 문장 임베딩과 Attention 모델로 멀티레이블 감정을 학습하여 MLER 모델을 구축하였다. 학습 횟수(Epoch)는 30회로 설정한 다음 Accuracy로 성능을 비교하였다. [표 5-5]는 Epoch에 따른 MLER 모델의 성능을 비교하여 나타낸다.

[표 5-5] Epoch에 따른 MLER 성능 (단위 : %)

Epoch	5	10	15	20	25	30
Original	60.2	61.7	63.0	63.5	63.3	60.8
Process	69.9	75.8	77.6	78.4	76.2	74.8

Model : KoBERT + Attention

[표 5-5]와 같이 원본 데이터(Original)보다 처리 데이터(Process)로 학습한 모델의 성능이 전체적으로 향상된 것을 볼 수 있다. MLER 모델의 Epoch가 20회일 때 성능이 가장 좋았으며 이때 Accuracy는 Original은 63.5%, Process는 78.4%로 14.9% 정도 성능이 향상되었다. 따라서 감정인식의 성능향상을 위해서는 비정형 대화 텍스트를 처리하고 클래스의 균형을 맞추는 방법과 모델의 Epoch가 중요하다는 것을 알 수 있다. [그림 5-4]는 Epoch에 따른 MLER 모델의 성능 그래프를 나타낸다.



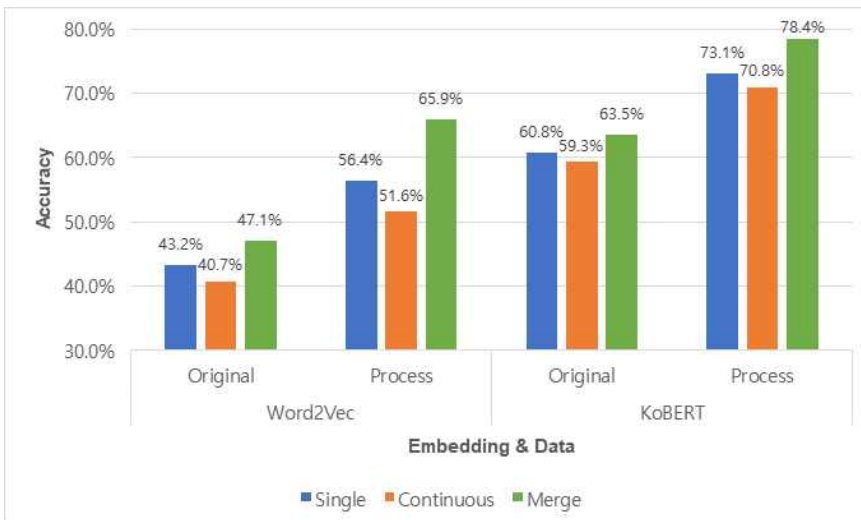
[그림 5-4] Epoch에 따른 MLER 성능 그래프

다음으로는 2단계로 Word2Vec과 KoBERT의 문장 임베딩 성능을 평가한다. 다양한 방법으로 비교하기 위해 단발성 대화 데이터 셋(Single)과 연속적 대화 데이터 셋(Continuous), 그리고 두 가지를 병합한 데이터 셋(Merge)을 구분하여 실험하였다. 또한 전처리 전 데이터를 원본 데이터(Original)로 정의하고 전처리한 데이터를 처리 데이터(Process)로 정의하였다. 세 가지 데이터 셋의 Original과 Process로 문장 임베딩하고 Attention 모델로 학습한 MLER 모델의 Accuracy를 비교하였다. [표 5-6]은 임베딩과 데이터에 따른 MLER 모델의 성능을 나타낸 것이다.

[표 5-6] 임베딩&데이터에 따른 MLER 성능 (단위 : %)

Embedding & Data Data set	Word2Vec	KoBERT
	Original / Process	Original / Process
Single	43.2 / 56.4 (↑ 13.2)	60.8 / 73.1 (↑ 12.3)
Continuous	40.7 / 51.6 (↑ 10.9)	59.3 / 70.8 (↑ 11.5)
Merge	47.1 / 65.9 (↑ 18.8)	63.5 / 78.4 (↑ 14.9)

[표 5-6]과 같이 세 가지 데이터 셋에서 모두 Original 보다 Process로 학습한 모델의 성능이 더 향상되었고 Merge의 성능이 가장 좋은 것을 확인할 수 있다. 또한 Continuous의 클래스 불균형이 더 심하므로 Single의 성능이 더 좋은 것을 볼 수 있다. 문장 임베딩 결과는 Word2Vec 보다 KoBERT를 사용했을 때 성능이 더욱 향상되었다. 제한된 데이터만 학습한 방법보다는 더욱 많은 양의 데이터가 사전 학습된 문장 임베딩의 성능이 더 좋다는 것을 알 수 있다. 또한 전처리 후 성능은 제한된 데이터만 학습했을 때 미세하게 더 향상된 것을 볼 수 있다. 따라서 더 정확한 감정인식과 성능향상을 위해 데이터의 균형을 맞추는 방법과 비정형 대화 텍스트의 유형을 파악하고 그에 맞는 처리 방법이 필요하다. [그림 5-5]는 임베딩과 데이터에 따른 MLER 모델의 성능을 비교한 그래프를 나타낸다.



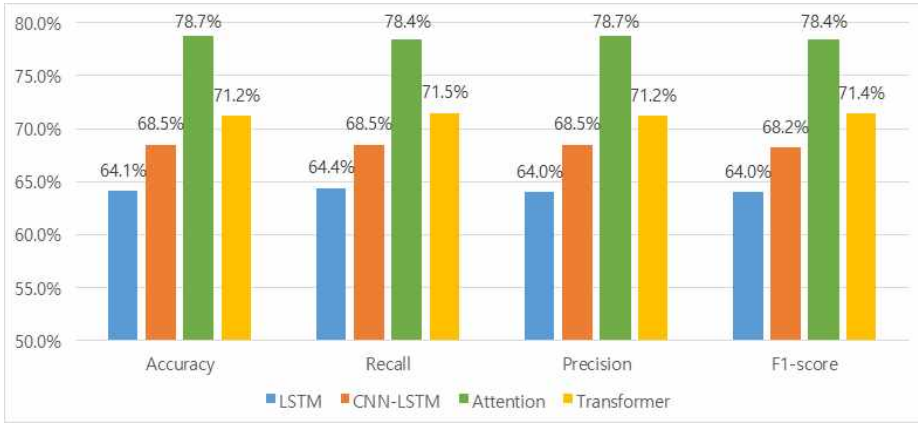
[그림 5-5] 임베딩&데이터에 따른 MLER 성능 그래프

다음으로는 3단계로 LSTM, CNN-LSTM, Attention, Transformer로 구축한 MLER 모델의 대표 감정으로 성능을 평가한다. Merge 데이터를 전처리하고 KoBERT로 문장 임베딩 한 다음 4가지 딥러닝 모델을 각각 학습시킨 후 비교 실험하였다. 전체 데이터에서 train과 test는 8:2의 비율로 구성하였고 train을 통한 학습과 test를 통한 평가를 진행하였다. 평가 방법은 분류된 감정 클래스를 중심으로 Accuracy, Precision, Recall, F1-score를 사용하였다. [표 5-7]은 4가지 딥러닝 모델별 MLER의 Accuracy로 성능을 비교한 결과를 나타낸다.

[표 5-7] 모델별 MLER 성능 (단위 : %)

Model \ Evaluation	LSTM	CNN-LSTM	Attention	Transformer
Precision	64.1	68.5	78.7	71.2
Recall	64.4	68.5	78.4	71.5
F1-score	64.0	68.5	78.7	71.2
Accuracy	64.0	68.2	78.4	71.4

[표 5-7]과 같이 제시한 4가지 딥러닝 모델 중에서 Attention으로 구축한 MLER 모델의 Accuracy가 78.4%로 가장 좋은 성능을 보여주었다. LSTM과 CNN-LSTM은 Accuracy가 비슷하였지만, LSTM만 단독으로 사용한 모델보다는 CNN을 결합한 CNN-LSTM 모델의 성능이 4.2% 정도 향상된 것을 볼 수 있었다. Transformer의 성능이 Attention보다 낮은 원인은 학습 데이터의 양이 부족했기 때문으로 파악되었다. [그림 5-6]은 4가지 딥러닝 모델별 MLER의 Accuracy 성능을 비교한 그래프를 나타낸다.



[그림 5-6] 모델별 MLER 성능 그래프

마지막으로는 4단계로 감정 클래스별 MLER의 성능을 평가한다. 성능이 가장 좋은 Attention 모델의 감정 클래스별 Precision, Recall, F1-Score를 비교하여 성능을 평가한다. [표 5-8]은 감정 클래스별 MLER의 성능을 나타낸 것이다.

[표 5-8] 감정 클래스별 MLER 성능 (단위 : %)

Model \ Evaluation	Neutral	Surprise	Angry	Sad	Happy	Dislike	Fear
Precision	54.8	67.5	71.6	75.9	98.4	91.3	91.3
Recall	56.1	70.2	74.5	80.1	95.6	81.6	90.0
F1-score	54.8	68.9	73.1	78.7	97.1	85.7	90.0

Model : KoBERT + Attention

[표 5-8]과 같이 감정 클래스별 MLER 결과는 다른 감정보다 ‘Happy’ 감정 클래스의 분류 성능이 가장 좋았다. 이는 7가지 감정을 긍·부정으로 분류했을 때 ‘Happy’ 클래스는 긍정으로, 다른 감정들은 부정으로 분류되는 특징이 있기 때문으로 해석된다. 또한 ‘Surprise’ 클래스는 긍·부정의 감성을 모두 포함하고 있어서 다른 감정보다 성능이 낮게 분류되었다. 분류가 모호한 ‘Neutral’ 클래스는 성능이 가장 낮았다. 이러한 문제점을 해결하기 위해 더 정확한 감정인식을 위한 연구가 필요하다. 따라서 데이터의 불균형을 해결하고 비정형 대화 텍스트의 유형을 파악하여 처리한 다음 문장을 멀티 레이블 클래스로 분류하여 감정을 인식하면 문장에

내재한 다양한 감정을 파악할 수 있어서 더 정확한 감정인식이 가능하다. [그림 5-7]은 감정 클래스별 MLER의 성능을 그래프로 나타낸 것이다.



[그림 5-7] 감정 클래스별 MLER 성능 그래프

b. 비교 평가

제안한 방법의 비교 평가는 총 2단계로 구성된다. 1단계는 SemEval-2018 멀티 레이블 데이터 셋을 활용하여 멀티 레이블 감정인식(MLER) 모델의 멀티 레이블 분류 결과를 비교 평가한다. 2단계는 대화 문맥 연관성(*corrCC*)을 적용한 MLER 모델을 비교 평가한다.

먼저 1단계로는 제안한 MLER 모델을 검증하기 위해 멀티 레이블의 분류 평가 방법인 Macro-F1과 Micro-F1을 사용하여 비교 평가한다. 멀티 레이블 감정으로 레이블 된 한글 데이터 셋이 없으므로 영어 데이터 셋인 SemEval-2018(Semantic Evaluation-2018)[68]을 한국어로 번역한 후 활용한다. SemEval-2018의 작업 중에서 Ec(Emotion classification)는 트위터 데이터에 대한 11가지 감정 중 하나 이상으로 분류되어 있어서 제안한 방법의 검증을 위해 사용한다. 11가지 감정은 ‘Anger, Anticipation, Disgust, Fear, Joy, Love, Optimism, Pessimism, Sadness, Surprise, Trust’로 분류된 감정을 본 논문에서 사용한 7가지 감정과 매칭하여 ‘Surprise, Anger-Angry, Sadness-Sad, Joy&Love-Happy, Disgust-Dislike, Fear’의 6가지 감정을 비교한다. [표 5-9]는 LSTM, CNN-LSTM, Transformer 모델과

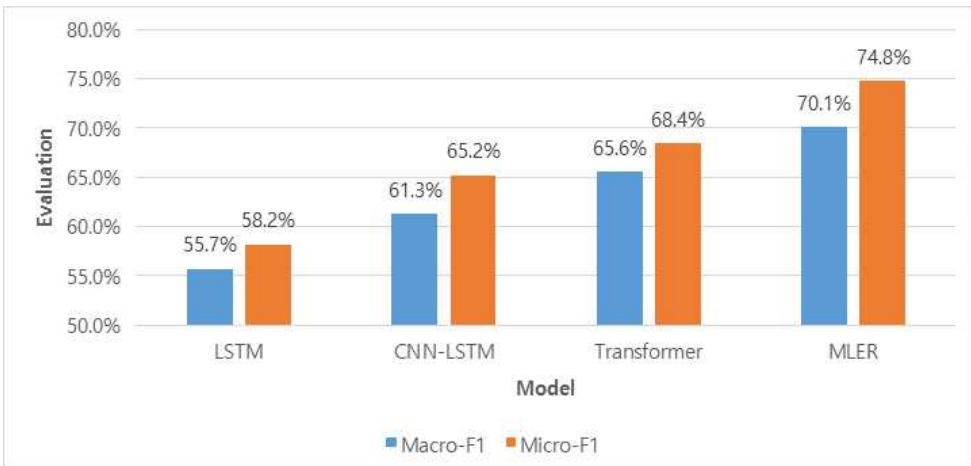
제안한 MLER 모델의 멀티 레이블 분류 결과인 Macro-F1과 Micro-F1을 비교한 것이다.

[표 5-9] 모델별 멀티 레이블 분류 비교 (단위 : %)

Model \ Evaluation	LSTM	CNN-LSTM	Transformer	MLER
Macro-F1	55.7	61.3	65.6	70.1
Micro-F1	58.2	65.2	68.4	74.8

MLER : KoBERT + Attention

[표 5-9]와 같이 모델별 멀티 레이블 분류 성능을 비교한 결과 제안한 MLER 모델의 Macro-F1과 Micro-F1의 성능이 가장 좋은 것을 볼 수 있다. LSTM, CNN-LSTM 모델보다는 제안한 모델의 성능이 16.6%, 9.6% 향상되었고 Transformer 모델보다는 제안한 모델의 성능이 6.4% 향상되었다는 것을 확인할 수 있다. 제안한 모델의 성능이 가장 좋은 이유로는 Attention 모델이 단어들 사이의 의존성을 더욱 잘 파악하기 때문에 대화에서 연관성을 분석하는 데 효과적이기 때문으로 판단된다. 그리고 더 많은 학습 데이터를 수집하여 모델을 개선한다면 성능향상이 가능하다. [그림 5-8]은 모델별 멀티 레이블 감정인식 결과의 비교 그래프를 나타낸다.



[그림 5-8] 모델별 멀티 레이블 분류 비교 그래프

또한 대화 문맥의 연관성을 적용한 MLER 모델을 검증하기 위해 감정과 문장의 유사도를 비교한다. 원본 데이터(Original)와 MLER 모델, 그리고 MLER 모델에 대화 문맥의 연관성을 적용한 MLER+corrCC 모델을 감정과 문장의 유사도를 이용하여 비교한다. 유사도는 감정과 문장 내 단어들의 코사인 유사도의 합으로 구한다. 코사인 유사도란 두 벡터 간 코사인 각도를 이용하여 구할 수 있는 유사도를 의미한다. 두 벡터의 방향이 같은 경우에는 1이 되고, 90도의 각을 이루면 0, 180도의 반대 방향을 가지면 -1이 된다. 식 (8)은 코사인 유사도를 구하는 식을 나타낸다.

$$\text{similarity} = \cos(\theta) = \frac{A \cdot B}{\|A_{vert}\| \|B_{vert}\|} = \frac{\sum_{i=1}^n A_i \times B_i}{\sqrt{\sum_{i=1}^n (A_i)^2} \times \sqrt{\sum_{i=1}^n (B_i)^2}} \quad (8)$$

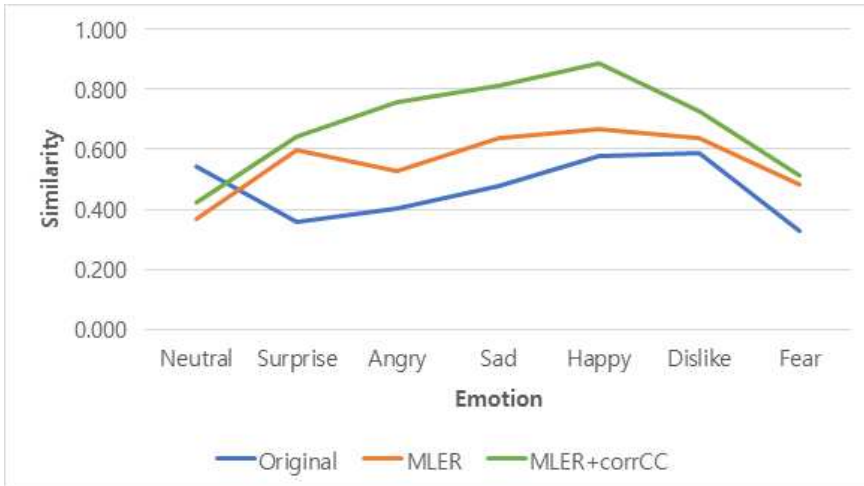
식 (8)을 활용하여 감정과 Original, MLER, MLER+corrCC 모델의 문장 유사도를 비교한다. [표 5-10]은 감정과 문장의 유사도를 비교한 결과를 나타낸다.

[표 5-10] 감정과 문장의 유사도 비교

Model Emotion	Original	MLER	MLER+corrCC
Neutral	0.542	0.366	0.421
Surprise	0.356	0.598	0.641
Angry	0.403	0.529	0.758
Sad	0.479	0.638	0.811
Happy	0.580	0.670	0.884
Dislike	0.587	0.637	0.726
Fear	0.328	0.481	0.515
Average	0.468	0.553	0.680

[표 5-10]과 같이 유사도를 비교한 결과 제안한 MLER+corrCC 모델의 유사도가

가장 높은 것을 볼 수 있고 Original보다 MLER 모델의 유사도가 높은 것을 확인할 수 있다. 이는 Original보다 MLER 모델이 내재한 더 정확한 감정을 인식하며 제안한 MLER+corrCC 모델이 문맥 연관성을 적용하여 가장 정확한 감정을 인식한다는 것을 의미한다. [그림 5-9]는 감정과 문장의 유사도 비교 그래프를 나타낸다.



[그림 5-9] 감정과 문장의 유사도 비교 그래프

다음으로는 대화 예시를 통해 Original과 MLER 모델, MLER+corrCC 모델의 감정인식 결과를 비교한다. [그림 5-10]은 대화 예시를 나타내고, [표 5-11]은 대화 문장의 예시를 나타낸다. 또한 [표 5-12]는 Original에 분류된 감정 클래스와 MLER, 제안하는 방법의 감정인식 결과를 비교하여 나타낸다.



[그림 5-10] 대화 예시

[표 5-11] 대화 문장 예시

Sentence	Contents
a1	어머! 정말? 언제?
b1	아마... 연말쯤? 늦어도 연초에는...
a2	축하해. 근데 왜 지금까지 아무 말도 없었어?
b2	얼마 전에 결정 났어요. 결혼하고 곧장 미국 가기로.
a3	미국 어디?
b3	뉴욕이요. 가서 저는 공부하고. 운이 좋았어요.

[표 5-12] 감정인식 결과 비교

Emotion	MLER	MLER+ <i>corrCC</i>
Surprise	Surprise -Neutral-Angry	Surprise -Happy-Neutral
Neutral	Neutral- Sad -Surprise	Surprise -Sad-Neutral
Neutral	Surprise -Neutral-Fear	Happy -Surprise-Neutral
Neutral	Neutral- Sad -Surprise	Surprise -Neutral-Sad
Neutral	Neutral- Angry -Surprise	Surprise -Neutral-Sad
Happy	Happy -Sad-Neutral	Happy -Neutral-Surprise

[표 5-12]와 같이 Original은 단일 감정으로 분류되어 있고 문장 대부분이 'Neutral'로 분류되어 있지만, MLER 모델을 통해 문장에 내재한 다양한 감정을 인식할 수 있다. 또한 대화 문맥의 연관성 적용 결과 전체 대화문에서 가장 많이 느껴지는 'Surprise'와 'Happy' 감정이 가장 많이 인식된 것으로 분석되었다. 따라서 멀티 레이블 감정 인식(MLER)을 통해 문장에 내재한 다양한 감정을 인식하고 대화 문맥의 연관성을 적용하면 대화문 전체 감정의 흐름에 영향을 받아 문장별 단일 감정으로 인식할 때보다 더 정확한 감정인식을 할 수 있어서 감정인식의 성능 향상이 가능하다.

VI. 결론 및 제언

본 논문에서는 감정인식 성능향상을 위해 대화 문맥의 연관성(*corrCC*)을 적용한 멀티 레이블 감정인식(MLER) 모델을 제안하였다. 데이터 셋으로는 AI-HUB에서 제공하는 ‘한국어 감정 정보가 포함된 단발성 대화 데이터 셋’과 ‘연속적 대화 데이터 셋’을 사용하였다. 감정 클래스별로 데이터의 수가 불균형 데이터로 구성되어 데이터의 수를 5,500개로 균형을 맞춰주었다.

비정형 대화 텍스트의 처리 과정으로 초성체, 중성체, 구어체, 신조어, 줄임말, 비속어 등의 비정형 단어를 처리하였다. Okt를 활용하여 형태소를 분석한 다음, 데이터 셋의 문장 임베딩은 KoBERT로 학습하여 벡터화하였다. 학습된 벡터와 감정 클래스를 LSTM, CNN-LSTM, Attention, Transformer 모델을 활용하여 멀티 레이블 감정을 인식하였고 Attention 모델의 성능이 가장 좋았다. 데이터 셋은 문장별 단일 감정으로 분류되어 있지만, 멀티 레이블 감정인식 방법을 활용하면 문장에 내재한 다양한 감정을 인식할 수 있다. 또한 대화 데이터를 특성에 맞게 SL, 1-turn CL, Multi-turn CL 대화문으로 분류하고 MLER 모델을 적용하여 대화문의 감정을 인식하였다. 대화문 간의 감정 분석을 통해 대화 문맥의 연관성을 도출하고 이를 적용한 감정인식 성능향상이 가능함을 보여주었다.

제안한 모델을 성능 평가한 결과 비정형 대화 텍스트의 포함 여부와 데이터의 불균형에 따라 감정인식 성능에 차이가 있음을 보여주었다. 그리고 전처리한 데이터로 학습한 딥러닝 모델의 Accuracy가 14.9% 향상된 것을 볼 수 있었다. 또한 4가지 딥러닝 모델의 Accuracy, Precision, Recall, F1-score를 비교한 결과 CNN, CNN-LSTM, Transformer보다 Attention 모델이 78.7%로 감정인식에 가장 높은 Accuracy를 보여주었다. 또한 멀티 레이블의 분류 평가 방법인 Macro-F1과 Micro-F1을 비교한 결과 제안한 MLER 모델의 성능이 74.8%로 가장 좋았으며, 원본과 MLER, MLER+*corrCC* 모델의 감정 유사도를 비교해본 결과 제안하는 MLER+*corrCC* 모델이 대화문의 감정을 가장 잘 인식하는 것으로 평가되었다.

제안한 모델을 통해 데이터의 불균형을 해소하고 비정형 대화 텍스트의 유형을 파악한 다음 그에 맞는 처리 방법을 적용하여 문장의 MLER 결과 문장에 내재한 감정의 변화를 볼 수 있어서 더 정확한 감정인식이 가능함을 증명하였다. 또한 대화 문맥 연관성을 적용한 MLER 모델을 활용하면 대화에서 의미를 이해하고 문장 내 대표 감정 이외의 내재한 세밀한 감정들까지 인식할 수 있어서 더 정확한 감정인식이 가능하다. 이러한 세밀한 감정의 변화를 통해 대화에서 감정의 연관성을 파악할 수 있고 감정인식의 성능향상이 가능하다.

또한 데이터가 많아질수록 Accuracy가 상승한다는 것을 확인하였다. 물론 짧은 길이의 텍스트만으로 일상 대화의 감정을 완벽히 분석해 내기에는 한계가 있었다. 하지만 보다 많은 데이터를 수집하여 더 다양한 분야의 대화 데이터를 통해 충분한 학습이 이루어진다면 더 높은 Accuracy의 감정인식이 가능하다. 다양한 분야의 학습을 위해 감정이 라벨링 된 일상 대화 데이터 셋이 요구된다. 추가적인 데이터를 수집하여 모델을 구축한다면 더 높은 감정인식의 성능향상이 가능할 것이다.

본 연구는 대화에서 감정인식의 Accuracy와 성능을 향상하는데 기여할 수 있다. 또한 감정을 정확하게 인식하는 AI를 인간과 직접 상호 작용하는 대화형 인터페이스에 적용하여 상담 치료나 감성 공학, 감성 마케팅, 감성 교육 등 여러 분야에서 활용할 수 있다. 향후 연구로는 멀티 레이블 감정을 구체적으로 반영하여 연속 대화에서 멀티 레이블 감정을 인식하는 딥러닝 모델을 만들고 연속 감정을 예측하는 대화 시스템을 구축할 계획이다.

참고 문헌

- [1] Gyung-Mok Yoon. “Performance Improvement of Movie Recommendation System Using Genetic Algorithm and Adjusting Artificial Neural Network Parameters.” The Journal of KINGComputing, Vol. 10, No. 5, pp. 56-64, 2014.
- [2] Junghan Seo, and Jinho Park. “Data Filtering and Redistribution for Improving Performance of Collaborative Filtering.” The Journal of KINGComputing, Vol. 17, No. 4, pp. 13-22, 2021.
- [3] Guiyoung Son, Wonyoung Lee, Kyungjin Han, and Sunghyun Kyeong. “The study of feature vector generation and emotion recognition using EEG signals.” The Journal of KINGComputing, Vol. 16, No. 2, pp. 72-79, 2020.
- [4] MyungJin Lim, JuHyun Shin. “Continuous Emotion Recognition Method applying Emotion Dimension.” 2021 Spring Conference of KISM, Vol. 10, No. 1, pp. 173-175, 2021.
- [5] Dong-Won Shin, Yeon-Soo Lee, Jung-Sun Jang, and Hae-Chang Rim. “Using CNN-LSTM for Effective Application of Dialogue Context to Emotion Classification.” Annual Conference on Human and Language Technology, pp. 141-146, 2016.
- [6] Ji-Eun Park, Sun-Ju Sohn, and Jin-Hun Sohn. “Classification and Intensity Assessment of Korean Emotion Expressing Idioms for Human Emotion Recognition.” Journal of the Ergonomics Society of Korea, Vol. 31, No. 5, pp. 617-627, 2012.
- [7] Sangwoo Kang, Hongmin Park, and Jungyun Seo. “Emotion Classification of User’s Utterance for a Dialogue System.” Korean journal of cognitive science, Vol. 21, No. 4, pp. 459-480, 2010.
- [8] SangJin Park. “Sensitivity identification method for new words of social media using the naïve bayes classification.” Master’s thesis, Chosun University, 2018.

- [9] Moungho Yi, MyungJin Lim, Hoon Ko, and JuHyun Shin. “Method of Profanity Detection Using Word Embedding and LSTM.” *Mobile Information Systems*, 2021.
- [10] Matthew Purver, and Stuart Battersby. “Experimenting with distant supervision for emotion classification.” In *Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics*, pp. 482–491, 2012.
- [11] Yoon Kim. “Convolutional Neural Networks for Sentence Classification.” *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 1746–1751, 2014.
- [12] Sepp Hochreiter, and Jürgen Schmidhuber. “Long short-term memory.” *Neural computation*, Vol. 9, No. 8, pp. 1735–1780, 1997.
- [13] “From model-centric to data-centric AI development.” *Week 35*, <https://jiho-ml.com/weekly-nlp-35/> (accessed. Nov., 10, 2021)
- [14] Mauajama Firdaus, Hardik Chauhan, Asif Ekbal, and Pushpak Bhattacharyya. “MEISD: a multimodal multi-label emotion, intensity and sentiment dialogue dataset for emotion recognition and sentiment analysis in conversations.” In *Proceedings of the 28th International Conference on Computational Linguistics*, pp. 4441–4453, 2020.
- [15] Yanghoon Kim, Hwanhee Lee, and Kyomin Jung. “AttnConvnet at SemEval-2018 task 1: attention-based convolutional neural networks for multi-label emotion classification.” In *Proceedings of The 12th International Workshop on Semantic Evaluation*, pp. 141 - 145, 2018.
- [16] Iqra Ameer, Noman Ashraf, Grigori Sidorov, and Helena Gómez Adorno. “Multi-label emotion classification using content-based features in Twitter.” *Computación y Sistemas*, Vol. 24, No. 3, pp. 1159–1164, 2020.
- [17] HyeRi Jo, “A study of Human Internality which appear through the Expression of Face : focused on my works.” *Master’s thesis*, Chosun University, 2016.
- [18] 김광수, 김아중, “감정 커뮤니케이션”, 한나래, 2008.
- [19] Douglas M. McNair, Maurice Lorr, and Leo F. Droppleman. “Manual for

- the Profile of Mood States.” Educational and Industrial Testing Services, San Diego. 1971.
- [20] Plutchik, Robert. “Emotions and life: Perspectives from psychology, biology, and evolution.” American Psychological Association, 2003.
- [21] Thayer, Robert E. “The biopsychology of mood and arousal.” Oxford University Press, 1990.
- [22] WonHo Park. “Parting Lyrics Emotion Classification Model using LSTM.” Master’s thesis, Chosun University, 2020.
- [23] HoogSeok Lee. “A thesis on apply emotion to location-based service.” Master’s thesis, Konkuk University. 2013.
- [24] Ekman, Paul. “An argument for basic emotions.” *Cognition & emotion*, Vol. 6, No. 3-4, pp. 169-200, 1992.
- [25] Ekman, Paul, and Daniel Cordaro. “What is meant by calling emotions basic.” *Emotion review*, Vol. 3, No. 4, pp. 364-370, 2011.
- [26] Harmon-Jones, Cindy, Brock Bastian, and Eddie Harmon-Jones. “The discrete emotions questionnaire: A new tool for measuring state self-reported emotions.” *PloS one*, Vol. 11, No. 8, e0159915, 2016.
- [27] Cowen, Alan S., and Dacher Keltner. “Self-report captures 27 distinct categories of emotion bridged by continuous gradients.” *Proceedings of the national academy of sciences*, Vol. 114, No. 38, E7900-E7909, 2017.
- [28] “폴 에크만이 주장하는 미세한 표현.” 윈드폴마인드, <https://wonderfulmind.co.kr/microexpressions-according-to-paul-ekman/> (accessed. Sep., 22, 2021)
- [29] “비정형 데이터”, Wikipedia, <https://ko.wikipedia.org/wiki?curid=1867082> (accessed. Sep., 23, 2021)
- [30] 이재용, 고경훈, 김정숙. “데이터기반행정 정착을 위한 지방자치단체 관리체계 정립방안 연구.” 한국지방행정연구원, 2021. 4.
- [31] Jeongmin Kim, and Jongjin Kook. “Design and Implementation of a LSTM-Based YouTube Malicious Comment Detection System.” *Smart Media Journal*, Vol. 11, No. 2, pp. 18-24, 2022.
- [32] “데이터 마이닝의 한계를 알아본다.” 다음 블로그, <http://blog.daum.net/statsas/17202436> (accessed. Oct., 15, 2021)

- [33] Park DH, Song DH. “Economic and Cultural Devices of Ideal Data Activation.” Internet & Security FOCUS, KISA, 2014.
- [34] Xueqiang Zeng, Qifan Chen, Sufen Chen, and Jiali Zuo, “Emotion label enhancement via emotion wheel and lexicon.” Mathematical Problems in Engineering, 2021.
- [35] MyungJin Lim. “A Study on the Expressionless Emotion Analysis for Improvement of Face Expression Recognition.” Master’s thesis, Chosun University, 2017.
- [36] Aman, Saima, and Stan Szpakowicz. “Identifying expressions of emotion in text.” International Conference on Text, Speech and Dialogue. Lecture Notes in Computer Science, Vol. 4629, pp. 196-205, 2007.
- [37] Kangbok Lee, Jongbum Baik, Soowon Lee. “Estimating a Pleasure-Displeasure Index of Word based on Word Similarity in SNS.” Journal of Computing Science and Engineering, Vol. 20, No. 3, pp. 159-164, 2014.
- [38] Jeongin Kim, Taekeun Hong, and Pankoo Kim. “Replacing Out-of-Vocabulary Words with an Appropriate Synonym Based on Word2VnCR.” Mobile Information Systems, 2021.
- [39] So-Yeon Kim, andHeonchang Yu. “Korean Multiple Sensibility Analysis Technique of SNS Unstructured Data.” 2018 Summer Conference of KACE, Vol. 22, No. 2, pp. 147-149, 2018.
- [40] Mangi Kang. “Sentiment analysis in text mining using ensemble-HMM.” Master’s thesis, Hanyang University, 2016.
- [41] Takayuki Hasegawa, Nobuhiro Kaji Naoki Yoshinaga, and Masashi Toyoda. “Predicting and eliciting addressee’s emotion in online dialogue.” Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics, Vol. 1, pp. 964-972, 2013.
- [42] Sang-Woo Kang, Hong-Min Park, and Jung-Yun Seo. “Emotion classification of user’s utterance for a dialogue system.” Korean Journal of Cognitive Science, Vol. 21, No. 4, pp. 459-480, 2010.
- [43] Dong-Won Shin, Yeon-Soo Lee, Jung-Sun Jang, and Hae-Chang Lim. “Emotion Classification in Dialogues Using Embedding Features.” Annual

- Conference on Human and Language Technology, pp. 109–114, 2015.
- [44] Robert Plutchik. “A general psychoevolutionary theory of emotion.” In *Emotion: Theory, Research, and Experience: Vol. 1, Theories of Emotion*, pp. 3–33, 1980.
- [45] JiYoung Lee and Franck Dernoncourt. “Sequential Short-Text Classification with Recurrent and Convolutional Neural Networks.” *Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 2016.
- [46] JongYoon Won, and KunChang Lee. “Multi-Label Classification Approach to Effective Aspect-Mining.” *Information Systems Review*, Vol. 22, No. 3, pp. 81–97, 2020.
- [47] Ye, Lu, Rui-Feng Xu, and Jun Xu. “Emotion prediction of news articles from reader’s perspective based on multi-label classification.” *2012 international conference on machine learning and cybernetics, IEEE*, Vol. 5, pp. 2019–2024, 2012.
- [48] Jabreel, Mohammed, and Antonio Moreno. “A deep learning-based approach for multi-label emotion classification in tweets.” *Applied Sciences* 9, No. 6: 1123, 2019.
- [49] Yeongbeom Lim, San Kim, Jin Yea Jang, Saim Shin, and Minyoung Jung. “KE-T5-Based Text Emotion Classification in Korean Conversations.” *The 33rd annual conference on human & cognitive language technology*, 2021.
- [50] Jesse Read, Bernhard Pfahringer, Geoff Holmes, and Eibe Frank. “Classifier chains for multi-label classification.” *Joint European conference on machine learning and knowledge discovery in databases*, Springer, Berlin, Heidelberg, pp. 254–269, 2009.
- [51] Jesse Read, Bernhard Pfahringer, Geoff Holmes, and Eibe Frank. “Classifier chains for multi-label classification.” *Machine Learning*, Vol. 85, No. 3, pp. 333–359, 2011.
- [52] Johannes Fürnkranz, Eyke Hüllermeier, Eneldo Loza Mencía, and Klaus Brinker. “Multilabel classification via calibrated label ranking.” *Machine learning*, Vol. 73, No. 2, pp. 133–153, 2008.

- [53] Eyke Hüllermeier, Johannes Fürnkranz, Weiwei Cheng, and Klaus Brinker. “Label ranking by learning pairwise preferences.” *Artificial Intelligence*, Vol. 172, No. 16, pp. 1897–1916, 2008.
- [54] Jun Huang, Guorong Li, Shuhui Wang, Zhe Xue, and Qingming Huang. “Multi-label classification by exploiting local positive and negative pairwise label correlation.” *Neurocomputing*, Vol. 257, pp. 164–174, 2017.
- [55] “open dictionary pro.” Naver open dictionary PRO, <https://open-pro.dict.naver.com/> (accessed. Sep., 2, 2021)
- [56] “List of New Internet Terms in Korea.” Wikipedia, https://ko.wikipedia.org/wiki/대한민국의_인터넷_신조어_목록 (accessed. Sep., 23, 2021)
- [57] “abbreviation.” namu wiki, <https://namu.wiki/줄임말?from=준말> (accessed. Nov., 20, 2021)
- [58] “standard Korean dictionary.” National Institute of Korean Language, <https://stdict.korean.go.kr/main/main.do> (accessed. Oct., 5, 2021)
- [59] Jonghwan Oh, Sooyeon Jang, Joonhwan Lee. “An Exploratory Study on Usage of Korean Consonants and Vowels for Delivering Emotion.” *Journal of Korea Multimedia Society*, Vol. 17, No. 7, pp. 866–878, 2014.
- [60] MyungJin Lim, Moungho Yi, Pankoo Kim, and JuHyun Shin. “Multi-label Emotion Recognition Technique considering the Characteristics of Unstructured Conversation Data.” *Mobile Information Systems*, 2022.
- [61] “한국어 형태소 분석기 체험 및 비교”, github, https://soohee410.github.io/compare_tagger (accessed. Sep., 23, 2021)
- [62] Myungjin Lim, Moungho Yi, Juhyun Shin. “Performance Improvement Method of Deep Learning-based Emotion Recognition.” *The Journal of KINGComputing*, Vol. 17, No. 5, pp. 88–95, 2021.
- [63] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. “Bert: Pre-training of deep bidirectional transformers for language understanding.” *arXiv preprint arXiv:1810.04805*, 2018.
- [64] Taekeun Hong. “A Method of Video Contents Similarity Measurement based on Text-Image Embedding.” *Doctoral dissertation, Chosun University*, 2022.

- [65] MyungJin Lim, SeonMi Kim, and JuHyun Shin. “Association Prediction Method Using Correlation Analysis between Fine Dust and Medical Subjects.” *Smart Media Journal*, Vol. 7, No. 3, pp. 22–28, 2018.
- [66] MyungHo Yi. “Swear Word Detection Method Using The Word Embedding and LSTM.” Master’s thesis, Chosun University, 2020.
- [67] “Cognitive Technology–Language Intelligence.” AI Hub, https://aihub.or.kr/kefi_data_board/language_intelligence (accessed. Feb., 23, 2021)
- [68] “SemEval-2018”, International Workshop on Semantic Evaluation 2018, <https://alt.qcri.org/semeval2018/> (accessed. Feb., 12, 2021)