

출품번호

1424

제67회 전국과학전람회

단발성 대화 데이터 기반의 다중 감정 분석 모델 구축

2021. 8 . 27 .

출품학생	유선희
지도교사	최진우
구분	학생부
출품부문	산업 및 에너지

목차

1. 서론

가. 탐구 동기 및 목적 -----	4
나. 이론적 배경 -----	4

2. 연구개요

가. 연구 기간 및 일정 -----	6
나. 선행 연구 조사 -----	7
다. 연구 방법 -----	8

3. 연구 과정 및 결과

가. 실험 데이터 수집 및 분류 -----	9
나. 모델 제작 -----	11
다. 모델 성능 평가 -----	16
라. 실험 결과 -----	16

4. 결론

가. 결론 및 제언 -----	18
나. 활용 전망 -----	19
다. 추후 연구 -----	19

5. 참고 문헌

〈표 차례〉

[표 1] 연구 일정	7
[표 2] 감정별 데이터 수	9
[표 3] 데이터 속 예시 문장	10
[표 4] 데이터 분류	11
[표 5] relu층이 두 층일 때, 정확도	16
[표 6] relu층이 세 층일 때, 정확도	16

〈글상자 차례〉

[글상자 1] 토큰화 과정 코드	12
[글상자 2] Countvectorization 과정	12
[글상자 3] x축과 y축 저장정보	12
[글상자 4] 모델 정의 및 컴파일	13
[글상자 5] 모델 학습 코드	13
[글상자 6] 감정을 예측하는 함수 코드	13
[글상자 7] 차트에 표시될 수치 계산 코드	14
[글상자 8] 레이더 차트 표현	15

〈그림 차례〉

[그림 1] Oversampling에 대한 도식	5
[그림 2] KoNLPy에 포함된 ‘아버지가방에들어가신다’ 예문을 품사태깅한 클래스	6
[그림 3] A Recap of Count Vectorization	6
[그림 4] 새로운 방식 인공지능 모델 구조	8
[그림 5] 사용한 데이터세트의 형식	9
[그림 6] 데이터세트 품사 태깅 과정	11
[그림 7] 데이터 차트 디자인	15
[그림 8] 예측 결과 사진(1)	17
[그림 9] 예측 결과 사진(2)	17
[그림 10] 결과 예측 사진(3)	18

요약

본 연구에서는 감정정보가 포함된 한국어 단발성 대화 데이터를 사용하여 다중 감정 분석 모델을 구축하였다. 해당 연구에서는 기존의 구축된 다중 감정 분석 모델과 달리 새로운 방식으로 연구를 진행하였다. 사용자가 입력한 한 문장에는 한 가지 감정이 아닌 그 이상의 감정이 담겨 있다는 점을 착안하여 다중 감정 분석 문제를 다중 분류 문제가 아닌 이진 분류 문제로 해결하였다. 그 결과, 국내에서 정확도 70%를 밀돌던 다중 감정 분석 분야를 약 82%까지 올리는 모델을 제작하였다.

키워드 : 기계학습, 자연어 처리, 다중감정분석, 감정 분류 모델

1. 서론

가. 탐구 동기 및 목적

1) 탐구 동기

최근 코로나19로 인해 사람간의 대면 대화가 줄어들고, 자연스레 인터넷을 통한 비대면 대화가 늘게 되었다. 비대면 방식의 대화는 화상통화로도 가능하지만 그 과정이 복잡하기 때문에 대부분의 사람들은 텍스트를 통해 다른 사람들과 소통한다. 이에 따라, 본 연구에서는 직접 얼굴을 보지 않고 작성된 글을 통해서도 작성자의 감정을 파악할 수 있는 인공지능을 구축하기로 하였다.

감정 분석이라는 포괄적인 탐구 주제를 잡고, 선행 연구를 분석하던 중, 인공지능 감정 분석 연구는 이미 많이 진행되어 왔음을 확인할 수 있었다. 하지만, 국내의 다중 감정 분석 연구 분야는 국외에 비해 부족하다는 점을 알게 되었다. 다중 감정 분석 연구는 국외에서 활발하게 진행되어 다양한 모델이 공유되어 왔지만, 국내 연구에서는 다중 감정 분석보다는 긍정, 부정 두 가지 감정으로 분류하는 연구가 주로 진행되어왔다. 그 결과, 국내에서 진행된 다중 감정 분석 연구 감정 점수는 분류 정확도 면에서 기존의 긍정, 부정 연구 모델보다 정확도가 낮은 편임을 확인했다. 이에 본 연구는 기존 다중 감정 분석 모델과는 다른 방식으로 문제를 접근하여 다중 감정 분석 모델의 정확도를 높여보려고 한다.

2) 탐구 목적

본 탐구를 통해 알아보고자 하는 것은 다음과 같다.

1. 단발성 대화 데이터를 학습시킨 다중 감정 분석 모델을 구축한다.
2. 기존의 다중 감정 분석 모델과 비교했을 때, 더 높은 정확도를 가진 모델을 구축한다.
3. 다중 감정 분석 문제를 이진 분류 문제로 접근하여 다중 감정 분석 모델을 구축하고, 해당 방식이 높은 정확도를 보이는지 확인한다.

나. 이론적 배경

1) 자연어(Natural Language)

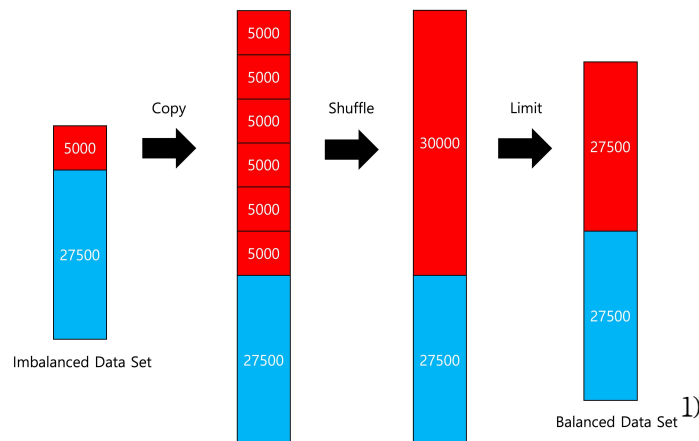
컴퓨터에서 사용하는 기계어와 구분되어 일상생활에서 의사소통을 위하여 사용하는 언어를 가리키는 단어이다. 주로 인공지능 분야에서 연구하는 목표 중 하나가 컴퓨터 환경에서 자연어를 이해, 모방하는 것이다.

2) 자연어 처리(Natural Language Processing, NLP)

자연어 분석, 이해, 생성 등의 기술을 통틀어 이르는 말로, 인공지능의 주요 분야이다. 이해는 컴퓨터가 입력된 자연어에 따라 동작하는 기술이고, 생성은 컴퓨터 언어를 사람이 이해할 수 있는 자연어로 변화하는 기술이다. 분석은 그 정도에 따라 형태소 분석, 통사 분석, 의미 분석, 화용 분석의 4가지로 나눌 수 있다. 1950년대부터 기계 번역 등의 자연어 처리 기술이 연구되었고, 1990년대에 들어 대량의 데이터를 활용하는 기계학습 기반 및 통계적 자연어 처리 기법이 사용되었다. 최근에는 딥러닝 기술이 적용되어 더욱 심도 있고 인간의 판단 능력과 가까워진 자연어 처리 기술이 선보여지고 있다. 이렇게 처리된 자연어는 정보 검색, 기계 번역, 질의응답 등 다양한 분야에 응용되고 있으며, 특히 인공지능 산업에서는 필수 요소이기 때문에 현재도 많은 이들이 연구하고 있다.

3) 오버샘플링(Oversampling)

데이터가 불균형 분포를 가지는 경우, 모델 학습 시 편향이 발생할 가능성이 있다. 이러한 문제를 해결하기 위해 나온 개념으로, 오버 샘플링은 낮은 비율 클래스의 데이터 수를 늘림으로써 데이터 불균형을 해소하는 아이디어이다.



[그림 1] Oversampling에 대한 도식

3) KoNLPy

KoNLPy는 한국어 정보처리를 위한 파이썬 패키지이다. 자연어 처리 중 형태를 분리하는 데이터 전처리 과정에서 주로 사용된다. 패키지 안에는 Khma, Twitter(Okt), Mecab 등과 같은 클래스가 들어 있으며, 이들 중 시간적인 성능 면을 포함하여 Mecab의 효율이 가장 좋다. [그림 2]는 ‘아버지가방에들어가신다’라는 예문이 어떤 식으로 분류되는지 보여주는 그림이다.

1) <https://d2.naver.com/helloworld/7753273>

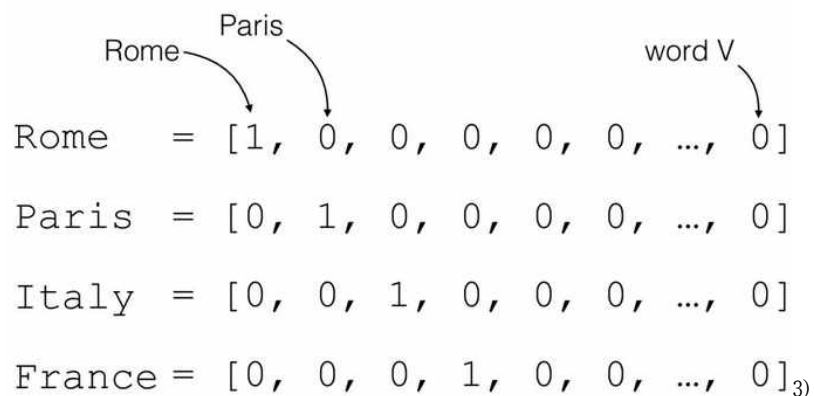
Hannanum	Kkma	Komoran	Mecab	Twitter
아버지가방에 들어가 / N	아버지 / NNG	아버지가방에 들어가신다 / NNP	아버지 / NNG	아버지 / Noun
이 / J	가방 / NNG		가 / JKS	가방 / Noun
시ㄴ다 / E	에 / JKM		방 / NNG	에 / Josa
	들어가 / VV		에 / JKB	들어가신 / Verb
	시 / EPH		들어가 / VV	다 / Eomi
	ㄴ다 / EFN		신다 / EP+EC	

2)

[그림 2] KoNLPy에 포함된 ‘아버지가방에들어가신다’ 예문을 품사태깅한 클래스

4) 카운트 기반 벡터화(Count Vectorization)

Count Vectorization은 텍스트마다 등장하는 단어의 빈도수를 파악해 하나의 카운트 벡터로 만든다. 카운트 벡터 생성 후 해당 텍스트와 다른 텍스트 사이의 벡터 거리를 계산하여 텍스트 사이의 유사도를 파악한다.



[그림 3] A Recap of Count Vectorization

2. 연구개요

가. 연구 기간 및 일정

1) 연구 일정

2) <https://konlpy.org/ko/latest/>

3) <https://towardsdatascience.com/natural-language-processing-count-vectorization-with-scikit-learn-e7804269bb5e>

기 간	활동
1월 15일 ~ 2월 20일	선행 연구 분석 및 탐구 주제 선정
2월 21일 ~ 2월 26일	탐구 계획 설정
2월 27일 ~ 3월 9일	데이터 수집 및 분류
3월 10일 ~ 4월 10일	인공지능 모델 제작 1차
4월 11일 ~ 4월 13일	모델 성능 평가 및 예측 결과 확인
4월 14일 ~ 5월 15일	활용방안 탐구 및 보고서 작성 1차
7월 15일 ~ 7월 31일	인공지능 모델 제작 2차
8월 1일 ~ 8월 16일	최종 보고서 작성

[표 1] 연구 일정

나. 선행 연구 조사

선행 연구 조사는 dbpia⁴⁾, 구글 스칼라⁵⁾를 기반으로 ‘다중 감정 분석, 자연어 처리, 인공지능, 감정 분석 모델’ 키워드 중심으로 조사하였다.

1) 사용자 감정 포스팅 기반 다중 감정 모델 개발[1]

해당 연구에서는 Parrott 감정 모델을 기반으로 다중 감정 분류 모델을 구축하였다. 해당 모델의 분류 감정은 분노, 두려움, 즐거움, 사랑, 슬픔 총 5가지 감정으로 이루어져있다. 또한, 논문을 쓴 저자가 직접 명시한 50만 개의 감정 정보를 활용하였다. 이 연구에서 차별화되는 부분은 단어들의 의미론적 관계와 글의 패턴을 기반으로 감정을 유추했고, 이모지 데이터를 활용하여 이모지 유무에 대한 모델의 성능을 비교하였다. 그 결과 LSTM 모델이 57.4%로 가장 높으며, emoji 데이터가 포함될 때 모델의 성능이 58.3%로 향상되었다는 결과를 도출하였다. 반어법, 은유법과 같은 복잡한 감정 표현의 가능성을 제시하였다.

2) 인터넷 뉴스 댓글 기반의 다중 감정 분석 모델 개발 및 적용[2]

해당 연구에서는 4가지 감정(기쁨, 슬픔, 분노, 혐오)으로 분류한 데이터를 바탕으로 기계학습 기반 다중 감정 분석 모델을 구축하였다. 기사와 댓글을 연관하여 총괄적인 측면에서 문맥을 파악하여 판단한 감정 데이터를 기반으로 한 모델이다. 해당 모델은 79%의 정확도를 달성하였다.

3) BERT 언어 모델을 이용한 감정 분석 시스템[3]

문서 내 텍스트가 나타내는 단어와 문맥을 기반으로 감정수치를 계산하여 긍정 또는 부정을 결정하였다. 이 연구는 영화평 데이터 32만개를 구축하여 감정 분석 연구를 진행하였으며 최근

4) <https://www.dbpia.co.kr/>

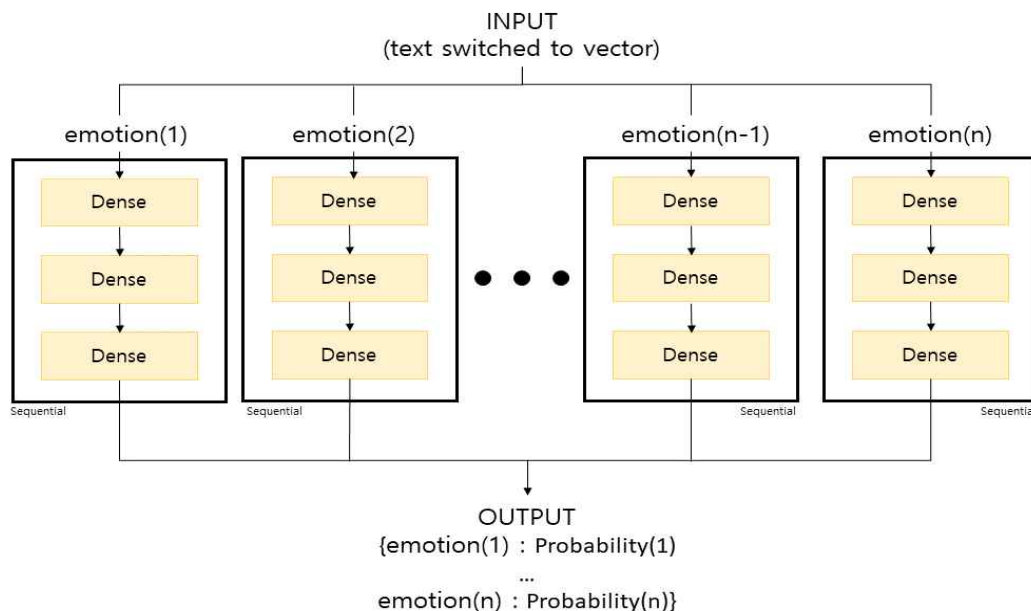
5) <https://scholar.google.co.kr/>

자연어처리 분야에서 높은 성능을 보여주는 KoBERT 모델을 이용하였다. 그 결과 89.90%로 높은 성능을 보여주었다. 해당 연구는 다중 감정 분석 관련 연구는 아니지만, 국내의 긍/부정 분류 인공지능의 성능을 보여주는 연구였다, 다른 연구와 비교했을 때, 국내 다중감정분석 연구가 부족하다는 것을 알 수 있다.

다. 연구 방법

연구에 필요한 데이터를 수집하고, 해당 데이터를 이용하여 파이썬 라이브러리와 패키지 등을 조사하고 탐구한 뒤, 가장 적합한 형식으로 다중 감정 분석 모델을 제작한다. 그 후, 은닉층의 수에 변화를 주면서 테스트 데이터를 통해 인공지능의 정확도를 확인했다. 또한, 그 결과를 레이더 차트를 이용하여 표현한다.

본 연구와 기존 연구의 차별화 점은 다음과 같다. 본 연구는 기존의 다중 감정 분석 모델과는 조금 다른 방식의 방법을 진행한다. 기존의 다중 감정 분석 분야에서는 여러 감정들 중 하나로 출력되는 형태이다. 하지만, 이런 형태는 감정을 정확히 분석했다고 표현하기는 어렵다고 생각한다. 그 이유는 한 문장에 대해 한 감정만이 존재하지 않고, 한 문장이 여러 감정을 동시에 담고 있을 수 있기 때문이다. 따라서 이러한 문장의 특성을 착안하여 본 연구에서는 여러 감정 각각에 대해 예측하여 결과 값을 도출하고, 해당 결과 값을 통합하여 어떤 문장을 입력했을 때, 해당 문장에 어떤 감정들이 담겨 있는지 출력되는 모델을 제작한다. 이러한 방식으로 모델을 제작할 경우, 다중 감정 분석 모델을 다중 분류 문제가 아닌 이진 분류 문제로 해결해 나가기 때문에 기존의 모델보다 더 높은 정확도를 가진다.



[그림 4] 새로운 방식 인공지능 모델 구조

[그림 4]는 본 연구에서 구축한 모델을 표현한 모식도이다. [그림 4]에서 볼 수 있듯이, 여러 감정을 각각 따로 학습시키기 때문에 기존의 다중 감정 분석 모델과 비교했을 때, 높은 정확도를 보이고 여러 감정을 통합하여 출력하기 때문에 한 문장에 대해서 여러 감정에 대한 정보를 확인할 수 있다.

3. 연구 과정 및 결과

가. 실험 데이터 수집 및 분류

1) 데이터 수집

연구를 수행하기 위해 여러 감정으로 분류되어있는 데이터가 필요하다. 필요한 데이터는 'AI hub'⁶⁾에서 얻을 수 있었다. 제공된 다양한 데이터 중 이 연구에서는 SNS 글 및 온라인 댓글에 대한 웹 크롤링으로 작성된 데이터세트인 '한국어 감정 정보가 포함된 단발성 대화 데이터셋'⁷⁾의 사용 승인을 받아 연구 데이터로 이용하였다. 승인 받은 데이터는 SNS 글 및 온라인 댓글에 대한 웹크롤링을 실시하여 선정된 문장이며, 문장 단위 작업을 수행할 수 있도록 문장 분리 작업을 거쳤다. 해당 데이터는 모두 38594개의 문장으로, '공포', '놀람', '분노', '슬픔', '중립', '행복', '혐오', 7가지 감정으로 분류되어 있으며, 글자 수 분포는 23.7 ± 13.6 자이다.

	A	B
1	Sentence	Emotion
2	언니 동생으로 부르는게 맞는 일인가요..??	공포
3	그냥 내 느낌일뿐겠지?	공포
4	아직너무초기라서 그런거죠?	공포
5	유치원버스 사고 났다던데	공포
6	근데 원래이런거맞나요	공포
7	남자친구가 떠날까봐요	공포
8	이거 했는데 허리가 아플수도 있나요? ;;	공포
9	내가불안해서꾸는걸까..	공포
10	일주일도 안 남았당...ㅠㅠ	공포
11	약은 최대한 안먹으려고 하는데좋은 음식있나요?0	공포
12	구직활동 하면서 남는시간은 뭘로 활용해야 되지..	공포
13	괜찮은분 같아서 괜히 조금해지네요 ㅊ	공포

[그림 5] 사용한 데이터세트의 형식

각 감정에 대한 데이터 수는 다음과 같다.

감정	공포	놀람	분노	슬픔	중립	행복	혐오
데이터 수(개)	5468	5898	5665	5267	4830	6037	5429

[표 2] 감정별 데이터 수

[표 2]를 통해 감정별 데이터의 분포를 확인했을 때, 놀람과 행복의 데이터 수가 다른 감정에 비해 많고, 중립 감정이 다른 감정과 비교했을 때, 상대적으로 적다는 것을 알 수 있다. 기존의 감정 분석 모델의 방식에서는 다중 분류 문제로 해결하기 때문에 7가지 감정의 데이터 수를 모두 비슷하게 설정하였지만, 본 연구에서는 다중 감정 문제를 이진 분류로 해결하고 각각의 결과를 합치는 방식이기 때문에 [표 2]의 감정에 대한 데이터 수를 모두 같게 맞추지는 않았다. 따라서 감정별로 예측 정확도의 차이가 발생한다.

해당 데이터의 감정 분류는 사람이 진행하였고, 분류 결과, 감정 데이터에 포함되어 있는 문

6) <https://aihub.or.kr/>

7) <https://aihub.or.kr/opendata/keti-data/recognition-language/KETI-02-009>

장들의 예시는 다음 [표 3]와 같다.

감정	예시 문장
공포	걱정되서 잠이안와요!! / 삼성의 앞날이 걱정이다. 또한번 imf사태가 올것같아 / 조그만 개도 무서운데 엄청 큰 개를
놀람	대박.. 진짜 탈퇴할 줄 이야.. / 아죄송합니다 ^^ 진짜 깜짝놀랐어 / 인도가이런기술을 가 지고있었다니 놀랍군요 / 그것도 참 신기해
분노	꼭 벌받을겁니다 당신 / 호들갑 떨지 마라! 제발 저런짓 좀 하지맙시다 / 진짜 너무분합니다
슬픔	어깨라도넓었으면 옷빨이라도 잘살텐데.. / 너무 슬프고 세상을 다 잃은 기분이고... 세상에 마음 돌곳이 없다. / 누군가를 좋아한다는게 이렇게 힘든건지
중립	저 문 어떻게 여는 거예요? / 이것이 공정한 사회요 법치이다. 생각해봅시다. / 서울인구가 1000만이라고 한다.
행복	생일진심으로 축하해요 / 응원할게요^^ /요즘은 스포츠 뉴스를 보면 넘 기분 좋은거 있죠? /그동안 기복없이 계속못하는거 보다 기복있어도 이렇게 해주는게 월났다
혐오	머리에 뭐가 들었나? 그것이 궁금하다 / 올해 연말은 최악이네요 일에 관련된 사람들 카톡 받기싫어 / 진짜 막장도 이런 막장이 없다

[표 3] 데이터 속 예시 문장

데이터 속 문장을 살펴보면, 정치, 경제, 스포츠 등 다양한 사회 분야 기사에 달린 댓글임을 확인할 수 있다. [표 3]에서 볼 수 있듯이, 데이터 속 대부분의 문장에서 맞춤법과 띄어쓰기가 정확하게 지켜지지 않았다. 따라서 데이터 처리 과정에서 해당 단어들을 고칠 필요성이 있다.

공포, 놀람, 분노, 슬픔, 행복, 혐오 감정은 예시 문장을 보고 충분히 이해할 수 있다. 하지만, 중립 감정의 경우에는 데이터 내용을 통해 일반적인 대화를 중립이라 생각하고 예측을 진행하였다.

2) 데이터 분류

본 연구에서는 전체 데이터를 7:3 비율로 모델 학습용 데이터와 모델 테스트용 데이터로 분류하였다. 7:3 비율의 데이터를 처리해 학습용 데이터 27016 문장, 테스트용 데이터 11578 문장으로 분류하였다. 이후에는 데이터 불균형 현상을 막기 위해 오버샘플링을 진행하였다.

예를 들어, [표 2]를 보면, 공포 데이터의 수는 5468개이다. 이를 7:3 비율로 나누면 훈련용 데이터는 모두 3827개를 얻을 수 있다. 이 3827개의 데이터와 비교할 데이터들은 전체 27016개의 문장에서 3827개를 제외한 23189개의 데이터이다. 하지만, 23189개의 데이터는 3827개와 비교했을 때 약 6배 정도 많은 양이다. 만약, 6배 정도 차이 나는 데이터를 이용하여 인공지능 모델을 학습시키면, 데이터 불균형 현상이 발생하여 편향이 발생할 가능성이 있다. 따라서 3827개의 데이터 수를 27016개의 데이터로 맞추기 위하여 3827개의 데이터를 6배 정도로 복제

하였다.

앞서 예를 들어 설명한 공포 감정의 훈련용 데이터뿐만 아니라 나머지 데이터들에 대해서도 각각 6배를 복제하여 오버샘플링을 진행하였다. 그 결과는 [표 4]를 통해 정리되어 있다. [표 4]의 데이터 비중을 나타낸 퍼센트는 소수 둘째자리에서 반올림하였다.

emotion	train dataset		test dataset	
	분석할 감정	비교할 감정	분석할 감정	비교할 감정
공포	3827*6 (49.8%)	23189 (50.2%)	1642*6 (49.8%)	9936 (50.2%)
놀람	4128*6 (52.0%)	22888 (48.0%)	1769*6 (52.3%)	9809 (47.7%)
분노	3965*6 (50.8%)	23051 (49.2%)	1699*6 (50.8%)	9879 (49.2%)
슬픔	3686*6 (48.7%)	23330 (51.3%)	1580*6 (48.7%)	9998 (51.3%)
중립	3381*6 (46.2%)	23635 (53.8%)	1449*6 (46.2%)	10129 (53.8%)
행복	4225*6 (52.7%)	22791 (47.3%)	1811*6 (52.7%)	9767 (47.3%)
혐오	3800*6 (49.5%)	23216 (50.5%)	1628*6 (49.5%)	9950 (50.5%)

[표 4] 데이터 분류

나. 모델 제작

1) 실험 환경

다중 감정 분석 모델 구축을 위한 개발 언어로는 파이썬(Python)을 사용했으며, 인공지능 학습에 최적화된 케라스(keras)⁸⁾ 패키지를 사용하여 모델을 구축하였다.

2) 품사태깅 및 토큰화 과정

정제한 데이터를 python의 pandas를 통해 불러온다. 불러온 데이터를 한국어 처리 패키지 KoNLPy를 통해 띄어쓰기 알고리즘과 정규화를 이용해 맞춤법이 틀린 문장을 고치고, 형태소 분석과 품사를 태깅한다. KoNLPy에는 Kkma, Komoran, Mecab, Okt등과 같은 클래스가 들어 있다. 본 연구에서는 데이터세트 안에 ‘아버지가방에들어가신다’와 같은 띄어쓰기가 완전히 지켜지지 않은 형태의 문장은 없기 때문에 가장 효율이 좋은 Mecab보다 Okt를 사용하여 데이터의 품사를 태깅하였다.

```
from konlpy.tag import Okt
okt_emotion = Okt()
okt_emotion.pos(u'오늘 하루도 수고했어')
```

```
[('오늘', 'Noun'), ('하루', 'Noun'), ('도', 'Josa'), ('수고', 'Noun'), ('했어', 'Verb')]
```

[그림 6] 데이터세트 품사 태깅 과정

pos의 파라미터는 norm과 stem이 있는데, norm은 문장을 정규화하고, stem은 문장을 근어로 표현한다. 각각의 예를 들어보면, norm의 경우, ‘그래용ㅎㅎ’를 ‘그래요’로 변환하고,

8) Keras library: <https://keras.io/>

stem은 ‘그래요’를 ‘그렇다’로 변환한다. 본 연구에서는 Countvectorization을 원활하게 진행하기 위해 정규화 과정을 진행하였다. 근어로 표현하는 과정은 진행할 경우, 문장이 기존 의도와 다른 의도로 변형되어 본 연구에 맞지 않아 진행하지 않았다. 따라서 토큰화하는 과정에서 정의한 함수는 다음과 같다.

```
def emotion_tokenize(doc):  
    return['/'.join(t) for t in okt_emotion.pos(doc, norm=True, stem=False)]
```

[글상자 1] 토큰화 과정 코드

Countvectorization 과정을 구현하기 위해 nltk 자연어 처리 패키지를 이용하여 한 감정에서 자주 출현하는 단어를 탐색할 수 있는 기능을 이용하였다. Countvectorization 과정 코드는 다음과 같다.

```
import nltk  
emotion_text = nltk.Text(tokens, name='EMOTION')  
emotion_frequency_count = 3000; #1000, 2000, 3000  
emotion_selected_words = [f[0] for f in emotion_text.vocab().most_common(emotion_frequency_count)]
```

[글상자 2] Countvetorization 과정

Countvectorization이므로 frequency_count가 존재한다. [글상자 2]에서 frequency_count는 1000, 2000, 3000으로 바뀌가면서 연구를 진행하였다. 숫자를 늘릴수록 정확도는 높아지지만, 3000 이상으로 실험할 경우에는 메모리 에러가 발생하여 최종적으로 3000으로 설정하였다.

```
def emotion_term_frequency(doc):  
    return[doc.count(word) for word in emotion_selected_words]  
x_fear_train = [fear_term_frequency(d) for d,_ in train_fear_docs]  
x_fear_test = [fear_term_frequency(d) for d,_ in test_fear_docs]  
y_fear_train = [c for _,c in train_fear_docs]  
y_fear_test = [c for _,c in test_fear_docs]
```

[글상자 3] x축과 y축 저장정보

x축의 데이터에는 문서에 들어가는 단어의 개수(단어들의 빈도수 정보), y축에는 1 또는 0으로 분류한 결과를 저장하였다. x축에서는 함수를 정의하여 단어의 빈도수 정보를 표현하였다.

3) 신경망 구축 과정

#모델 정의

```
emotion_model = tf.keras.models.Sequential([  
    tf.keras.layers.Dense(64,activation= 'relu' ,input_shape=emotion_frequency_count,))  
    tf.keras.layers.Dense(64,activation= 'relu' ),  
    tf.keras.layers.Dense(1,activation= 'sigmoid' )])
```

#컴파일

```
emotion_model.compile(optimizer=tf.keras.optimizers.RMSprop(lr=0.001),  
    loss=tf.keras.losses.binary_crossentropy,  
    metrics=[tf.keras.metrics.binary_accuracy])
```

[글상자 4] 모델 정의 및 컴파일

[글상자 4]는 모델 정의 및 컴파일 코드이다. 본 연구에서 사용하는 인공 신경망은 3층으로 구성하였다. 3층은 활성화함수 relu를 사용하여 64개의 유닛으로 구성된 두 층과 sigmoid 함수를 사용한 출력 층 한 층으로 이루어져있다. 본 연구는 다중 감정 분석이지만, 다중 분류 문제가 아닌 이진 분류 문제로 다중 감정 분석을 접근하기 때문에 sigmoid 함수를 출력 층에 사용하였다.

[글상자 4]에서 정의한 모델을 사용하여 실험을 진행하고, relu층을 늘리면서 추가 실험을 진행하여 정확도가 가장 높은 모델을 정의하였다. 그 결과, [글상자 4]의 relu층이 두 층일 경우가 가장 높았다. 또한, 손실 함수는 분류 문제에서 활용하는 binary_crossentropy를 사용하고, 옵티마이저는 RMSprop를 사용하여 경사하강법을 진행하였다.

4) 모델 학습 과정

본 연구에서 사용한 모델은 epochs는 50, batch size는 512로 설정하였다.

```
emotion_model.fit(x_emotion_train, y_emotion_train, epochs=50, batch_size=512)  
emotion_results = emotion_model.evaluate(x_emotion_test, y_emotion_test)
```

[글상자 5] 모델 학습 코드

5) 7가지 감정 모델 결과 통합

7가지 감정 각각에 대해 2)번에서 4)번 과정까지 모두 진행하여 모든 감정에 대해 인공지능이 학습하도록 하였다. 이후에 어떤 문장을 입력했을 때, 그 문장 속에 7가지 감정이 어느 정도 비율로 들어갔는지를 계산하도록 하였다. 해당 함수의 코드는 다음과 같다.

```
def predict(predict):
    emotion1_token = emotion1_tokenize(predict)
    emotion1_tfq = emotion1_term_frequency(emotion1_token)
    emotion1_data = np.expand_dims(np.asarray(emotion1_tfq).astype('float32'),axis=0)
    emotion1_score = float(emotion1_model.predict(emotion1_data))
    emotion1 = round(emotion1_score*100)
(중략)
    emotion7_token = emotion7_tokenize(predict)
    emotion7_tfq = emotion7_term_frequency(emotion7_token)
    emotion7_data = np.expand_dims(np.asarray(emotion7_tfq).astype('float32'),axis=0)
    emotion7_score = float(emotion7_model.predict(emotion7_data))
    emotion7 = round(emotion7_score*100)

    total = emotion1+emotion2+emotion3+emotion4+emotion5+emotion6+emotion7
```

[글상자 6] 감정을 예측하는 함수 코드

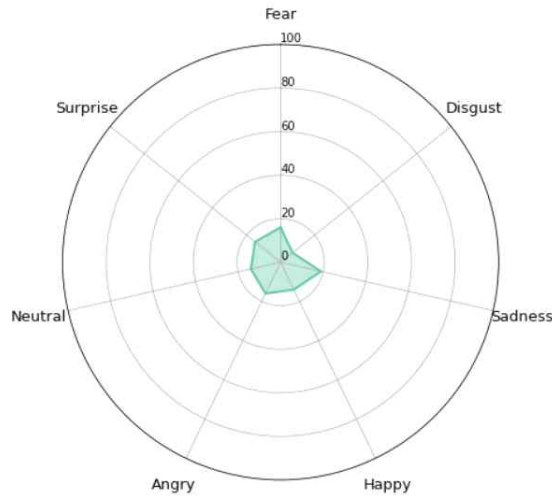
6) 레이더 차트 표현

7가지 감정에 대해 분석한 결과를 시각화하기 위해서 레이더 차트를 이용한다. 레이더 차트로 표현하기 위해서 각각의 수치의 합을 100%로 맞춰 비율을 계산한 뒤, 출력하도록 만들기 위해 다음과 같은 코드를 이용한다.

```
df = pd.DataFrame({
    'Character': ['RESULT'],
    'Fear': [round(100/(total)*fear)],
    'Disgust': [round(100/(total)*disgust)],
    'Sadness': [round(100/(total)*sadness)],
    'Happy': [round(100/(total)*happy)],
    'Angry': [round(100/(total)*anger)],
    'Neutral': [round(100/(total)*neutral)],
    'Surprise': [round(100/(total)*surprise)],})
```

[글상자 7] 차트에 표시될 수치 계산 코드

레이더 차트 디자인은 [그림 7]과 같고, 해당 레이더 차트는 [글상자 8]의 코드로 진행했다.



[그림 7] 데이터 차트 디자인

```

labels = df.columns[1:]
num_labels = len(labels)
angles = [x/float(num_labels)*(2*pi) for x in range(num_labels)]
angles += angles[:1]
my_palette = plt.cm.get_cmap("Set2", len(df.index))
fig = plt.figure(figsize=(15,20))
fig.set_facecolor('white')
for i, row in df.iterrows():
    color = my_palette(i)
    data = df.iloc[i].drop('Character').tolist()
    data += data[:1]
    ax = plt.subplot(3,2,i+1, polar=True)
    ax.set_theta_offset(pi / 2)
    ax.set_theta_direction(-1)
    plt.xticks(angles[:-1], labels, fontsize=13)
    ax.tick_params(axis='x', which='major', pad=15)
    ax.set_rlabel_position(0)
    plt.yticks([0,20,40,60,80,100],['0','20','40','60','80','100'], fontsize=10)
    plt.ylim(0,100)
    ax.plot(angles, data, color=color, linewidth=2, linestyle='solid')
    ax.fill(angles, data, color=color, alpha=0.4)
    plt.title(row.Character, size=20, color=color,x=-0.2, y=1.2, ha='left')
plt.tight_layout(pad=5)
plt.show()

```

[글상자 8] 레이더 차트 표현

다. 모델 성능 평가

모델 성능 평가를 위해 남겨둔 시험 데이터를 통해 정확도를 계산하였다. 나-3)에서 언급했듯이, 모델 평가는 relu층을 3층으로 했을 때와, 4층으로 했을 때를 각각 3번 진행하여 산술평균을 통해 계산하였다. 정확도에 대한 결과는 다음 [표 5]와 [표 6]과 같다. 정확도 계산은 소수 둘째자리에서 반올림했다.

가) relu층이 두 층일 때, 실험 결과

(%)	공포	놀람	분노	슬픔	중립	행복	혐오
1차	83.58	81.09	81.01	84.43	80.54	87.00	76.57
2차	82.91	80.66	82.05	85.07	80.92	86.99	76.96
3차	83.95	80.75	82.24	85.01	80.97	87.27	77.24
평균	83.48	80.83	81.77	84.84	80.81	87.09	76.92

[표 5] relu층이 두 층일 때, 정확도

나) relu층이 세 층일 때, 실험 결과

(%)	공포	놀람	분노	슬픔	중립	행복	혐오
1차	81.11	80.28	81.12	83.35	79.74	86.45	76.26
2차	80.00	80.78	81.55	84.02	78.63	86.84	75.00
3차	81.93	79.60	81.63	84.34	79.61	87.03	76.00
평균	81.01	80.22	81.43	83.90	79.33	86.77	75.75

[표 6] relu층이 세 층일 때, 정확도

[표 5]와 [표 6]에서 각각의 정확도를 비교해보면, 행복의 정확도가 가장 높다는 것을 알 수 있다. 감정의 정확도와 [표 2]의 데이터 수를 비교했을 때, ‘행복’ 감정에 대한 정확도가 높다는 것을 데이터 수와 비교했을 때, 그 이유를 확인할 수 있다. 하지만, 정확도가 가장 낮은 ‘혐오’ 감정은 데이터 수가 가장 적진 않지만, 낮은 정확도를 보였고, 오히려 데이터 수가 상대적으로 적은 ‘슬픔’ 감정에 대한 정확도가 높은 수치를 보였다.

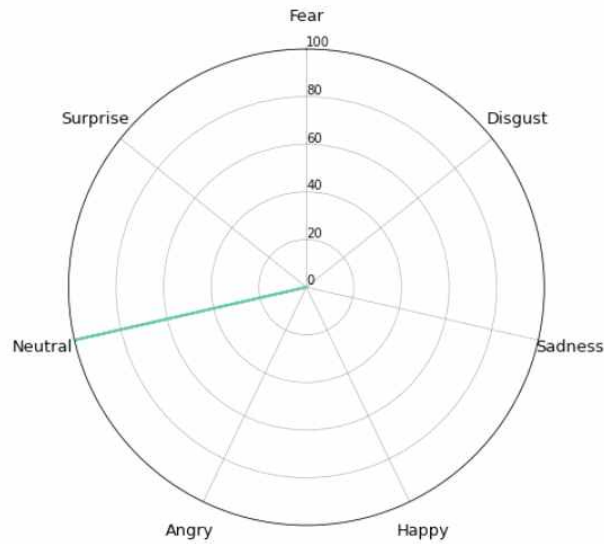
[표 5]를 통해 전체 층을 3개로 했을 때(relu층이 두 층일 때), 7가지 감정에 대한 정확도의 평균은 약 82.25% 정도였고, 전체 층을 4개로 했을 때(relu층이 세 층일 때), 81.20% 정도의 결과를 얻었다. 이 두 값을 비교했을 때, 전체 층을 3개로 했을 때가 전체 층을 4개로 했을 때보다 전체적으로 정확도가 크게 변하지 않고, 오히려 전체 층이 3층일 때, 더 높은 결과를 가지고 있어 본 연구의 다중 감정 분석 모델은 최종적으로 relu층을 두 층으로 하였다.

라. 실험 결과

실험 결과는 [그림 8], [그림 9], [그림 10]와 같은 형태로 출력된다.


```
predict("앞으로 미래산업이 나아가야할 길을 제대로 보여준 소프트웨어같음.")
```

RESULT

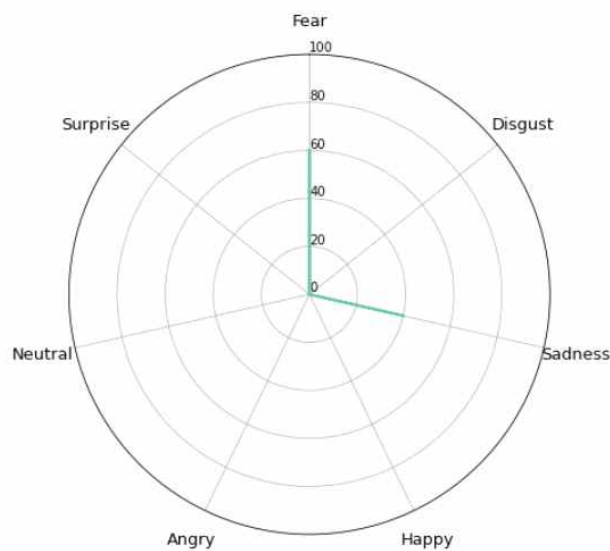


[그림 8] 예측 결과 사진(1)

[그림 8]은 ‘앞으로 미래 산업이 나아가야할 길을 제대로 보여준 소프트웨어 같음.’이라는 문장을 입력하였다. 해당 문장의 예측 결과는 [그림 8] 레이더 차트에서 볼 수 있듯이, 중립 감정 100%가 나왔다. 본 연구는 많은 감정 점수를 동시에 출력하는 방식이지만, 해당 결과처럼 특정 문장에 대해서는 한 감정으로 치우쳐 결과를 출력되는 것을 확인할 수 있다.

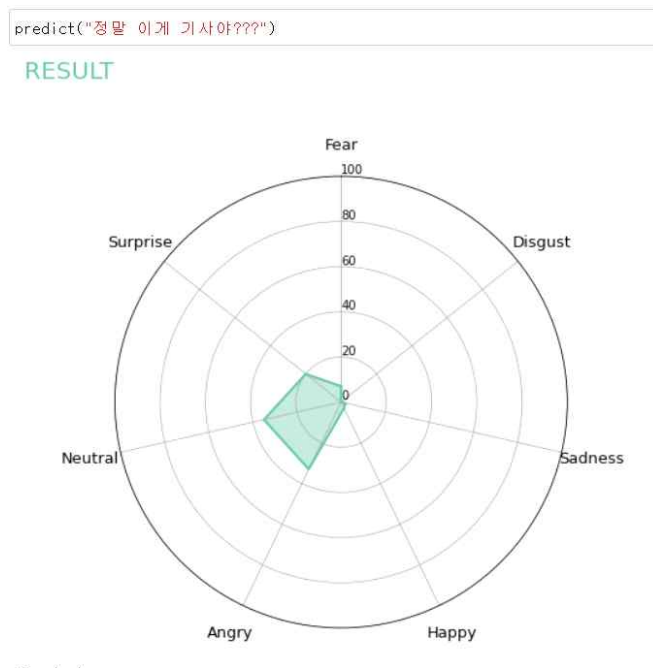
```
predict("곧 헤어질거라는 생각에 연애가 늘 조마조마해요ㅠ")
```

RESULT



[그림 9] 예측 결과 사진(2)

[그림 9]은 ‘곧 헤어질 거라는 생각에 연애가 늘 조마조마해요ㅠ’이라는 문장을 입력하였다. 해당 문장의 예측 결과는 레이더 차트에서 볼 수 있듯이, ‘공포’ 60%, ‘슬픔’ 40%의 예측 결과를 얻었다. [그림 9]의 결과는 한 문장에 대해서 두 가지 감정이 공존됨을 보여주는 결과를 출력하는 것을 확인할 수 있었다.



[그림 10] 결과 예측 사진(3)

[그림 10]은 ‘정말 이게 기사야???’라는 문장을 입력하였다. 해당 문장의 결과는 놀람, 중립, 화남과 약간의 공포 감정이 섞여 있다는 예측 결과를 얻어냈다. 해당 결과는 한 문장에 대해 세 가지 이상의 감정이 공존됨을 보여주는 결과를 출력하는 것을 확인했다.

[그림 8]을 통해 해당 모델은 문장에 따라 기존의 다중 감정 분석 모델처럼 한 가지 감정에 대한 출력도 있지만, [그림 9], [그림 10]을 통해 다양한 감정이 복합적으로 작용한다는 결과를 예측해낸 부분도 존재한다는 것을 확인하였다. 또한, 해당 모델은 사람이 분류한 데이터를 바탕으로 학습을 진행했다. 사람이 분류한 데이터 속 문장들 중에는 사람들이 반어법이라 생각해 표기된 감정들도 있었기 때문에 해당 모델이 반어법도 어느 정도 인지하여 예측하는 결과를 보이기도 했다.

4. 결론

가. 결론 및 제언

웹 크롤링을 통해 수집된 7가지 감정 정보가 포함된 데이터셋을 바탕으로 다중 감정 분석 모델을 제작하였다. 본 연구는 한 문장에 여러 감정이 동시에 존재할 수 있다는 점을 착안하여 기존의 연구와는 다른 방식으로 다중 감정 분석 모델을 제작하였다. 그 결과, 다중 분류 문제를 이진 분류 문제로 해결하여 기존의 70% 정도를 밀돌던 기존 국내 다중 감정 분석 연구에 비해 평균 약 82.25%의 정확도를 얻어낼 수 있었다. 또한, 몇 가지 문장을 직접 입력해보면서

모델을 확인해본 결과, 연구에서 사용한 방식이 신뢰성을 가지고 있다는 것을 확인할 수 있었다.

나. 활용 전망

본 연구에서 정확성을 높이는 방향으로 보완하여 다중 감정 분석 모델을 구축한다면, 사용자의 감정 분석 분야나 여러 텍스트로 이루어지는 의사소통 분야에서 널리 활용될 수 있다. 또한, 챗봇과 같은 AI 분야에서 좋은 모델로 작용할 수 있고, 인터넷 댓글 분석이나 게시물 분석으로 다양한 사람들의 가치관이나 현대 사회를 바라보는 관점에 대해 기존보다 더 융통성 있게 접근할 수도 있다.

기존의 다중 감정 분석 방식과는 다른 방식으로 접근하여 높은 정확도를 보여준 연구였다. 그 결과, 기존에 감정 분석 분야가 활용되는 곳에서 조금 더 사용자 친화적인 결과를 도출하는데 도움을 줄 수 있다.

다. 추후 연구

다음과 같은 방법을 이용하여 본 연구보다 더 높은 정확도를 가지는 모델을 제작할 수 있다. (가) 본 연구에서 사용한 데이터 수는 기존의 선행 연구에서 사용한 데이터 수에 비해 현저히 적기 때문에 더 많은 데이터를 대상으로 연구를 진행하면 높은 정확도를 가지는 모델을 제작할 수 있다.

나) 해당 연구는 카운트 기반의 벡터화를 통해 자연어 처리를 진행했지만, tf-idf와 같은 다른 다양한 처리 기법으로 벡터화를 진행하거나, 본 연구에서 사용한 간단한 신경망이 아닌 기존에 검증 받은 koBERT, LSTM과 같은 모델을 활용하여 연구를 진행하면 더 높은 정확도를 가지는 다중 감정 분석 모델을 구축할 수 있다.

본 연구는 단발성 대화 데이터를 기반으로 진행하였다. 추후 연구에서는 단발성이 아닌 대화가 오가는 연속성 대화 데이터를 기반으로 연구를 진행할 수 있다.

5. 참고문헌

- [1] 유지수, 한경식. “사용자 감정 포스팅 기반 다중 감정 모델 개발.” 한국HCI학회 학술대회 . (2020): 145-150.
- [2] 한재호, 김우정, 한경식. “인터넷 뉴스 댓글 기반의 다중 감정 분석 모델 개발 및 적용.” 한국HCI학회 학술대회 . (2018): 893-897.
- [3] 김택현 (Taek-hyun Kim), 조단비 (Dan-bi Cho), 이현영 (Hyun-young Lee), 원혜진 (Hye-jin Won), 강승식 (Seung-shik Kang). 2020. BERT 언어 모델을 이용한 감정 분석 시스템. 한국정보처리학회 학술대회논문집, 27(2): 975-977