트위터 데이터를 이용한 감정 분류 및 이모티콘 추천

김경남°, 김민주, 유지인, 이주혁, 황유진°
세종대학교 소프트웨어학과, 지능기전공학부 무인이동체공학
kkyy0126@naver.com, min942773@gmail.com, yji9602@naver.com, zero5.two4@gmail.com, yujine92@gmail.com

요 약

본 논문에서는 텍스트가 가진 감정 분류를 통한 이모티콘 추천 기법을 소개한다. 모델 학습을 위한 데이터 셋을 구축하고 Neural Network 를 사용하여 감정 분류 및 이를 통한 이모티콘 추천 분류기를 제안한다. 실험 결과, 문장이 가진 감정을 성공적으로 분류하고 그에 적합한 이모티콘을 추천하는 것을 확인하였다.

1. 서론

감정 분석(sentiment analysis)은 긍부정 형식의 양극(polar) 형태로 분류하는 경우가 대부분이다. 하지만 감정은 분노, 슬픔, 피곤, 기대, 행복 등 긍정과 부정만으로는 구분되지 않는 형태로 이루어져 있다. 따라서 기존의 이분법적인 사고를 넘어, 감정을 더욱 세분화하여 분류하고 해당 감정에 맞는 이모티콘을 추천해주는 분류기를 개발하고자 한다.

2. 시스템 설계

본 절에서는 전반적인 시스템 구조 및 전처리 시스템 구조, 모델 아키텍처에 대해 다룬다.

2.1 전체 시스템 구조

제안하는 이모티콘 추천 시스템의 전체적인 구조는 그림 1 과 같다. 파이썬 프레임워크인 Flask 를 이용하여 웹 인터페이스를 구현하였다. 해당 인터페이스에서 문장을 입력하면, 학습된 모델에서 입력값을 감정분류 후 결과에 따른 이모티콘을 추천하도록 구성하였다.

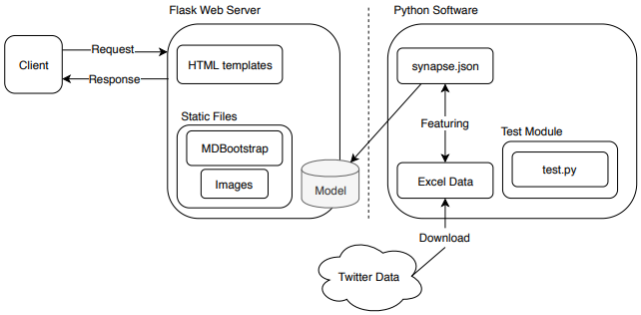

그림 1. 전체적인 시스템 구조

2.2 전처리 시스템 구조

제안하는 데이터 셋의 전처리 시스템의 구조는 그림 2 와 같다. 맞춤법 검사기를 통해 번역기가 인식할 수 있을 정도로 데이터를 변경 후, 번역기를 통해 영어로 가공한다. (그림 확인 후 bow 쓸거면 뒤에 이어서 간단하게)

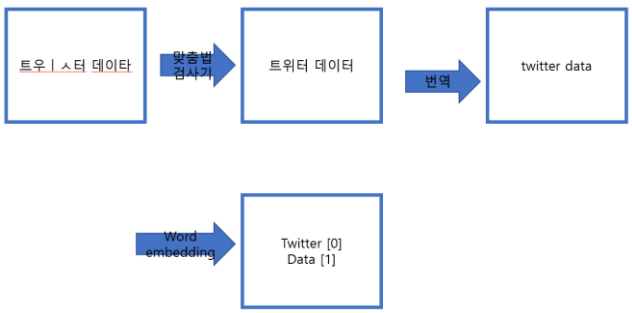

그림 2. 전처리 시스템 구조

2.3 모델 아키텍처

제안하는 모델 아키텍처는 그림 3 과 같다. 순차적 데이터인 자연어에서 순서의 중요성을 학습하는 LSTM 모델과, 단어 자체의 특징을 학습하는 CNN 모델을 함께 사용하였다.

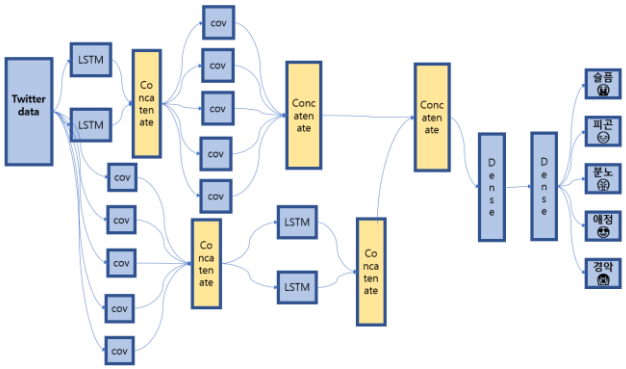

그림 3. 모델 아키텍처

3. 트위터 데이터 수집 및 정규화

본 절에서는 트위터 데이터 셋에 대한 소개와 전처리 과정에 대해 자세히 다룬다.

데이터 셋을 이용한 감정 분류 모델 제작에 대해 자세히 다룬다.

3.1 세부 분류 감정 선별

대부분의 선행연구에서는 분류기의 감정을 긍부정으로 분류하였으나, 주제에 맞춰 감정을 세분화하기 위해 로버트 플루치크(Robert Plutchik)가 제시한 감정의 바퀴(Plutchik's Wheel of Emotions)를 참고하였다. 감정의 바퀴 모델에서, 바퀴의 색깔은 진해질수록 더 강한 감정, 연해질수록 더 순한 감정을 나타낸다. 플루치크의 기본 감정에 대한 심리진화론에서, '기쁨과 슬픔', '분노와 공포'같은 주요 감정들은 양극의 쌍들로부터 개념화될 수 있다고 상정하고 있다. 따라서 여러 감정들 중 주요 감정들의 기본이 되는 감정인 슬픔😭, 지루😒, 분노😡, 애정😍, 경악😱 으로 세부 분류 기준를 최종 선정하였다.

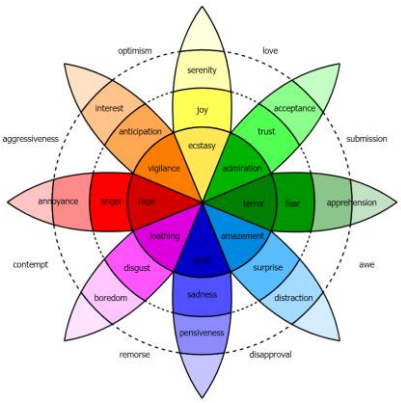

그림 4. 로버트 플루치크의 감정의 바퀴

3.2 학습 데이터 선정

트위터 데이터는 이모티콘 분류기가 가장 많이 활용될 것이라 예측되는 분야인 SNS 중 하나이다. 트위터 데이터는 API 를 활용하여 최신 데이터를 대용량으로 습득하기 용이하다. 또한 은어, 유행어 등 현재 SNS 에서 많이 사용되는 단어나 문장도 쉽게 많은 양을 얻을 수 있다. 이모티콘도 많이 사용된다는 특징이 있어 이모티콘을 라벨로 선정하여 크롤링할 때 유용할 것이라 생각되어 트위터 데이터를 선정하였다.

3.3 학습 데이터 수집

학습 데이터 수집을 위해 파이썬 라이브러리인 twitterscraper 를 이용하였다. 감정별 선정된 이모티콘을 포함하는 한국어 트윗을 각 감정당 15000 개, 총 75000 개를 수집하였다.

	A	B	C	D	E
		id	text		
0		8387813095938	@wooroo 아니그덩요		
1		8195898677723	콜록 "@vectorius: 콜럭! > @Coffeedang 오늘 커피		
2		8168019137359	콧물이 막!		
3		8078506787774	날씨가 더워지고있어요 걸어서 출근하기가 어려워지		
4		7828680563138	@GoNnnn 맞다.... 젠,,,, 액티베이션 불가... 폰은 어		
5		7806049283644	iOS5로 판올림! 그리고... 연락처가 날아감		
6		8387813095938	@wooroo 아니그덩요		
7		8195898677723	콜록 "@vectorius: 콜럭! > @Coffeedang 오늘 커피		
8		8168019137359	콧물이 막!		
9		8078506787774	날씨가 더워지고있어요 걸어서 출근하기가 어려워지		
10		7828680563138	@GoNnnn 맞다.... 젠,,,, 액티베이션 불가... 폰은 어		
11		7806049283644	iOS5로 판올림! 그리고... 연락처가 날아감		
12		1739115315898	@Sunmi5791 안돼안돼~~~우리 아직 할게 넘 많아..		
13		1739083691895	요리조리 생각 골똘히 해봅니다"@Duncan787: 선뜻		
14		1739060351138	굿모닝 환칼 I just finished a 10.0 km run with a time		
15		1739042526101	보이프렌드 Fan Meet yesterday http://www.youtube		
			@hn5830 요즘 바쁘신가봐여 트윗에서 보기 힘드네요		
16		1739019766130			

그림 5. 학습 데이터 수집 코드 및 결과

3.4 부적합 단어, 특수문자 제거

감정분석에 필요하지 않고 홍보의 성향을 띄는 "카카오톡", "상담", "아이돌", "추천"등의 단어와 링크를 제거하고, 특수문자를 제거하였다. 또한 트위터의 리트윗 기능에 의해 발생하는 데이터 중복문제를 해결하기 위해 리트윗 데이터를 제거하였다.

3.5 맞춤법 검사기

기존의 konlpy 역시 token 화과정에서 정규화 기능을 지원하나, 이는 "엌ㅋㅋㅋ"을 "어ㅋㅋㅋ"로 수정하는 정도의 기능을 한다. 그러나 트위터 데이터는 실생활 사용되는 은어나, 유행어가 등장하는 최신 데이터로 정규화 기능이 더 강인해야 하므로 네이버 맞춤법 검사기를 이용해 이를 정규화 하였다.

그림 6. 정규화 결과 비교.
위 konlpy 라이브러리와 이용,
아래 네이버 맞춤법 검사기 이용 비교

3.6 학습 데이터 형태 분석

wordcloud 를 이용하여 데이터에 많이 포함된 단어를 확인하였고, 감정을 표현하지 않는 단어를 제거 하였다. 또한 박스플롯을 이용하여, 수집한 트윗의 길이 편차를 확인하였고, 평균에서 크게 벗어나지 않는다는 결과를 얻어 길이에 대한 정규화는 하지 않았다.

그림 7. 수집한 데이터의 wordcloud

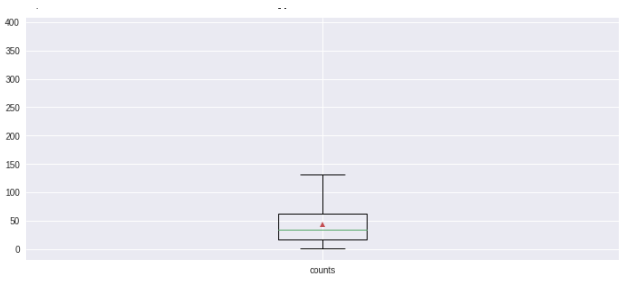

그림 8 . 수집한 데이터 길이를 나타내는 박스플롯

3.7 학습데이터 영어 번역

영어는 New York 같은 합성어나 It's 와 같이 줄임말에 대한 예외를 처리한다면, 띄어쓰기(space) 기준으로 토큰화(tokenization)를 진행해도 단어 토큰화가 잘 작동한다. 대부분의 경우 단어 단위로 띄어쓰기가 이루어지기 때문에, 띄어쓰기 기준 토큰화와 단어 토큰화의 결과값이 거의 비슷하다.

한국어의 경우 띄어쓰기가 되는 단위를 '어절'이라고 하는데, 어절은 '눈사람을∨만들던∨추억' 같이 조사 및 어미의 결합으로 문장에 사용된다. 즉 한국어는 영어와 달리 독립적이지 않은, 교착어이기 때문에 띄어쓰기 기준으로 토큰화를 시도하는 것은 NLP(Natural Language Processing) 에서 지양되고 있다. 따라서 성공적인 토큰화와 모델 적용을 위해 google translater 라이브러리를 활용하여 텍스트를 영어로 번역하였다.

3.8 keras 를 이용한 문장 token 화

오픈 소스 신경망 라이브러리인 keras 를 사용하여 번역한 영어 문장을 token 화 하였다. 또한 token 화 후 sequence 를 padding 하여 모델 학습 시에 input data 로 활용하였다.

3.9 GloVe 를 이용한 워드단어 embedding 의 weight 고정

Glove 란 미국 스탠포드 대학에서 2014 년에 개발한 워드 임베딩 방법론이다. 이는 Word2Vec 의 단어 벡터 사이의 유사도를 측정하는 데엔 뛰어나지만 말뭉치 전체의 문맥 파악은 어렵다는 단점을 보완한 것이다. pre-trained GloVe vectors 중 twitter 200dimension 파일을 이용하여 단어 embedding 의 weight 을 고정하였다.

4. 모델

본 절에서는 데이터 셋을 이용한 감정 분류 모델 제작에 대해 자세히 다룬다.

4.1 LSTM-CNN 과 CNN-LSTM 병합 모델

LSTM 은 자연어 처리 분야에서 많이 쓰이는 model 로, 순차적 데이터의 순서의 중요성을 학습하므로 많은 NLP 작업에서 선호된다. 또한, CNN 은 데이터에서 feature 를 추출하여 의미를 파악한다.

본 프로젝트의 특성 상, 특징을 추출하는 LSTM-CNN 과 시계열 분석을 위한 CNN- LSTM 모델을 병합하였다. 모든 channel 은 연결되어 있으며 max-pooling 을 사용하였다. 마지막 layer 는 Fully

Connected layer 로 softmax activation function 을 통한 예측을 나타낸다.

5. 실험 결과 및 분석

기존 연구는 대부분 영화 감상 댓글이나, 법원 자료와 같은 문법적으로 정돈되어있고, 신조어가 적은 데이터로 이루어졌고, 이는 실생활의 언어를 잘 반영하지 못한다는 한계가 있었다. 이러한 한계를 개선하기 위해 twitter 데이터를 이용하여 실생활의 언어를 반영하려는 연구가 있었다.

우리는 이러한 경향을 반영하여 twitter 데이터를 이용하였고, 이를 네이버 맞춤법 검사기를 이용하여 학습데이터의 bound 가 너무 넓어지지 않으면서, 실생활의 단어를 학습하도록 노력하였다.

또한 기존 연구와 다르게 구글번역기의 번역을 통해 의미가 불분명한 단어들이 번역이 무시되는 경향을 이용하기 위해 번역을 이용하여 데이터의 bound 를 줄였다.

그리고 조사 등 불용어가 문장에 의미를 끼치는 경향이 많은 한국어 문장을 그대로 학습하기보다, 정확도를 높이기 위하여 선행연구가 많이 진행된 영어 모델을 이용하여 감정분류를 하였다.

6. 결론

본 프로젝트는 기존의 단순한 긍, 부정의 이분법적 감정 분류를 넘어서 더 다양한 감정을 분류하는것을 목적으로 한다. 기존의 영화분류처럼 지정된 분야의 데이터가 아닌 실생활언어와 가까운 SNS 의 데이터를 사용함으로서 신조어, 일상생활어에 강인한 모델 을 목적으로 한다. 학습하기 어려운 신조어, 일상어 데이터를 학습데이터로 포함 시키기 위하여 네이버 맞춤법 검사기를 통한 정규화 과정과, google translater 라이브러리를 이용한 번역 과정을 거쳤다. LSTM 과 CNN 을 통해 감정분류 모델을 학습시켰다. 데이터의 특성과, 자료의 부족함으로 정확도가 매우 높게 나오지는 않았지만, 하나의 감정이 두드러지는 문장은 잘 검출된다.

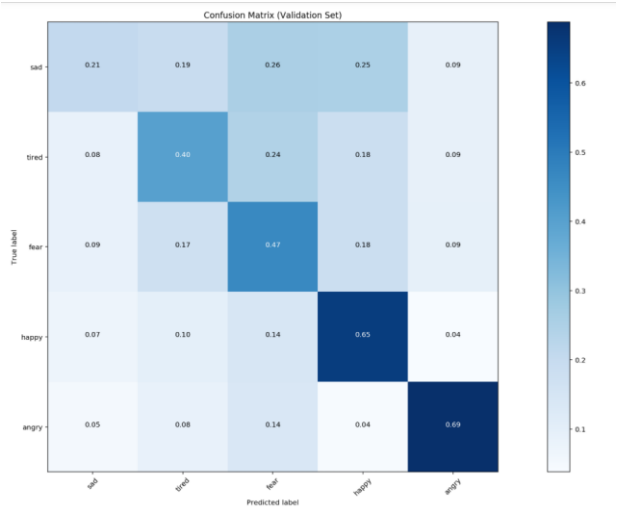

그림.9 모델의 정량적 평가

감사의 글

이 프로젝트는 세종대학교 인공지능 수업의 과제 프로젝트로, 담당 교수님인 최유경 교수님께 주제 선정 및 지원을 받았습니다. 또한 BOAZ 동아리에서 진행한 선행 프로젝트[3]를 참고하였고, 프로젝트 진행자 yunsikus 님 감사드립니다.

참고문헌

[1] yunsikus. (2018). “BOAZ_Project”. https://github.com/yunsikus/BOAZ_Project

[2] “형태소 분석기 성능 비교” https://ratsgo.github.io

[3] “[Keras] KoNLPy 를 이용한 한국어 영화 리뷰 감정 분석” https://cyc1am3n.github.io

[4] “Boaz_9th_Conference 감정기반 이모지추천 시스템” https://github.com/yunsikus/BOAZ_Project

[5] “KoreanSentimentAnalyzer” http://github.com/mrlee23/KoreanSentimentAnalyzer

[6] Convolutional Neural Networks for Sentence Classification (Yoo Kim, 2014)

[7] Twitter Sentiment Analysis using combined LSTM-CNN Models (Sosa, 2017)

[8] \“Neural Network Model” https://iamtrask.github.io//2015/07/12/basic-python-network

[9] A Pattern-Based Approach for Multi-Class Sentiment Analysis in Twitter (Bouazizi and Ohtsuki, 2017)

[10] tlkh. “Multi-class Emotion Classification for Short Texts”https://github.com/tlkh/text-emotion-classification 에서 검색됨