소셜미디어 감성분석을 통한 특정키워드에 대한 대중의 평판 분석

JH95

Chapter1. 분석 개요

1. 분석의 동기 및 목적

트위터의 트윗을 감성분석하여 특정 제품에 대한 평판 및 트렌드를 분석하는 모델을 구축한다.

코카콜라(Coca-Cola)는 트위터, 페이스북과 같은 소셜미디어에 오르내리는 자사의 데이터를 실시간으로 분석하여 부정적인 평가가 급등할 경우 그에 대한 즉각적인 대응을 하는 것으로 알려져 있다. 이번 분석에서는 트위터의 트윗에 한정한 감성분석을 실시하여 특정 시기에 특정 브랜드 또는 제품에 대한 대중의 평판을 긍정과 부정으로 분류한다. 이때 단순 분류가 아닌, 특정 개수의 트윗에서 긍정/부정 비율을 산출한다. 이는 부정 트윗이 일정 비율 이상 높아질 경우 기업이 대응전략을 마련해야하는 상황을 가정하기 위함이다.

2. 활용 데이터

감성분석기 구축을 위해 "네이버 영화 리뷰 데이터"를 활용하였다. 이 데이터를 활용하여 감성분석기를 구축한 뒤 트위터 크롤링을 통해 수집한 데이터를 감성분석한다.

3. 데이터분석 과정 및 방법

데이터분석 및 모델구축의 전체적인 과정은 다음과 같다.

1) EDA : 변수 및 데이터 수 확인, 클래스 간 분포 확인 및 시각화
2) 전처리 :
    a) 결측치 및 중복데이터 확인 및 제거
    b) 토큰화 (Tokenization)
    c) 정수인코딩
    d) 패딩
3) 신경망 구성
4) 학습
5) 크롤링한 데이터 감성분석 및 결과 산출

Chapter2. 단계별 분석 진행

1. EDA

네이버 영화 리뷰 데이터에 대한 EDA를 진행한다.

1) 변수 및 데이터 수 확인

	Train	Test
ID	150,000	50,000
Document	149,995	49,997
Label	150,000	50,000

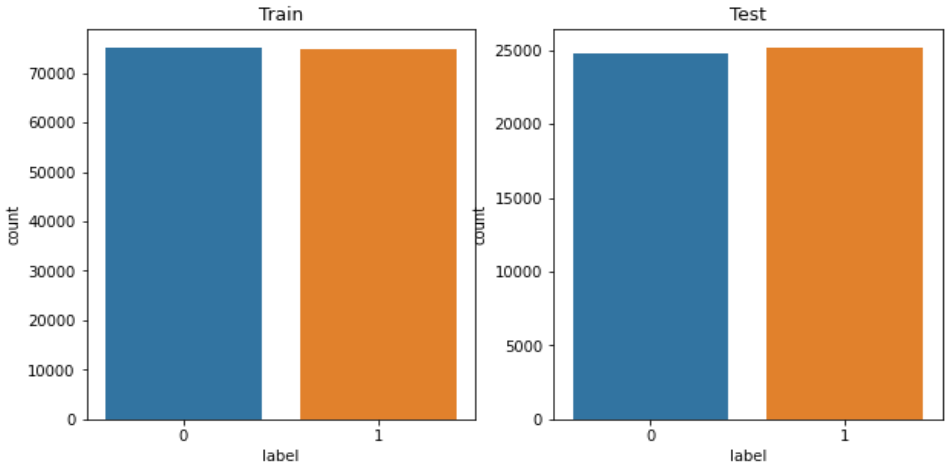
데이터는 위와 같이 Train / Test 셋으로 나뉘어져 있다. 또한 각 데이터셋은 ID, Document, Label 3개의 컬럼으로 이루어져 있다. 또한 Document 컬럼에서 각각 5개와 3개로 결측치가 있는 것을 확인할 수 있다. 따라서 결측치를 제거해주기 위한 작업을 전처리 단계에서 진행한다.

각각의 컬럼이 의미하는 바는 다음과 같다.

ID	리뷰어의 아이디를 나타내는 고유 숫자
Document	실제 리뷰가 담긴 텍스트 데이터
Label	부정(0), 긍정(1) 이루어진 이진 데이터

분석의 목적은 Document를 이용하여 Label을 예측하는 감성분석 모델을 만드는 것이다. 따라서 ID컬럼은 분석에 활용되지 않으므로 제거한다.

2) Label 컬럼 분포 확인



	Train	Test
0	75173	24827
1	74827	25173

Train과 Test 데이터 셋 모두 5:5에 달하는 균일한 분포를 보여주고 있다. 따라서 데이터 수 불균형을 해결하기 위한 작업은 불필요하다.

2. 전처리

1) 결측치 및 중복데이터 제거

앞선 EDA를 통해 발견한 결측치를 제거한다. 중복데이터 또한 이 과정에서 제거한다. 제거작업 후 데이터의 수는 Train/Test 각각 146182개와 49157개이다.

2) 토큰화

본격적인 자연어처리를 위해 Document 컬럼의 한글 리뷰를 토큰화 한다. 토큰화 하기에 앞서, 정규표현식을 이용하여 특수문자와 공백 등을 제거하고 한글 텍스트만을 남겼다.

또한 토큰화를 위해서 KoNLPy 패키지의 OKT를 이용하였다. 특히 한국어의 특징을 살리기 위해 형태소 토큰화(morpheme tokenization)를 진행하였다.

EX1) 아 더빙 진짜 짜증나네요 목소리 -> ['아', '더빙', '진짜', '짜증나다', '목소리']

EX2) 흠포스터보고 초딩영화줄오버연기조차 가볍지 않구나 -> ['흠', '포스터', '보고', '초딩', '영화', '줄', '오버', '연기', '조차', '가볍다', '않다'

3) 정수 인코딩

텍스트 데이터를 신경망이 학습할 수 있도록 숫자형태로 변환해준다. 세부 과정은 다음과 같다.

a) 케라스 Tokenizer의 word_index를 이용하여 각각의 토큰에 인덱스를 부여해준다. 이때 단어의 등장 빈도수가 많을 수록 낮은 인덱스가 부여된다. 총 43753개의 토큰을 가진 집합이 생성됐으며, 이때 43753번째 토큰이 등장 빈도수가 가장 낮다.

b) 최적화를 위해 단어 등장 빈도수가 3회 미만인 토큰은 제외한다. 이때 3회 미만인 토큰은 총43753개 중 24337 개로 이를 제외한 19416개의 토큰으로 신경망을 학습시킨다.

c) 케라스 Tokenizer의 text_to_sequence 이용하여 문장을 각 토큰에 해당하는 인덱스의 배열로 만든다.

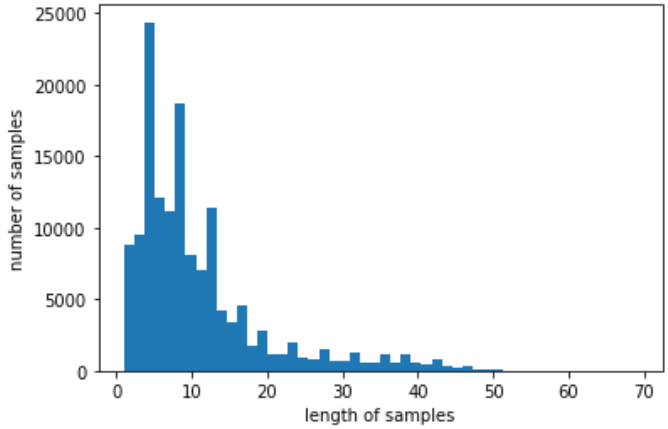
EX) ['아', '더빙', '진짜', '짜증나다', '목소리'] -> [50, 454, 16, 260, 659]

이때 19416번째 인덱스 이후의 토큰으로만 구성된 문장이 있을 수 있다. 이러한 샘플들을 제거해준다. 작업 후 최종적으로는 사용될 데이터의 수는 Train과 Test 각각 145162개와 48747개다.

4) 패딩

케라스의 pad_sequence를 이용하여 샘플들의 길이를 일정하게 맞추기 위해 패딩 작업을

실행한다.



샘플들의 문장 길이의 분포는 위의 그래프와 같다.

리뷰의 최대길이는 69 이고, 리뷰의 평균길이는 10.81이다. 특히 전체 샘플 중 94% 이상이 30이하의 문장 길이를 가지고 있다. 따라서 30의 길이로 패딩 작업을 실행한다.

EX) [50, 454, 16, 260, 659] -> [0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,50,454,16,260,659]

패딩 작업까지 마쳤다면, 데이터가 신경망에 입력될 모든 준비가 되었다.

3. 신경망 구성

```
Model: "sequential"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
embedding (Embedding)        (None, None, 100)         1941600
_________________________________________________________________
lstm (LSTM)                  (None, 128)               117248
_________________________________________________________________
dense (Dense)                (None, 1)                 129
=================================================================
Total params: 2,058,977
Trainable params: 2,058,977
Non-trainable params: 0
_________________________________________________________________
```

신경망은 위와 같이 구성하였다.

Sequential : 신경망의 층(layer)을 구성하기위한 토대를 만들기 위해 Sequential()을 선언해준다.

Embedding : 워드임베딩을 위해 임베딩 레이어를 쌓는다.

첫번째 인자 : 단어 집합의 크기 (여기서는 19416)

두번째 인자 : 출력 차원 (100)

이는 앞선 과정에서 정수인코딩 된 데이터들을 밀집벡터로 변환하기 위해서이다. 밀집벡터는 원-핫벡터에 비해 차원의 수가 작기 때문에 데이터를 처리함에 있어 효율적이다.

LSTM : 자연어와 같은 Sequential 데이터 처리에 유용하게 사용되는 LSTM 레이어를 쌓는다.

출력차원 : 128

*LSTM은 RNN의 "Long-term dependency" 즉 문장의 길이가 길어질수록 정확도가 떨어지는 문제를 해결하기 위해 고안된 방법이다.

Dense : 출력을 위해 Dense 레이어를 추가한다. 이때 activation (활성함수)를 sigmoid로 하여 0 또는 1을 결정할 수 있도록 한다.

4. 학습

신경망 학습을 목적은 손실함수를 최소화하는 파라미터들의 값을 찾는 것이다.

```
Epoch 1/15
1935/1936 [============================>.] - ETA: 0s - loss: 0.3865 - acc: 0.8240
Epoch 00001: val_acc improved from -inf to 0.84845, saving model to best_model.h5
1936/1936 [==============================] - 57s 29ms/step - loss: 0.3865 - acc: 0.8240 - val_loss: 0.3500 - val_acc: 0.8484
Epoch 2/15
1935/1936 [============================>.] - ETA: 0s - loss: 0.2951 - acc: 0.8743
Epoch 00002: val_acc improved from 0.84845 to 0.85437, saving model to best_model.h5
1936/1936 [==============================] - 57s 29ms/step - loss: 0.2950 - acc: 0.8743 - val_loss: 0.3439 - val_acc: 0.8544
Epoch 3/15
1936/1936 [==============================] - ETA: 0s - loss: 0.2445 - acc: 0.8974
Epoch 00003: val_acc did not improve from 0.85437
1936/1936 [==============================] - 56s 29ms/step - loss: 0.2445 - acc: 0.8974 - val_loss: 0.3554 - val_acc: 0.8507
Epoch 4/15
1935/1936 [============================>.] - ETA: 0s - loss: 0.2006 - acc: 0.9188
Epoch 00004: val_acc did not improve from 0.85437
1936/1936 [==============================] - 57s 29ms/step - loss: 0.2007 - acc: 0.9187 - val_loss: 0.3973 - val_acc: 0.8477
Epoch 5/15
1935/1936 [============================>.] - ETA: 0s - loss: 0.1608 - acc: 0.9357
Epoch 00005: val_acc did not improve from 0.85437
1936/1936 [==============================] - 56s 29ms/step - loss: 0.1609 - acc: 0.9356 - val_loss: 0.4618 - val_acc: 0.8450
Epoch 6/15
1935/1936 [============================>.] - ETA: 0s - loss: 0.1284 - acc: 0.9496
Epoch 00006: val_acc did not improve from 0.85437
1936/1936 [==============================] - 56s 29ms/step - loss: 0.1284 - acc: 0.9496 - val_loss: 0.5156 - val_acc: 0.8412
Epoch 00006: early stopping
```

Train 데이터셋의 20%를 Validation 셋으로 활용했으며, 학습 과정은 위와 같다.

손실함수(val_loss)가 더 이상 감소하지 않을 경우 학습을 멈추도록 Early stopping을 설정하고, 손실함수가 최저가 되는 지점(Epoch 2)을 저장하였다.

```
1524/1524 [==============================] - 8s 5ms/step - loss: 0.3549 - acc: 0.8487

 테스트 정확도: 0.8487
```

테스트셋을 이용한 테스트 결과는 약 85%의 정확도를 보여주고 있다.

5. 트위터 크롤링 및 감성분석

앞서 구축한 감성분석기의 분석 대상이 될 트윗을 크롤링한다. 트위터 크롤링을 위해 tweepy 모듈을 이용하였다. 크롤링 과정에서 리트윗(RT)은 한번만 기록하여 중복을 제거하였다. 크롤링 이후에는 각 트윗의 감정상태를 "Negative" 또는 "Positive"로 분류하였다. 또한 긍정과 부정비율을 산출하여 키워드에 대한 전반적인 트렌드를 확인할 수 있도록 하였다.

(1)영화 "침입자" 크롤링 및 감성분석

영화리뷰데이터를 토대로 만든 감성분석기이므로 먼저 영화 "침입자"를 키워드로 감성분석을 실시해보았다.

(https://drive.google.com/file/d/1okKq-9fyH9dGeaQPLBg7tSlT8l3lD8oh/view?usp=sharing , 분석결과)

전체 트윗 중 부정 트윗의 비율은 약 77%를 보이고 있다.

리뷰 내용	감정 분석 결과
아 침입자 괜히 봤어 ....아직도 머리아프고 정신 나간거 같아.....중간에 그만 보고 싶다 생각한 영화 처음이에요...후....귀여운 동주를 봐야겠다	Negative.
<침입자> 후기 : 할일 안하고 영화봐서 벌받은 기분	Negative.
영화 침입자, 송지효님 연기 진짜 쩜	Positive.
RT @itayloryou: 영화 <침입자> 스릴러보다는 추리소설 느낌인데 피곤하고 의욕없는 얼굴을 하고는 광기라는걸 뽑아내는 <송지효 배우님>의 연기가 영화 전체의 긴장감을 끌고가면서 묘하고 섬뜩한 호러를 만들어냄 https://t.co/hwP…	Positive.
RT @B5CXi: <공포의 침입자> 공포도 ★★★ 3 점 자극적 썸네일에 끌려서 봄. 괴기한 사건이 연달아 일어나며 루즈한 감이 없이 금방 시간이 지나가는 킬링타임 영화. 깜놀 구간이 조금 있으며 내용에 비해 엔딩이 다소 허무합니다. 겁이 많은 분이…	Negative.

몇 개의 샘플을 살펴보면, 위와 같이 부정리뷰는 Negative 로 긍정리뷰는 Positive 로 정확하게 분류하는 것을 확인할 수 있다.
또한 영화에 대한 전반적인 트렌드를 확인할 수 있는데, 영화내용이 괴기하여 보는 이로 하여금 불편함을 느낄 수 있다는 부정적인 평가가 많음을 확인할 수 있다. 반면에 배우들의 연기에 대한 평가는 대체로 긍정적이었음을 확인할 수 있다.

리뷰 내용	감정 분석 결과
결말마저 너무 한국적이었던 영화 침입자	Negative.
결백 침입자 야구소녀 볼 영화는 많은데 시간이 없네..	Negative.
결백이랑 침입자 나도못봤어.. 퇴근하고 밥먹고 운동하면 영화 못 봄 .. 전화중국어두 해야함.. (͡ ͜ ʖ ͡)	Negative.
결백이나 침입자가 개봉하긴 했으나 자본이 좀 어느정도 들어간 상업영화라고 볼 수 있는게 살아있다여서 이 영화가 어떻게 되는지에 따라 극장계에 영향이 크지 않을까 침입자 결백 둘다 평소대로 였음 아쉽겠지만 이… https://t.co/kFYOQ92rhl	Negative.

하지만 위와 같이, 긍정과 부정을 확단하기 어려운 상황에서 모두 "Negative"로 분류하면서 산출 결과가 부정 77%라는 극단적인 결과를 만들어냈다.

(2) "진비빔면" 크롤링 및 감성분석

이번에는 최근 오뚜기에서 팔도의 "팔도비빔면"을 추격하기 위해 출시한 "진비빔면"에 대한 대중의 평판을 알아보기 위해 크롤링을 해보았다.

(https://drive.google.com/file/d/1JsLuoHuC6WNzw4BPdbgYGmZEIRUFA52f/view?usp=sharing, 분석결과)

리뷰내용	감성분석결과
점심 진비빔면!!! 팔도꺼하고는 좀 다른맛!!! 맛있어!!! https://t.co/DsxzyIuUI4	Positive.
아 진비빔면 너무 맵다	Negative.
@Only_Luv_BTS 크킄ㅋ 저번에 진비빔면 먹어봤는데 별루였어요..팔도가 짱	Negative.
@P_Simsya 헉 저희집에도 진비빔면있어요! 제입맛에는 꽤 맛있었던거 같아요!!😊 저...저녁을 5시쯤 먹어서 지금 와서 보니 배고프더라구요...뭐먹을수도없어서 굶주린채로 있지만...ㅜㅡㅜ어서 내일되고 학교도마쳤음좋겠네요! 진비빔면 먹어야되니까!ㅠ	Positive.
진비빔면이 팔도비빔면보다 맛있다	Negative.
진비빔면마싯겟다	Negative.
@zmom_ 진비빔면 진쨔 맛있어요 요새 그거때문에 계속 얼음 만드는 중...	Negative.
진비빔면 맛있어	Negative.

전체 분석결과 중 일부 트윗을 살펴보자면, 같은 "맛있다"라는 의미를 담고 있는 트윗이라도 일부는 "Positive"로 일부는 "Negative"로 표현하는 것을 확인할 수 있다. 이는 감성분석기가 영화리뷰 데이터를 기반으로 만들어졌기 때문임을 예상할 수 있다. 영화에 대한 평을 "맛있다" 또는 "맛없다" 로 표현하지 않기 때문에 이에 대한 학습이 이루어지지않아 분류를 랜덤하게 하는 것으로 추측된다.

Chapter3. 결론 및 한계점

기업이 자사 제품 및 브랜드에 대한 트렌드를 쉽게 확인할 수 있도록 소셜미디어 감성분석모델을 구축해보았다. 구축한 모델을 활용하여 특정 키워드에 대한 긍정/부정 비율을 산출하여 전체적인 평판을 확인할 수 있었다. 하지만 긍정/부정을 명확하게 나눌 수 없는 텍스트를 분류함에 있어 문제가 발생하여 산출 결과가 극단으로 치우는 경향을 확인할 수 있었다. 이를 해결하기 위해 "중립"을 추가하는 등 좀 더 세분화하여 분류하는 모델을 구축할 수 있다. 또한 영화 리뷰 데이터를 기반으로 만든 감성분석기이기에 다른 분야에 대한 감성분석을 실시할 때는 정확도가 떨어지는 것을 확인할 수 있었다. 왜냐하면 각 산업 또는 분야마다 사용하는 어휘가 다르기 때문에, 기계가 그에 대한 학습이 되지 않으면 성능이 저하되기 때문이다. 따라서 기업이 분석하고자 하는 대상 및 목적에 맞는 감성분석기를 구축할 필요가 있음을 확인할 수 있었다.