

Week11 : Text Analysis Assignment Report

제출일: 2021.05.09

학번: 2019312072

이름: 정주호

1. 코드 설계 및 코드 설명

먼저 이미 전처리 된 한글 파일인 “naver_preprocess.csv” 파일에서 각 문장마다 키워드를 추출하여 한글의 긍정 부정을 나누어 놓은 “polarity.csv” 파일에 기반하여 각 키워드에 긍정, 부정 점수를 부여한다.

1) 나의 설계 : 각 문장별로 읽어서 KRWordRank로 키워드로 분할한 뒤 해당 키워드를 “polarity.csv”파일을 dataframe으로 읽은 뒤 search하려고 했다

2) 봉착한 문제 : “polarity.csv”파일에 한글 영역인 ngram 열이 키워드로 이루어져 있는 것이 아니라 n-gram으로 적혀있어 키워드로 search가 되지 않는다.

ex) ‘가’ -> '가*/JKS'

```
[ ] df = pd.read_csv('polarity.csv')
df.head()
```

	ngram	freq	COMP	NEG	NEUT	None	POS	max.value	max.prop
0	가*/JKS	1	0.0	0.0	0.0	0.0	1.0	POS	1.0
1	가*/JKS;있/VV	1	0.0	0.0	0.0	0.0	1.0	POS	1.0
2	가*/JKS;있/VV;있/EP	1	0.0	0.0	0.0	0.0	1.0	POS	1.0
3	가*/VV	3	0.0	0.0	0.0	0.0	1.0	POS	1.0
4	가*/VV;ㄴ다*/EF	1	0.0	0.0	0.0	0.0	1.0	POS	1.0

3) 해결 방안: 문장을 키워드로 나눌때 키워드로 추출하는 것이 아니라 ngram으로 나눌 수 있다면 “polarity.csv”파일에서 바로 해당 ngram을 찾을 수 있을 것이다.

```
[ ] from konlpy.tag import Okt
tokenizer = Okt()
print(tokenizer.morphs(lala_review.text.iloc[0]))

['극장', '에서', '3', '번', '이나', '봤습니다', '그래도', '보고싶은', '너무나', '행복한', '영화', '입니다']
```

-> LAB 강의에서 배운 OKT로 tokenize하면 문장이 한글 단어로 분할이 되기 때문에 ngram으로 찾기가 어렵다 (한계점1)

```
[▶] #문장을 n-gram으로 만들기
from konlpy.tag import Komoran
komoran = Komoran()
words = komoran.pos(lala_review.text.iloc[0], join=True)
```

-> konlpy 내장 모듈 중 Komoran을 사용하여 문장을 ngram으로 분할하여 이를 해결하고자 하였다.

```
[▶] for i in words:
      print(i)
```

```
[👤] 극장/NNG
     에서/JKB
     3/SN
     번/NNB
     이나/JX
     보/VV
     았/EP
     습니다/EC
     그래도/MAJ
     보/VV
     고/EC
     싶/VX
     은/ETM
     너무나/MAG
     행복/NNG
     하/XSV
     ㄴ/ETM
     영화/NNG
     이/VCP
     ㅂ니다/EC
```

4) Sentiment Score 구하기


수업시간에 배운 'Vader' 방식이나, 'Sentiwordnet' 방식을 사용하였으나 영어 기반이기때문에 **synsets**에서 태그가 되지 않는다. (한계점2) 따라서 **Sentiment Score**을 직접 구하는 함수를 작성하였다.


* **Total Sentiment Score = positive score - negative score**

- **ngram**으로 분할한 각 문장 별로 각 단어에 해당하는 긍정, 부정 점수를 "polarity.csv" 파일에서 확인한다. 긍정 단어 점수들의 총합과 부정 단어들의 점수의 총합을 구하여 **Total sentiment score**을 구한다.
- 해당 문장의 **Total Sentiment Score**가 0점이상이면 **positive(1)**, 이하이면 **negative(0)**을 부여한다.

```
[ ] sentiment = 0.0
    pos = 0.0
    neg = 0.0
    for word in words:
        neg += df[df['ngram'] == word]['NEG'].values.sum()
        pos += df[df['ngram'] == word]['POS'].values.sum()

    sentiment = pos-neg
```

 sentiment

 1.480177167

5) Labeling

“naver_preprocess.csv” 파일에 sentiment 열이 따로 없고 score만 있기 때문에 score 바탕으로 1~5점이면 negative(0), 6~10점이면 positive(1)로 라벨링한 후 열을 추가한다.

```
[ ] lala_review['label'] = np.where(lala_review['score']>=6,1,0)
```

lala_review.head(15)

	score	text	idx	user	written_at	agree	disagree	body	label
0	10	극장에서 3번이나 봤습니다 그래도 보고싶은 너무나 행복한 영화입니다	12180460	후니뽀(beau****)	2017.01.15 17:00	1	1	극장에서 3번이나 봤습니다 그래도 보고 싶은 너무나 행복한 영화입니다	1
1	10	꿈의나라로~!!!!	12180563	더리버리(rapp****)	2017.01.15 17:16	2	1	꿈의 나라로	1
2	9	반전과 공감이 있는 로맨스 영화	12180612	조윤경(jykj****)	2017.01.15 17:26	2	1	반전과 공감이 있는 로맨스 영화	1
3	8	마지막에 남녀가 만나서 사랑을 이뤘으면 행복한 결말이었지만 그렇지 못해서 더 애잔함...	12180615	상하(shtb****)	2017.01.15 17:27	2	0	마지막에 남녀가 만나서 사랑을 이뤘으면 행복한 결말이었지만 그렇지 못해서 더 애잔함...	1

2. 한계점

- 1) “polartiy.csv” 파일에서 ngram으로 표현되어 있는 단어들에서 한글만 뽑아낼 수 있다면 OKT를 이용하여 키워드비교가 가능할 것 같다. 본 연구에서는 ngram에서 키워드를 추출하는 방법을 찾아내지 못하여 돌아간 것 같다.
- 2) Sentiment score을 구할 때 기존 Lab에서 실행한 ‘vader’방식이나 ‘sentiwordnet’함수를 수정하면 구할 수 있을 것이라 생각했다. 본 연구에서는 수정할 때 마다 오류가 나서 한글이기 때문에 다를 것이라 생각하여 sentiment score을 구하는 함수를 새롭게 만들었지만 만약 기존 ‘vader’방식이나 ‘sentiwordnet’함수를 수정하여 구할 수 있다면 훨씬 더 간단해질 것 같다.