Python을 활용한 텍스트 마이닝 6.텍스트 분석-영문 텍스트 마이닝1

http://www.imdb.com/title/tt0110912/?ref_=nv_sr_1

영어로 된 영화 평점 사이트에서 리뷰를 크롤링하고 간단한 텍스트 마이닝을 해보겠습니다.

```
import nltk
from nltk.corpus import stopwords
stopWords = set(stopwords.words('english'))
lemmatizer = nltk.wordnet.WordNetLemmatizer()


with open('dark_knight_review.txt','r',encoding ='utf-8')
    lines = f.readlines()
    f.close()

reviewedList=[]

for line in lines:
    reviewed=''
    tokens = nltk.word_tokenize(line)    ##토큰화

    for token in tokens:
        if token.lower() not in stopWords:      ##stopwor
            reviewed += ' '+lemmatizer.lemmatize(token)
    reviewedList.append(reviewed)
```

먼저 토큰화(tokenize), 기본형 형태로 만들기(lemmatize)와 stopwords를 제거하는 작업을 했습니다.

Dark knight라는 영화의 리뷰 100개를 크롤링하여 만든 텍스트 파일을 불러와서 위의 순서로 코딩했습니다. 참고로 stopwords의 경우 소문자의 형태만 인식하기 때문에 .lower()를 통해 토큰을 전부 소문자로 변환했습니다.

```
with open('dark_lemmatized.txt','w',encoding='utf-8') as f
    for reviewed in reviewedList:
```


Search

검색

Recently posted

Recommendation System 1. 5월 13, 2018

Regression with Machine Learning 4. Regularization for spartsity(희소 학습) 4월 8, 2018

경사 하강법(Gradient Descent) 3월 27, 2018

Regression with Machine Learning 3. Constrained Least Squares(제약 최소제곱) 3월 18, 2018

Regression with Machine Learning 2. Stochastic Gradient Descent(확률적 경사법) 3월 18, 2018

Posts

2018년 5월 (1)

2018년 4월 (1)

결과를 다시 텍스트 파일로 저장하고...

이제 이 리뷰에서 가장 많이 등장하는 명사를 추출해보겠습니다.

혹시 중간에 nltk패키지의 오류로 실행이 안되면 nltk.download()를 입력해서 필요한 것을 받을 수 있습니다.

```
import nltk
from collections import Counter

nounlist=[]

with open('dark_lemmatized.txt','r',encoding ='utf-8') as
    lines = f.readlines()
    f.close()

for line in lines:
    tokens = nltk.word_tokenize(line)      #토큰으로 만들기
    tags = nltk.pos_tag(tokens)            #토큰 별 품사를 t

    for word, tag in tags:
        if tag in ['NN','NNS','NNP','NNPS']:      #명사를 뽑
            nounlist.append(word.lower())

counts = Counter(nounlist)
print(counts.most_common(10))
```

```
[('movie', 340), ('batman', 249), ('film', 247),
('joker', 177), ('dark', 113), ('ledger', 100),
('knight', 95), ('heath', 88), ('time', 83), ('action',
67)]
```

결과를 출력하면 다음과 같이 리뷰에서 가장 많이 등장한 10개의 단어가 빈도 순으로 출력된 것을 볼 수 있습니다.

동사와 형용사도 동일한 방법으로 추출하시면 됩니다.

전체 리뷰의 토큰 개수를 구할 수도 있습니다.

```
import nltk
from nltk.corpus import stopwords
stopWords = set(stopwords.words('english'))

with open('dark_knight_review.txt','r',encoding ='utf-8')
    lines = f.readlines()
    f.close()

token_numbers=[]

for line in lines:
    line = nltk.word_tokenize(line.lower())
1
1
                token_numbers.append(word)
```

File failed to load: /extensions/MathMenu.js

2018년 3월 (5)
2018년 1월 (1)
2017년 9월 (2)
2017년 8월 (6)
2017년 7월 (12)
2017년 6월 (7)

Etc

사이트 관리
로그아웃
글 RSS
댓글 RSS
WordPress.org

category

misc (2)
Python (17)
- coding with python (1)
- installation (5)
- Neural Network (1)
- Text Mining (10)

R (14)
- 2017 Weather Contest (4)
- machine learning (5)
- Packages & Base (2)
- Recommendation System (2)

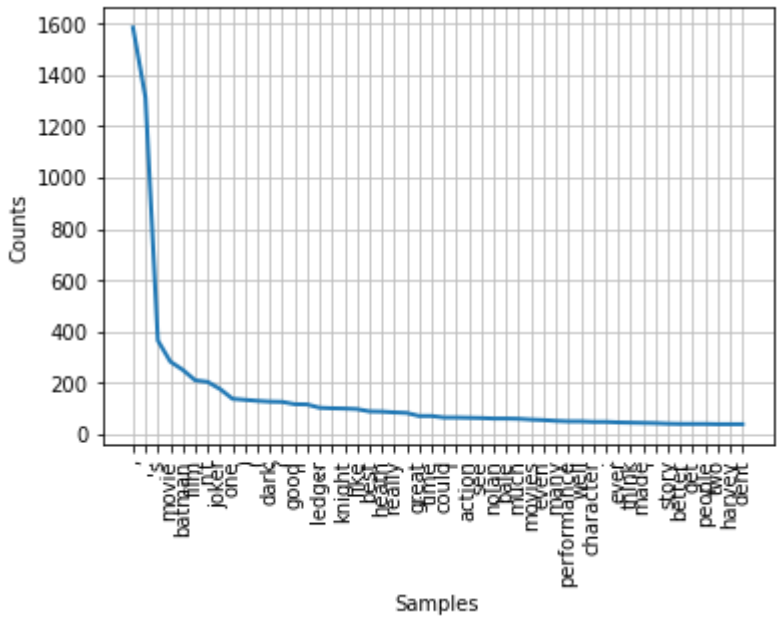
Visualization (2)

```
16
17 corpus = nltk.Text(token_numbers)
18
19 print(len(corpus.tokens))       #토큰의 수
20 print(len(set(corpus.tokens)))          #중복 아닌 토큰 수
```

```
20575 6069
```

그래프를 이용해서 토큰의 등장 횟수를 시각화 할 수도 있습니다

```
1 corpus.plot(50)
```



'이나 's와 같은 보통의 상황에서는 불필요한 단어들이 많이 등장하는데 stopwords에서 추가적으로 옵션을 두어 필터링이 되도록 하는 방법도 있습니다.

kis0403 7월 16, 2017 Text Mining 댓글 없음 편집

← 2017 날씨 빅데이터 콘테스트 2. 데이터 탐색(EDA)

Python을 활용한 텍스트 마이닝 7.텍스트 분석-영문 텍스트 마이닝2 →

답글 남기기

kis0403로(으로) 로그인 함. 로그아웃?

댓글

댓글 달기