영화 리뷰 감성분석

AI_17기 김재윤

Section 2 Challenge

문제: 감성분석

- 감성분석: Sentiment Analysis
  - 텍스트에 나타난 태도, 의견, 성향과 같은 주관적 데이터를 분석하는 자연어 처리 기술
  - 유권자의 태도, 소셜 미디어 분석, 리뷰 분석, 가짜뉴스 분석 등 여러가지 방면에서 중요한 응용처를 보여주고 있음

- 출처: http://www.konantech.com/core/wp-content/uploads/2014/06/TechReport_Sentiment_Analysis%EA%B9%80%EB%AC%B8%ED%9D%AC%ED%8C%80%EC%9E%A5%EB%8B%98%EC%9D%B8%ED%84%B0%EB%B7%B0.pdf

데이터: imdb의 영화 리뷰 데이터

IMDB Movie Reviews (Binary Sentiment)

- 24801개의 긍정/부정 평가 훈련 데이터
- 24909개의 테스트 데이터
- 이루어진 데이터 자체는 영화평/감성평가(0/1)로 심플

- 감성 평가 데이터로서 리뷰 데이터는 가장 기본적인 데이터이고, 분량이 충분히 많다고 판단되었으며, 케글에 이미 풀린 코드가 없었음.

- 출처: 케글, https://www.kaggle.com/datasets/thedevastator/imdb-large-movie-review-dataset-binary-sentiment

목표: ?

가설 증명보다는 관찰해보고 싶은 것이.

- 가지고 있는 재료가 부족한 만큼, 목표로 특정 독립변수가 이러하면~ 어떤 결과가 나올것이라는 결론을 내리기는 어려움.

- **물음 1.** 단순히 단어의 뉘앙스만을 판별하는 감성평가가 과연 정확도를 높게 담보할 것인가?
  - 'good'이라는 단어가 들어가면 긍정이라고 판단하는 모델은 옳은가? 반어법이라거나, 인용, 여러가지 다양한 발화 방식에 의해 긍정어가 들어가는 부정적 평가가 가능하지 않은가?
- **물음2.** 그럼에도 불구하고 단어만으로 감성평가가 된다면, 대체 어떤 단어들이 감성평가의 키가 될 것인가?
- **물음3.** 하이퍼 파라미터와 관련된 여러가지 궁금증 (후술 예정)

분석: 인코딩

- 자연어 데이터는 전처리 과정이 특히나 중요함.
- 가장 쉬운 발상: **단순히 센다 - Count**: ‘머신러닝’이라는 단어가 100번 나왔으므로 머신러닝에 100의 가중치를 준다.

- 하지만 모든 문서에서 계속 자주 나오는 단어(문법어휘 등)이 중요한 단어일까?
  - ex)’나는’ 이라는 말이 들어가면 긍정적인 리뷰인가?

분석: 인코딩

- **단어 빈도(Term Frequency)**: 한 문장 내에서 단어가 나온 빈도: '머신러닝'이 한 문장에서 나온 빈도: '나는 머신러닝을 배운다'에는 머신러닝이라는 단어가 1번 들어가므로 TF=1
- **문서 빈도의 역수(Inverse Document Frequency)**: 전체 문서에 출연한 빈도. 100개의 예문중 '머신러닝'이 나온 문서는 50개이므로 IDF=0.5 (실제로는 공식에 의해 약 log2)

- **TF-IDF** : TF와 IDF의 곱.
  - TF: 한 문장에서 자주 나온 단어이되
  - IDF: 일부 문서에서만 등장하는 단어에 가중치를 둔다.

분석: 인코딩

- 전처리 추가요소: input 자체는 하나였기 때문에, 이것을 다양한 방식으로 인코딩 하는 것으로 feature selection을 대신할 수 있었다!
- **max_df** : 0.5인경우, 50% 이상의 문서에 등장하는 단어는 제외. 0.5~0.75까지 실험
- **max_feature**: 영향력이 높은 단어를 선택적으로 골라 분석에 활용. feature importance등을 추가로 측정할 필요 없이 필요한 feature를 즉시 체크할 수 있었다. 60000~500까지 실험
- **ngram_range**: 'good'과 'movie'가 들어가면 좋은영화? 'good movie'가 들어가면 좋은영화이지 않을까? 연이은 단어를 한 개체로 처리하였을 경우 어떻게 될 것인가? (1,1)부터 (1,3)까지 실험
- **물음3(revisit).** 적은 파라미터, 적은 데이터로 어떤 예측이 가능할까?

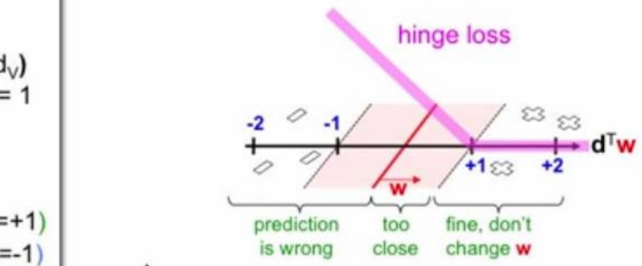
분석: 모델 설정

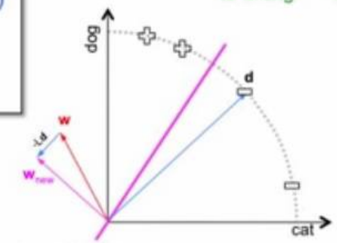
- 회귀 or **분류**?
- 분류 문제. 대용량의 텍스트 데이터 처리에 적합한 **Passive Aggressive Classifier** 라는 분류기를 사용.
- 데이터 하나를 받아, 선형벡터를 업데이트하고 즉시 데이터를 버리는 방식으로 업데이트가 계속되는 분류기.
- 출처: https://www.youtube.com/watch?v=TJU8NfDdqNQ

Passive Aggressive Algorithm

- On-line algorithm: learn from massive streams of data
  - get an example, update classifier, throw away example

initialize $\mathbf{w} = (0,\ldots,0)$
monitor a stream:
  receive new doc $\mathbf{d} = (d_1 \ldots d_V)$
  apply tf.idf, normalize $\|\mathbf{d}\| = 1$
  predict positive if $\mathbf{d}^T\mathbf{w} > 0$
  observe true class: $y = \pm 1$
  want to have:
    $\mathbf{d}^T\mathbf{w} \geq +1$ if positive ($y=+1$)
    $\mathbf{d}^T\mathbf{w} \leq -1$ if negative ($y=-1$)
  same as: $y(\mathbf{d}^T\mathbf{w}) \geq 1$
  loss: $L = \max(0, 1 - y(\mathbf{d}^T\mathbf{w}))$
  update: $\mathbf{w}_{new} = \mathbf{w} + yL\mathbf{d}$





Aggressive:
$\mathbf{d}^T\mathbf{w}$ ... was L "short" of y
$\mathbf{d}^T\mathbf{w}_{new} = \mathbf{d}^T(\mathbf{w} + yL\mathbf{d})$
$= \mathbf{d}^T\mathbf{w} + yL\mathbf{d}^T\mathbf{d}$
$= \mathbf{d}^T\mathbf{w} + yL = y$

분석: 모델 설정

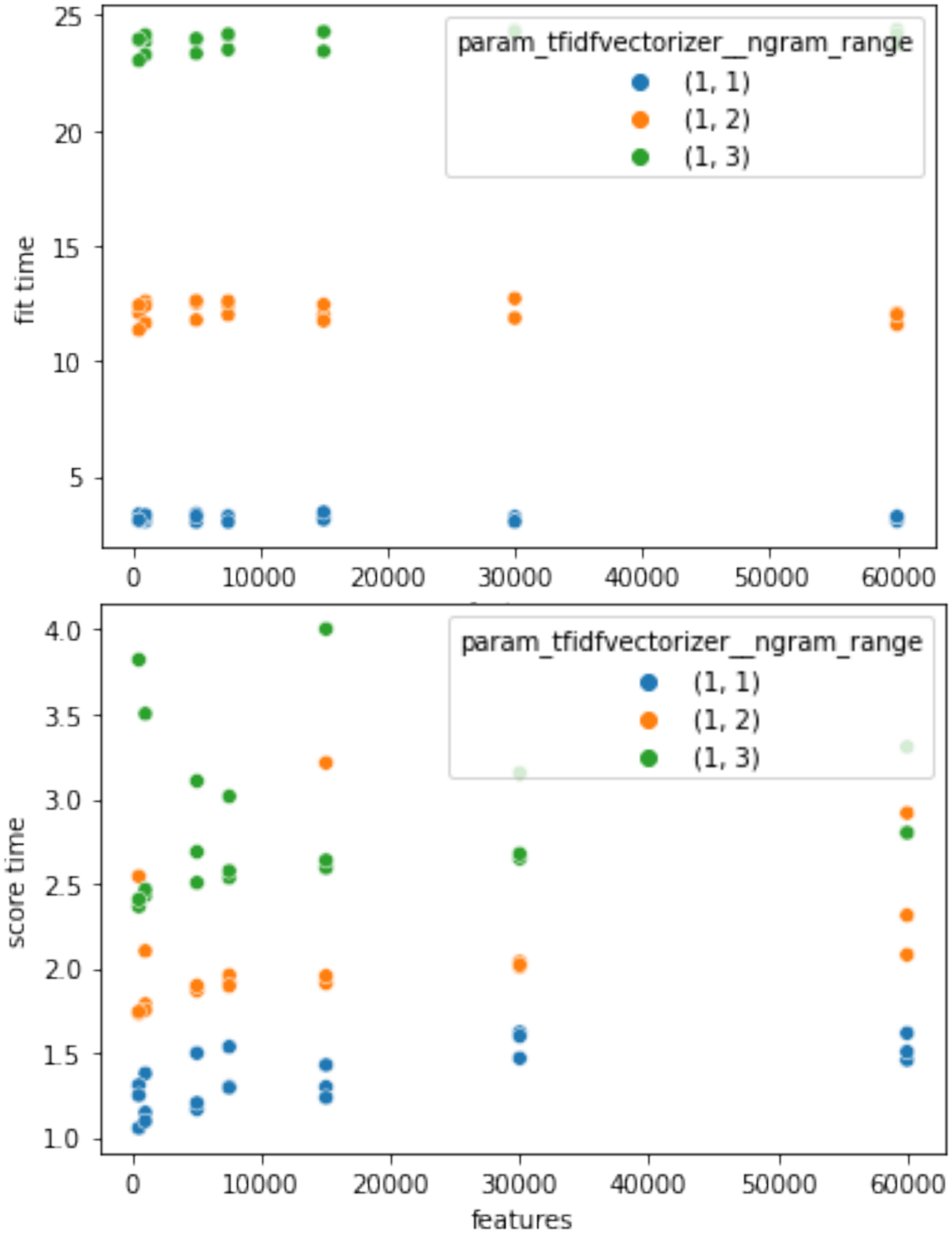
- 훈련 데이터의 참 거짓 라벨 비율이 50:50이므로 어느 지표를 써도 괜찮지만(최빈값 기준 모델 성능 0.5), 많이 사용해온 **f1스코어**를 중심으로 성능 책정.
- 모델의 성능 책정에 대한 또다른 지표로, 스코어 뿐만 아니라 **소요 시간**도 같이 책정해보도록 함. 인코딩 방식에 따라 사용하는 용량과 처리시간의 차이가 극단적인 차이가 나기 때문.

- **GridSearch, EarlyStopping**으로 파라미터 서치를 진행.

결과 해석

소요시간 / ngram

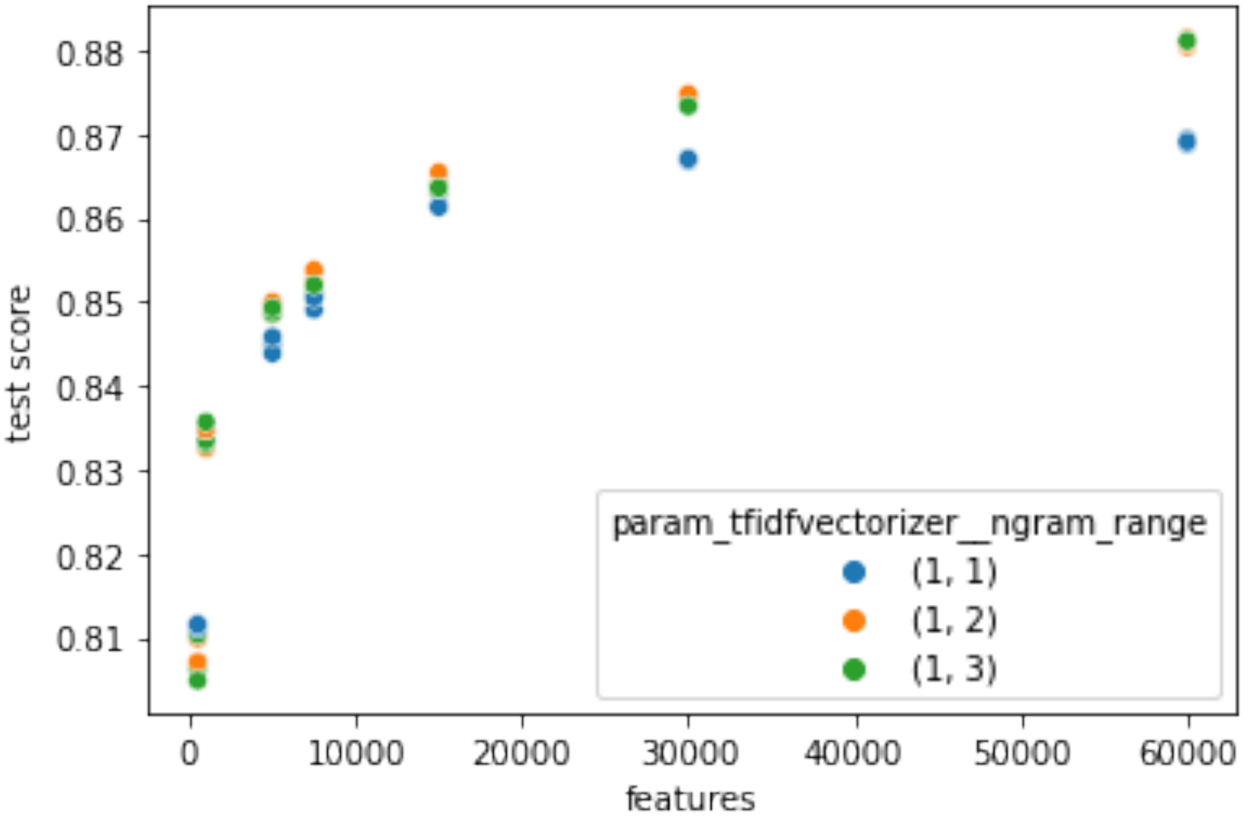
- Fit Time: ngram (1,1)일 경우 3초 근방, (1,2)이면 12초 근방, (1,3)이라면 24초 근방의 fit 시간이 걸림.
- Score time: 반면 측정 시간은 차이가 있기는 하나, 크지는 않음(1초남짓씩 차이)



소요시간 / ngram

- (1,1) (1,2) 사이는 시간차가 나는 만큼 성능차가 나지만 (1,2) (1,3)사이는 시간차가 있으나 성능은 오히려 떨어지는 것을 볼 수 있음

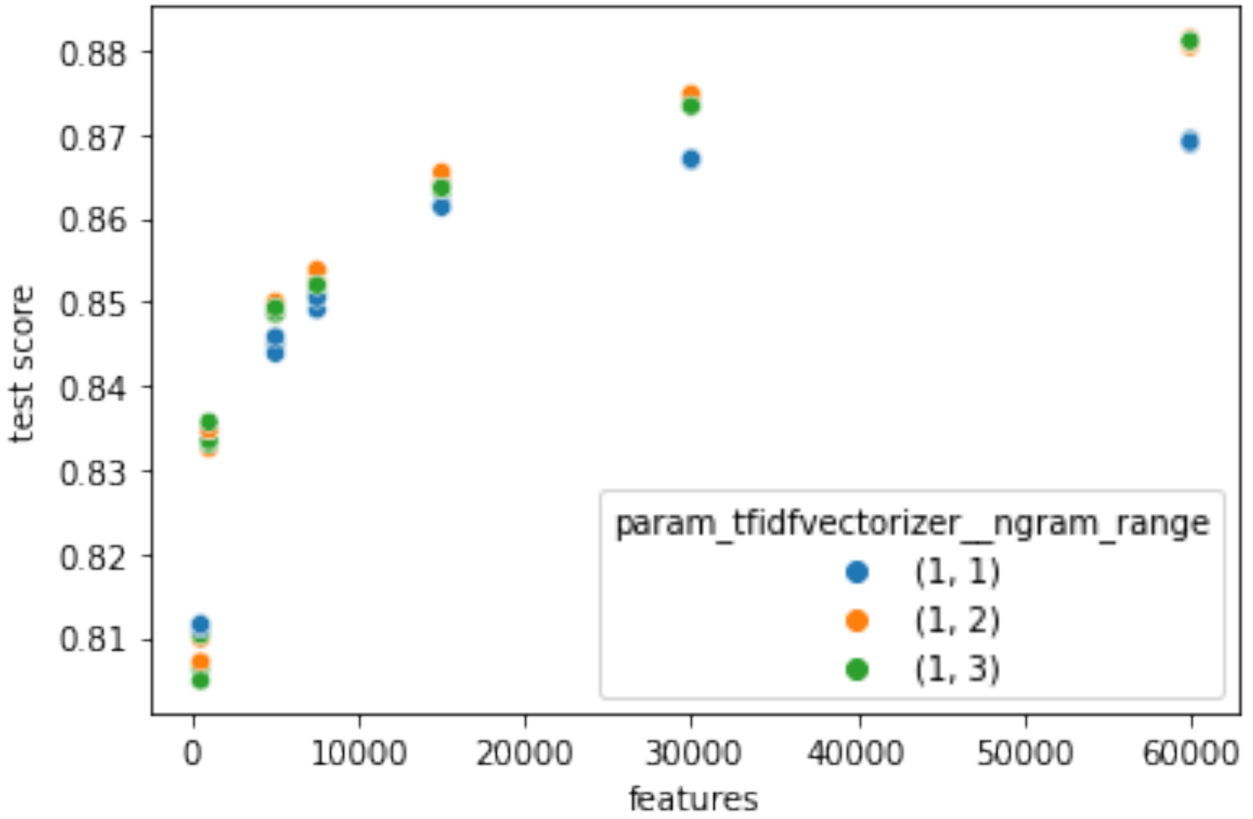
- 실제로 관찰시 importance가 높은 feature 상위권에 windowsize가 2 이상인 feature는 별로 없음.



feature / Ngram

- feature의 갯수에 따라 정확도는 log 그래프를 그리며 증가하는 것을 볼 수 있음.

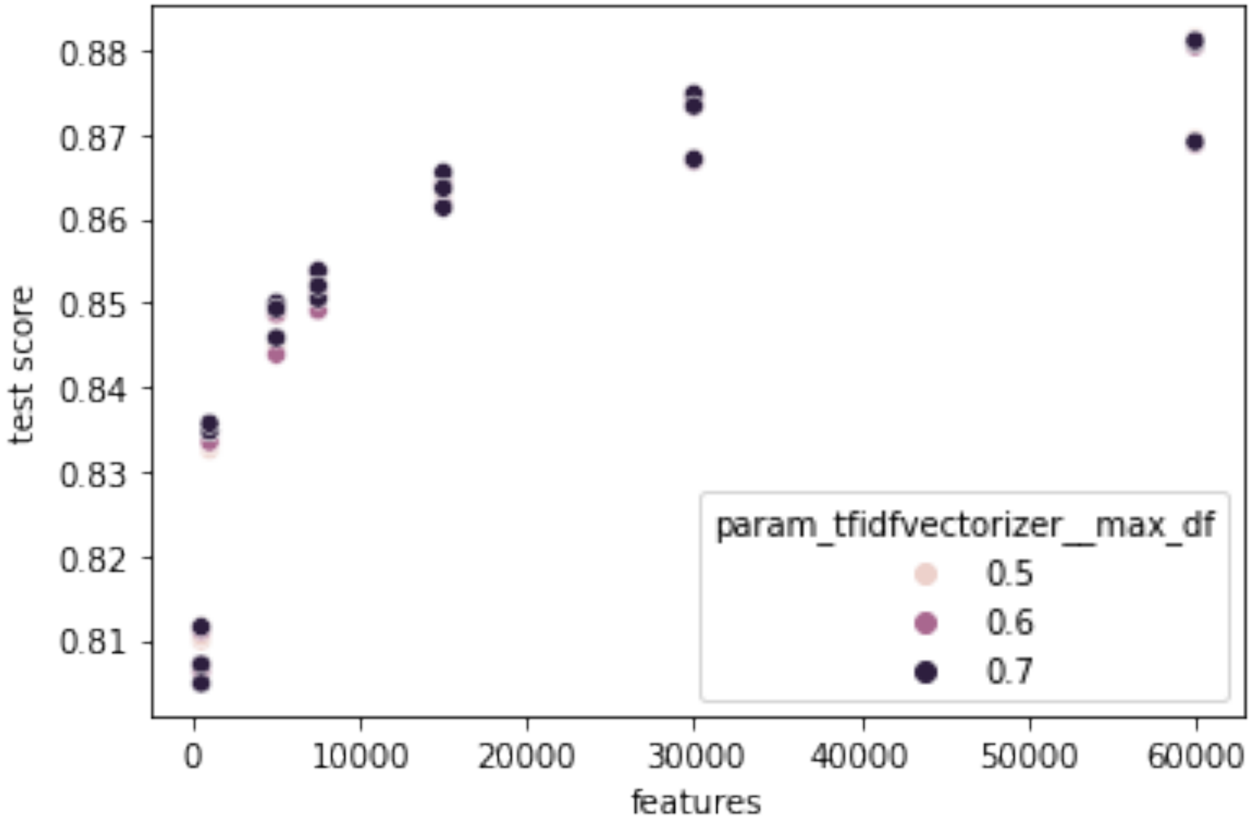
- 대략적인 근사를 해 본다면:
  - ngram* log(feature) = score
- 꼴의 식이 나올 것으로 예상



feature / max df

- max_df는 큰 영향력을 발휘하지 못함.

- 굉장히 뭉쳐있으나 값이 낮은경우 미세하게 성능이 좋아지는 것을 관찰할 수 있음.



단어 분석

- 호오를 나타내는 표현: like, good, bad, great, don
- 영화에 관한 직접적인 묘사 표현: story, movies, watch, acting, characters, plot, films, seen
- 부사: just, really
- time, think, people, make…

like	370.495628
just	347.584591
good	325.926006
story	281.414862
really	275.370764
time	274.542111
bad	256.048488
great	239.955702
people	231.844883
don	227.779597
movies	222.566435
watch	206.624304
think	202.072798
make	197.835911
way	196.858234
seen	195.172650
acting	194.333971
characters	192.685719
plot	190.352167
films	185.927395

like
just
really
good
story
people
time
great
watch
don
bad
make
seen
movies
characters
character
think
scenes
end
does
little
love
know
acting
plot
way
better
life
did
films
actors
best
director
scene
years
things
going
old
say
thing
horror
didn
watching
watched
actually
look
want
ve
new
real
man
end
comedy
young
series
pretty


like
just
good
story
time
people
bad
seen
acting
funny
way
scene
films
lot
years
world
look
action
pretty
girl
old
role
new
sure
big
ll
script
know
quite
better
guy
kind
book
tv
dvd
worst
fun
performance
actually
10
work
comedy
long
bit
saw
young
does
doesn
don
movies
life
little
ve
got
things
going
make
character
watching
thought
director
actors
isn
thing
best
say
cast
fact
real
feel
right
love
end
man
didn
probably
family
scenes
did
interesting
makes
want
far
original
times
think
great
watch
music
point
plot
really
characters
horror
gets
hard
minutes
come
series

결론: 실험결과

훈련 정확도: 0.9890680377377329
검증 정확도: 0.892764080978152
테스트 정확도: 0.8626198083067093

- **물음 1.** 단순히 단어의 뉘앙스만을 판별하는 감성평가가 과연 정확도를 높게 담보할 것인가? (된다.)
- **물음2.** 그럼에도 불구하고 단어만으로 감성평가가 된다면, 대체 어떤 단어들이 감성평가의 키가 될 것인가? (생각보다 정말 직관적인 호오표현, 부사, 영화관련 표현들)
- **물음3.** 적은 파라미터, 적은 데이터로 어떤 예측이 가능할까? (500단어 voca로 **80.28**)