LSTM 모델 기반 영화 리뷰 긍/부정 예측

그영화봤조 김건희 윤남기 정지훈

@2021.06.10 멋쟁이사자 고급 인공지능 과정

네이버 영화 리뷰 감성 분석 목차

발표 목차

1. 네이버 영화 리뷰 사용 데이터 & 라이브러리
2. 데이터 전처리
   2-1. 기초 데이터 전처리 : 데이터 탐색 및 중복 제거
   2-2. 딥러닝 학습을 위한 전처리 : 불용어처리, 토큰화, 정수인코딩, 패딩
3. LSTM을 사용한 모델링과 학습
4. 리뷰 감성분석
5. 네이버 영화 리뷰 워드 클라우드
6. 프로젝트 후

개요

네이버 영화 리뷰 데이터를 이용하여 데이터 전처리를 거쳐
딥러닝을 활용한 LSTM 모델링, 시각화를 통한 *감성분석에 그 의의를 둠.

* 감성분석 (Sentiment Analysis) : 텍스트에 들어있는 의견이나 감성, 평가, 태도 등의 주관적인 정보를 컴퓨터를 통해 분석하는 과정

1. 네이버 영화 리뷰 데이터

- 회원 아이디(id), 영화리뷰(document), 긍/부정 유무(label)로 구성된

15만개의 리뷰 데이터를 학습 데이터로, 5만개의 리뷰 데이터를 테스트 데이터로 활용

	id	document	label
0	9976970	아 더빙.. 진짜 짜증나네요 목소리	0
1	3819312	흠...포스터보고 초딩영화줄....오버연기조차 가볍지 않구나	1
2	10265843	너무재밋었다그래서보는것을추천한다	0
3	9045019	교도소 이야기구먼 ..솔직히 재미는 없다..평점 조정	0
4	6483659	사이몬페그의 익살스런 연기가 돋보였던 영화!스파이더맨에서 늙어보이기만 했던 커스틴 ...	1

< 네이버 영화 리뷰 데이터 예시>

1. 네이버 영화 리뷰 데이터 분석을 위한 사용 라이브러리

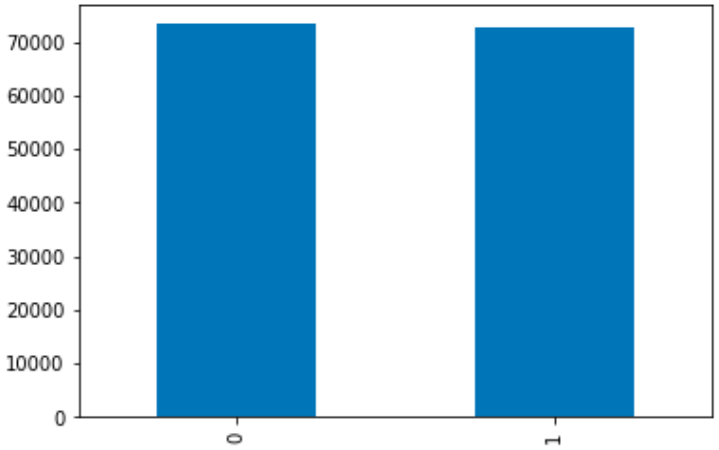
- TenserFlow라이브러리의 Keras모델을 이용한 다양한 클래스를 사용.

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import re    #전처리 축약표현 사용
from konlpy.tag import Okt        #트위터분석기(한글분석-데이터수집처리 ex.0109)
from tensorflow.keras.preprocessing.text import Tokenizer    #integer Encoding
from tensorflow.keras.preprocessing.sequence import pad_sequences  #패딩하기
from tensorflow.keras.layers import Embedding, Dense, LSTM
from tensorflow.keras.models import Sequential,load_model
from tensorflow.keras.optimizers import Adam
from tensorflow.keras.callbacks import EarlyStopping, ModelCheckpoint
%matplotlib inline
```

2. 데이터 전처리 : 기초 데이터 탐색 및 중복 제거

- 영화 리뷰 데이터의 label 컬럼은 긍정(1), 부정(0)으로 labeling 되어 있음
- 학습 데이터 및 평가 데이터 리뷰(document)에 중복 리뷰가 포함되어 있어 이를 제거함

<학습 데이터 중복 제거 전>

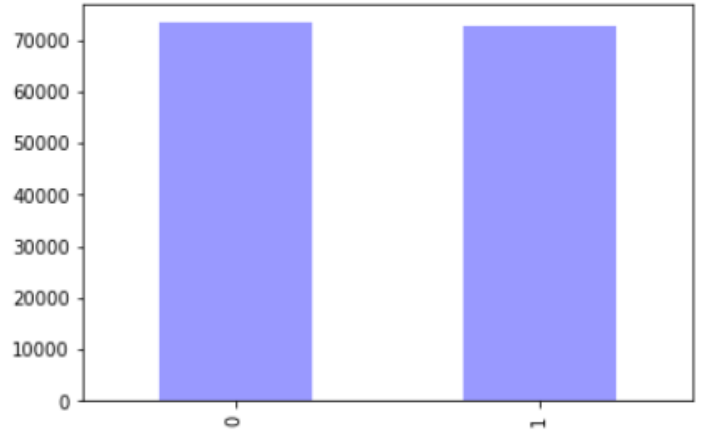


label	count
0	75173
1	74827

부정(0) : 75,173건
긍정(1) : 74,827건
총 150,000건

3,817건 소거

<학습 데이터 중복 제거 후>

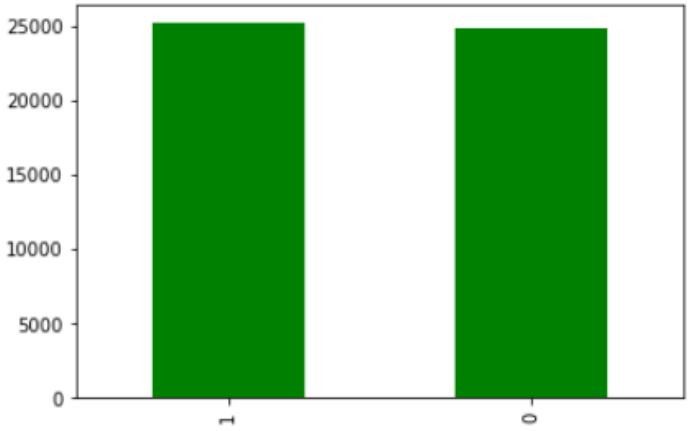


	label	count
0	0	73342
1	1	72841

부정(0) : 73,342건
긍정(1) : 72,841건
총 146,183건

2. 데이터 전처리 : 기초 데이터 탐색 및 중복 제거

<평가 데이터 중복 제거 전>

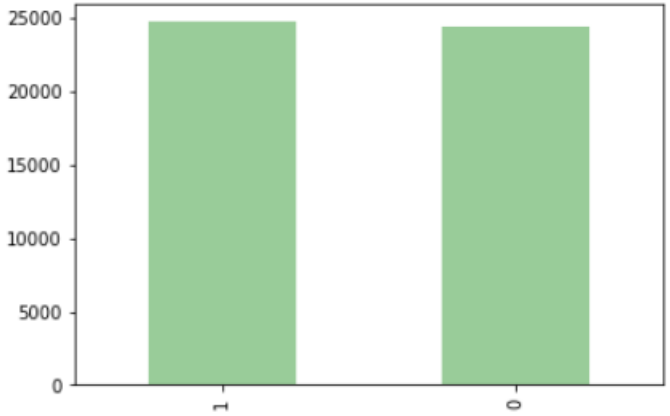


	label	count
0	0	24827
1	1	25173

부정(0) : 24,827건
긍정(1) : 25,173건
총 50,000건

842건 소거

<평가 데이터 중복 제거 후>



	label	count
0	0	24446
1	1	24712

부정(0) : 24,446건
긍정(1) : 24,712건
총 49,158건

2. 데이터 전처리 : 불용어 처리(Stowords)

- 불용어처리를 통해 리뷰 데이터의 토큰화를 위해 부적절한 단어를 제거하는 과정을 거침

document
굳 ㅋ
GDNTOPCLASSINTHECLUB
뭐야 이 평점들은 나쁘진 않지만 10점 짜리는 더더욱 아니잖아
지루하지는 않은데 완전 막장임 돈주고 보기에는
3D만 아니었어도 별 다섯 개 줬을텐데 왜 3D로 나와서 제 심기를 불편하게 하죠
음악이 주가 된, 최고의 음악영화
진정한 쓰레기
마치 미국애니에서 튀어나온듯한 창의력없는 로봇디자인부터가,고개를 젖게한다
갈수록 개판되가는 중국영화 유치하고 내용없음 폼잡다 끝남 말도안되는 무기에 유치한c...
이별의 아픔뒤에 찾아오는 새로운 인연의 기쁨 But, 모든 사람이 그렇지는 않네

1) 한국어 불용어 모음 파일 활용
2) 단모음 제거
3) 특수문자 제거

document
평점 나쁘진 않지만 더욱 아니잖아
지루하지는 않은데 완전 막장 주고 보기 에는
아니었어도 다섯 줬을텐데 나와서 심기 불편하게 하죠
음악 주가 최고 음악 영화
진정한 쓰레기
마치 미국 애니 에서 튀어나온듯 창의력 없는 로봇 디자인 부터가 고개 젖게 한다
갈수록 개판 되가는 중국영화 유치하고 내용 없음 잡다 끝남 안되는 무기 유치한 남무...
이별 아픔 찾아오는 새로운 인연 기쁨 모든 그렇지는 않네
괜찮네요 오랜 포켓몬스터 잼밌
한국 독립영화 한계 그렇게 아버지 된다와 비교

단모음, 특수문자를 포함한 불용어 제거 전으로
무분별한 영어의 남발과 단모음, 특수문자로 인해 딥러닝 학습에 제약이 있음

불용어 처리 이후 토큰화를 위한 준비가 되어 있는 과정

=> 위 과정을 거쳐 정제된 최종 학습 데이터 샘플 : 145,278건 / 테스트 데이터 샘플 48,789건

2. 데이터 전처리 : 토큰화(Tokenization)

- 토큰화(Tokenization) 를 통해 리뷰 각 문장 마다 단어 단위와 의미를 갖는 문자열로 나누는 과정을 거침

```
[['아더', '빙', '진짜', '짜증나다', '목소리'],
 ['흠', '포스터', '보고', '초딩', '영화', '줄', '오버', '연기', '조차', '가볍다', '않다'],
 ['너', '무재', '밓었', '다그', '래서', '보다', '추천', '다'],
 ['교도소', '이야기', '구먼', '솔직하다', '재미', '없다', '평점', '조정'],
 ['사이',
  '몬페',
  '그',
  '익살스럽다',
  '연기',
  '돋보이다',
  '영화',
  '스파이더맨',
  '에서',
  '늙다',
  '보이다',
  '크다',
  '스틴던스트',
  '너무나도',
  '이쁘다',
  '보이다'],
```

```
['막',
 '걸음',
 '마',
 '떼다',
 '초등학교',
 '학년',
 '생인',
 '살다',
 '영화',
 'ㅋㅋㅋ',
 '별',
 '반개',
 '아깝다',
 '움'],
```

2. 데이터 전처리 : 정수 인코딩(Integer Encoding)

- 정수 인코딩을 통해 단어에 정수를 부여하여, 리뷰에서 나온 단어를 빈도 순으로 정렬한 단어 집합으로 만드는 과정을 거침

<OOV> 1
영화 2
너무 3
정말 4
진짜 5
으로 6
에서 7
평점 8
연기 9
최고 10
드라마 11
스토리 12
이다 13
보고 14
감동 15
하는 16
감독 17
그냥 18
하고 19
내용 20
까지 21

< 단어 집합 >

등장 빈도가 2번 이하인 희귀 단어의 수: 25474
단어 집합에서 희귀 단어의 비율: 54.77926155301809
전체 등장 빈도에서 희귀 단어 등장 빈도 비율: 1.9761734039734433

단어 집합에서 등장 빈도가 2번 이하인 단어의 수를 제외함

단어 집합의 크기 : 41,578개

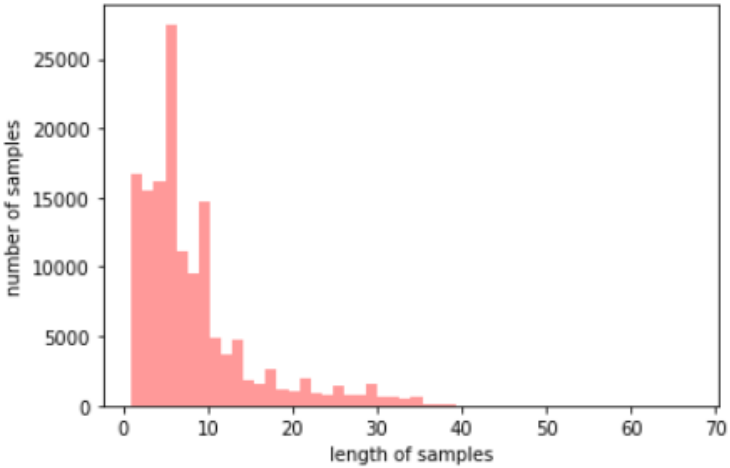
'모의': 41517, '바랫다': 41518, '토스트': 41519, '국내외': 41520, '겨누던': 41521, '메꿀': 41522, '해럴슨': 41523, '노무현정부': 41524, '하이든': 41525, '글레머': 41526, '졸잼꿀잼임': 41527, '리피아': 41528, '호그': 41529, '쓸을려': 41530, '통교': 41531, '수정역': 41532, '못생겄다': 41533, '절대영도': 41534, '우훗': 41535, '스왜': 41536, '한기대': 41537, '찔빛': 41538, '데루': 41539, '겻다': 41540, '엠플런스': 41541, '격해': 41542, '버스광고': 41543, '모꼬': 41544, '핫팩': 41545, '보슬영화': 41546, '성조기': 41547, '짓자': 41548, '호서': 41549, '쨔응태욱': 41550, '쥐기벨': 41551, '하묘': 41552, '비창': 41553, '밍쳥': 41554, '컾컦': 41555, '헬맷': 41556, '싸움개': 41557, '빅엿빅엿': 41558, '조르쥬': 41559, '내몬': 41560, '강비': 41561, '협심증': 41562, '역광': 41563, '비젼': 41564, '흉노족': 41565, '마루이': 41566, '머싰따능': 41567, '노형욱': 41568, '윤영삼': 41569, '초코': 41570, '대병소장': 41571, '신해혁명': 41572, '차이니즈': 41573, '차후': 41574, '섹귀들': 41575, '찎었': 41576, '디케이드': 41577, '수간': 41578}

< 41,578개의 단어 집합 >

2. 데이터 전처리 : 패딩(Padding)

- 패딩(Padding) 과정을 거쳐 리뷰들의 병렬 연산을 위해 리뷰들의 문장의 길이를 동일하게 맞춰줌

[영화 리뷰들의 문장 길이 분포 탐색]



리뷰의 최대 길이 : 67
리뷰의 평균 길이 : 8.26

전체 샘플 중 길이가
30 이하인 샘플의 비율 94%

30을 가장 긴 문장의 길이로 생각하고,
나머지 문장들을 0으로 채워줌

```
X_Train[:2]

array([[   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, 8940,   13,  154,  482],
       [   0,    0,    0,    0,    0,    0,    0,    0,    0,    0,    0,
          0,    0,    0,    0,    0,    0,    0,    0,    0,    0,  329,
         28,  439,    1, 1185,   20,  806,  496,   14]])
```

3. LSTM을 사용한 모델링과 학습

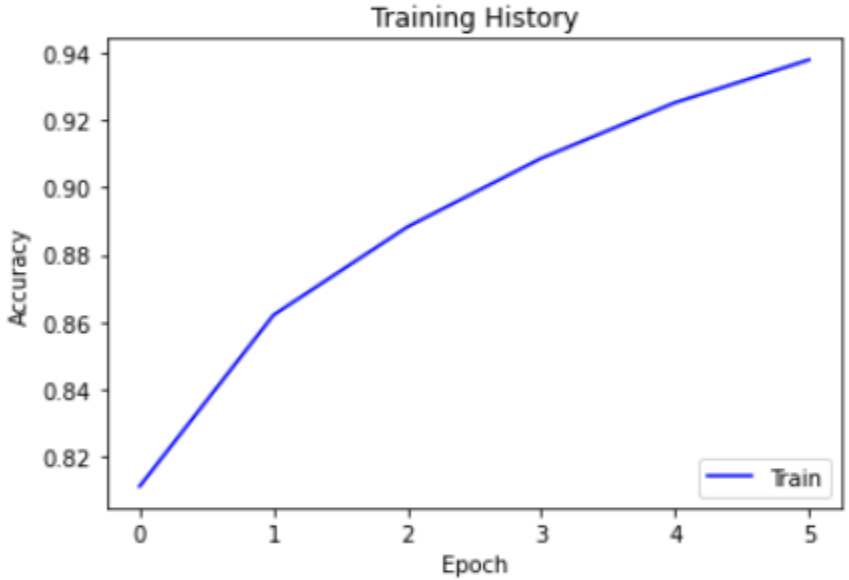
1) LSTM 모델 정의

```
Model: "sequential"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
embedding (Embedding)        (None, None, 100)         2351300
_________________________________________________________________
lstm (LSTM)                  (None, 128)               117248
_________________________________________________________________
dense (Dense)                (None, 1)                 129
=================================================================
Total params: 2,468,677
Trainable params: 2,468,677
Non-trainable params: 0
```

활성화함수 : sigmoid
최적화 : Adam
손실함수 : binary_crossentropy

3. LSTM을 사용한 모델링과 학습


```
1525/1525 [==============================] - 5s 3ms/step - loss: 0.3637 - acc: 0.8406
```



학습 히스토리

테스트 정확도 : 0.8406

4. 리뷰 감성 분석

꿀잼 ㅎㅎㅎ 강추! ㅋㅋㅋ

이런 망작이~~

들인 돈이 아깝다

이번 주말에 연인과 함께 시청 추천해요!

낮은 기대치 만큼 역시나 볼것 없는 영화였다.

```
sentiment_predict('꿀잼 ㅎㅎㅎ 강추! ㅋㅋㅋ')
```

99.45% 확률로 긍정 리뷰입니다.

```
sentiment_predict('이런 망작이~~')
```

99.42% 확률로 부정 리뷰입니다.

```
sentiment_predict('들인 돈이 아깝다')
```

98.73% 확률로 부정 리뷰입니다.

```
sentiment_predict('이번 주말에 연인과 함께 시청 추천해요!')
```

97.57% 확률로 긍정 리뷰입니다.

```
sentiment_predict('낮은 기대치 만큼 역시나 볼것 없는 영화였다.')
```

69.06% 확률로 부정 리뷰입니다.

4. 리뷰 감성 분석

우연치않게 봤다가 뒤통수맞음 신선함

후회없는 영화!

```
sentiment_predict('우연치않게 봤다가 뒤통수맞음 신선함')
```

71.30% 확률로 긍정 리뷰입니다.

```
sentiment_predict('후회없는 영화!')
```

54.80% 확률로 긍정 리뷰입니다.

4. 리뷰 감성 분석

여배우 굿, 남자배우도 굿, 조연도 굿, 근데 뭔가 조금 부족한 느낌은 뭘까…

뭔가 조금 부족한 느낌은 뭘까… 그래도 나름..

```
sentiment_predict('여배우 굿, 남자배우도 굿, 조연도 굿, 근데 뭔가 조금 부족한 느낌은 뭘까....')
```

64.50% 확률로 부정 리뷰입니다.

```
sentiment_predict('뭔가 조금 부족한 느낌은 뭘까...그래도 나름..')
```

91.88% 확률로 긍정 리뷰입니다.

둘 다 '부족한 느낌' 이 들어갔지만, 전체적인 문맥에 따라 긍정/부정 리뷰임을 잘 예측하고 있음

5. 영화 리뷰 워드 클라우드

네이버 영화 리뷰 워드클라우드

6. 프로젝트 후

- 리뷰의 전반적인 뉘앙스로는 긍/부정 판단이 어려움
- 많은 데이터를 활용하면 정확도가 높아지나 그만큼 전처리 및 학습 시간이 오래 걸림
- 부정 리뷰의 워드클라우드가 유의미하게 나오지 않았음