

Deep Learning for NLP Text Understanding

Prof. Joongheon Kim

https://sites.google.com/site/joongheonkim/joongheon@gmail.com

목표

- 목표
 - 데이터 분석의 기초 이해
 - 예제: IMDB 영화 리뷰 데이터 처리에 대한 기초에 대한 습득

```
import os
     import re
     import pandas as pd
     import tensorflow as tf
     from tensorflow.keras import utils
                                                                           IMDB데이터를 가져옴
     # IMDB 데이터다운로드

    □data set = tf.keras.utils.get file(
 9
             fname="imdb.tar.gz", #downloaded file name
             origin="http://ai.stanford.edu/~amaas/data/sentiment/aclImdb v1.tar.gz",
10
11
             extract=True)
12
13
    □def directory data(directory):
14
         data = \{\}
15
         data["review"] = []
16
         for file path in os.listdir(directory):
             with open(os.path.join(directory, file path), "r", encoding='utf-8') as file:
17
18
                  data["review"].append(file.read())
19
         return pd.DataFrame.from dict(data)
    □def data(directory):
         pos df = directory data(os.path.join(directory, "pos"))
         neg df = directory data(os.path.join(directory, "neg"))
23
         pos df["sentiment"] = 1
24
         neg df["sentiment"] = 0
26
         return pd.concat([pos df, neg df])
27
     train df = data(os.path.join(os.path.dirname(data set), "aclImdb", "train"))
     test df = data(os.path.join(os.path.dirname(data set), "aclImdb", "test"))
```

```
import os
                                                          모든 데이터가 디렉토리 안에 txt 파일 형태로 있어서
     import re
                                                          pandas의 데이터 프레임을 만들기 위해 변환작업을 진행해야 함
     import pandas as pd
                                                          → 아래의 두 개의 함수 필요
     import tensorflow as tf
     from tensorflow.keras import utils
     # IMDB 데이터다운로드

☐data set = tf.keras.utils.get file(
            fname="imdb.tar.gz", #downloaded file name
9
            origin="http://ai.stanford.edu/~amaas/data/sentiment/aclImdb v1.tar.gz",
11
            extract=True)
                                                                      각 파일에서 리뷰 텍스트를 불러오는 함수
12
13
   □def directory data(directory):
14
        data = \{\}
15
        data["review"] = []
16
        for file path in os.listdir(directory):
17
            with open (os.path.join (directory, file path), "r", encoding='utf-8') as file:
18
                data["review"].append(file.read()) [
                                                  • 데이터를 가져올 디렉토리를 인자로 받음
19
         return pd.DataFrame.from dict(data)
                                                  • 디렉토리 안에 있는 파일들을 하나씩 가져와 data["review"]에 하나씩 넣음
20
                                                  • pandas 데이터프레임으로 만들어서 변환함
   □def data(directory):
        pos df = directory data(os.path.join(directory, pos ))
        neg df = directory data(os.path.join(directory, "neg"))
23
        pos df["sentiment"] = 1
24
        neg df["sentiment"] = 0
26
        return pd.concat([pos df, neg df])
27
     train df = data(os.path.join(os.path.dirname(data set), "aclImdb", "train"))
     test df = data(os.path.join(os.path.dirname(data set), "aclImdb", "test"))
```

```
import os
                                                         모든 데이터가 디렉토리 안에 txt 파일 형태로 있어서
     import re
                                                         pandas의 데이터 프레임을 만들기 위해 변환작업을 진행해야 함
     import pandas as pd
                                                         → 아래의 두 개의 함수 필요
     import tensorflow as tf
     from tensorflow.keras import utils
     # IMDB 데이터다운로드

☐data set = tf.keras.utils.get file(
9
            fname="imdb.tar.gz", #downloaded file name
            origin="http://ai.stanford.edu/~amaas/data/sentiment/aclImdb v1.tar.gz",
11
            extract=True)
12
13
   □def directory data(directory):
14
        data = \{\}
15
        data["review"] = []
16
        for file path in os.listdir(directory):
17
            with open (os.path.join (directory, file path), "r", encoding='utf-8') as file:
18
                data["review"].append(file.read())
19
        return pd.DataFrame.from dict(data)
                                                                     각 리뷰에 해당하는 라벨값을 가져오는 함수
   □def data(directory):
21
        pos df = directory data(os.path.join(directory, "pos"))
23
        neg df = directory data(os.path.join(directory, "neg"))
24
        pos df["sentiment"] = 1
                                                • 폴더 이름을 지정하면 directory data함수를 호출하는데 이 때에
        neg df["sentiment"] = 0
                                                  pos폴더(긍정데이터)에 접근할지 neg폴더(부정데이터)에 접근할지를 통해
26
        return pd.concat([pos df, neg df])
                                                  각각의 데이터프레임을 얻음. 위 값은 각각 pos_df와 neg_df에 저장됨
27
     train df = data(os.path.join(os.path.dirname( .
                                                  라벨링 작업 → 긍정은 1, 부정은 0으로 만들고 데이터프레임을 통해 연동
     test df = data(os.path.join(os.path.dirname(data s
```

```
import os
     import re
     import pandas as pd
     import tensorflow as tf
     from tensorflow.keras import utils
     # IMDB 데이터다운로드
    □data set = tf.keras.utils.get file(
 9
             fname="imdb.tar.gz", #downloaded file name
             origin="http://ai.stanford.edu/~amaas/data/sentiment/aclImdb v1.tar.gz",
10
11
             extract=True)
12
13
    □def directory data(directory):
14
         data = \{\}
15
         data["review"] = []
16
         for file path in os.listdir(directory):
             with open (os.path.join (directory, file path), "r", encoding='utf-8') as file:
17
18
                 data["review"].append(file.read())
19
         return pd.DataFrame.from dict(data)
    ⊟def data(directory):
         pos df = directory data(os.path.join(directory, "pos"))
         neg df = directory data(os.path.join(directory, "neg"))
23
         pos df["sentiment"] = 1
24
         neg df["sentiment"] = 0
26
         return pd.concat([pos df, neg df])
                                                       앞에 설명한 두 함수를 사용하여 pandas 데이터프레임을 반환받는 구문
28
     train df = data(os.path.join(os.path.dirname(data set), "aclImdb", "train"))
     test df = data(os.path.join(os.path.dirname(data set), "aclImdb", "test"))
```

```
train_df.head()
reviews = list(train_df['review'])

# 문자열 문장 리스트를 토큰화
tokenized_reviews = [r.split() for r in reviews]
# 토큰화된 리스트에 대한 각 길이를 저장
review_len_by_token = [len(t) for t in tokenized_reviews]
# 토큰화된 것을 붙여서 음절의 길이를 저장
review_len_by_alphabet = [len(s.replace(' ', '')) for s in reviews]
```

```
review sentiment

The film begins with a bunch of kids in reform...

My favorite "Imperialism" movie and one of the...

Two great comedians in a great Neil Simon movi...

Just thought it was excellent and I still do...

Who can watch a movie, look at Lucy Liu and no...
```

```
train df.head()
     reviews = list(train df['review'])
33
                                                                  review 문장 리스트를 가져옴
34
     # 문자열 문장 리스트를 토큰화
                                                                  review에는 각 문장들을 리스트로 가지고 있음
35
     tokenized reviews = [r.split() for r in reviews]
36
     # 토큰화된 리스트에 대한 각 길이름 저장
37
     review len by token = [len(t) for t in tokenized reviews]
38
     # 토큰화된 것을 붙여서 유절의 길이를 저장
39
     review len by alphabet = [len(s.replace(' ', '')) for s in reviews]
```

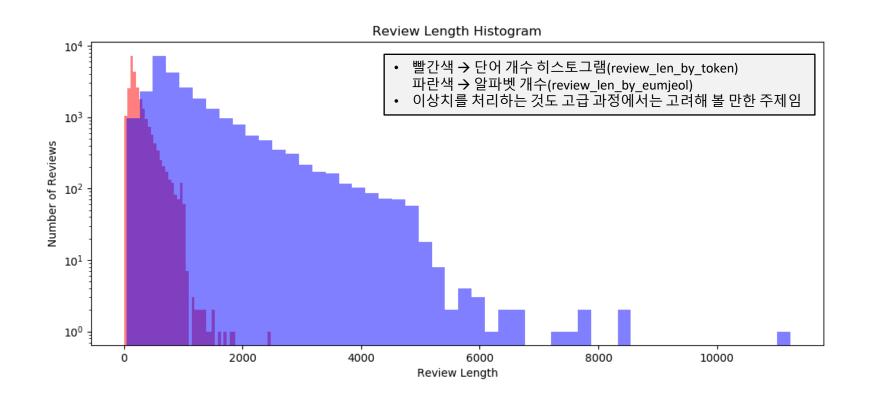
his movie as something as an outsider. That may be part of the reason for my disa with a little more camp value. As it is, My Name is Modesty is a deathly serious utstanding. As others have commented, she does appear a little too frail to be co ts. I never bought into the notion that this woman could handle a band of trained v Blaise character. I\'m convinced the concept has a lot of potential and I would cinema, young and old, was there to see talking animals make jokes, and whilst th y cared what happened to the tiger or whether Eddie Murphy made up with his daugh earch of a movie. The plot makes no sense, and the various characters drop in and have captured so accurately: that it\'s easy to make a cheap, low-quality film an f tomatoes that are floating near them; how far can "suspension of disbelief" go ective since I am a big fan the original 1944 movie. That, to me and many others, d trashed it, I didn\'t expect much, but you can\'t help but compare this with th red MacMurray, Barbara Stanwyck, Edward G. Robinson and others. Now I was seeing it was all over, I found it wasn\'t as bad as I had expected but it\'s no match f e two leads was missing and (2) being only 90 minutes, they rushed the story with atch for MacMurray and Stanwyck as "Walter Neff" and "Phyllis Dietrichson," respe Cobb was terrific as Keyes and Robert Webber as Norton, head of the insurance of etc., were all early \'70s instead of mid \'40s. Otherwise, the storyline was v the rest of my viewings of this classic story and film.', "Yikes. This is pretty doesn't decide how she wants to treat the material's theatrical origins (we get to keep reminding you that you're watching a film, whereas in fact it only serve tral performance is breath-takingly poor: stage-y and plummy, it's as if she's pl y be that her theatrical pedigree means that she is best able to handle the mater . Shaw's. Ben Kingsley turns in an average and disengaged turn, and Diana Rigg's d fe if this film is to be the evidence.". 'Evervthing about this film is hog wash.

```
train df.head()
    reviews = list(train df['review'])
                                                                                              review 문장의 파싱
33
34
     # 문자열 문장 리스트를 토큰화
35
     tokenized reviews = [r.split() for r in reviews]
     # 토큰화된 리스트에 대한 각 길이를 저장
36
37
     review len by token = [len(t) for t in tokenized reviews]
38
     # 토큰화된 것을 붙여서 음절의 길이를 저장
39
     review len by alphabet = [len(s.replace(' ', '')) for s in reviews]
```

- 문장에 포함된 단어와 알파벳의 개수에 대한 데이터 분석을 수월하게 하기 위함
- 데이터 분석을 위한 사전작업 완료 → 데이터 분석 실질적인 시작 가능

히스토그램으로 문장을 구성하는 단어의 개수와 알파벳의 개수를 표현함

```
import matplotlib.pyplot as plt
42
     # 이미지 사이즈 선언, figsize: (가로, 세로) 형태의 튜플로 입력
43
    plt.figure(figsize=(12, 5))
44
    # 히스토그램 선언
45
     # bins: 히스토그램 값들에 대한 버켓 범위
46
     # range: x축 값의 범위
     # alpha: 그래프 색상 투명도
48
     # color: 그래프 색상
49
     # label: 그래프에 대한 라벨
50
    plt.hist(review len by token, bins=50, alpha=0.5, color= 'r', label='word')
    plt.hist(review len by alphabet, bins=50, alpha=0.5, color='b', label='alphabet')
52
    plt.yscale('log', nonposy='clip')
53
    # 그래프 제목, x축 라벨, y축 라벨
54
    plt.title('Review Length Histogram')
55
    plt.xlabel('Review Length')
56
    plt.ylabel('Number of Reviews')
```



```
import numpy as np
58
59
     print('문장 최대길이: ', np.max(review len by token))
60
     print('문장 최소길이: ', np.min(review len by token))
61
     print('문장 평균길이: ', np.mean(review len by token))
     print('문장 길이 표준편차: ', np.std(review len by token))
62
     print('문장 중간길이: ', np.median(review len by token))
64
     # 사분위의 대한 경우는 0~100 스케일로 되어있음
65
     print('제 1 사분위 길이: ', np.percentile(review_len_by_token, 25))
66
     print('제 3 사분위 길이: ', np.percentile(review len by token, 75))
```



```
문장 최대길이: 2470
문장 최소길이: 10
문장 평균길이: 233.7872
문장 길이 표준편차: 173.72955740506563
문장 중간길이: 174.0
제 1 사분위 길이: 127.0
제 3 사분위 길이: 284.0
```

```
plt.figure(figsize=(12, 5))
69
    # 박스플롯 생성
70
     # 첫번째 파라메터: 여러 분포에 대한 데이터 리스트를 입력
     # labels: 입력한 데이터에 대한 라벨
    # showmeans: 평균값을 마크함
   □plt.boxplot([review len_by_token],
74
                 labels=['token'],
75
                 showmeans=True)
                                                             문장 내 단어 수에 대한 히스토그램
    plt.figure(figsize=(12, 5))
   □plt.boxplot([review len by alphabet],
79
                 labels=['alphabet'],
                 showmeans=True)
                                             2500
                                                                         0
                                                                                    박스 안 녹색 삼각형: 평균값
                                                                                    박스 최상위치: 제3사 분위
                                             2000
                                                                                    박스 최하위치: 제1사 분위
                                                                       馬の西
                                                                                    박스 안 주황색 줄: 중간값
                                             1500
                                             1000
                                                                            제3사 분위로 부터 3/2 지점
                                             500
                                                                             token
```

```
plt.figure(figsize=(12, 5))
69
     # 박스플롯 생성
     # 첫번째 파라메터: 여러 분포에 대한 데이터 리스트를 입력
     # labels: 입력한 데이터에 대한 라벨
     # showmeans: 평균값을 마크함
    □plt.boxplot([review len_by_token],
74
                  labels=['token'],
75
                  showmeans=True)
76
     plt.figure(figsize=(12, 5))
                                                    문장 내 알파벳 수에 대한 히스토그램
    □plt.boxplot([review len by alphabet],
79
                  labels=['alphabet'],
80
                  showmeans=True)
                                                                               0
                                                10000
                                                 8000
                                                 6000
                                                 4000
                                                 2000
                                                                             alphabet
```

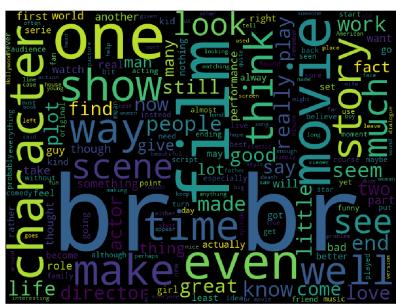
```
from wordcloud import WordCloud, STOPWORDS #
import matplotlib.pyplot as plt

wordcloud = WordCloud(stopwords = STOPWORDS, background_color = 'black', width = 800, height = 600).generate(' '.join(train_df['review']))

plt.figure(figsize = (15, 10))
plt.imshow(wordcloud)
plt.axis("off")
plt.show()

first world anothersiven bid other started
```

Word Cloud를 통한 주요 단어 시각화 (br이 크게 보이는 이유는
br>등의 HTML 태그 때문)



Word Cloud Installation) pip --trusted-host pypi.org --trusted-host files.pythonhosted.org install wordcloud

```
92
     import seaborn as sns
                                                  긍정/부정의 분포 확인
93
     import matplotlib.pyplot as plt
94
     sentiment = train df['sentiment'].value_counts()
95
96
     fig, axe = plt.subplots(ncols=1)
97
     fig.set size inches(6, 3)
98
     sns.countplot(train df['sentiment'])
99
     plt.show()
```

