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

예제 코드(Part 1)

```python
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(
        fname="imdb.tar.gz", #downloaded file name
        origin="http://ai.stanford.edu/~amaas/data/sentiment/aclImdb_v1.tar.gz",
        extract=True)

def directory_data(directory):
    data = {}
    data["review"] = []
    for file_path in os.listdir(directory):
        with open(os.path.join(directory, file_path), "r", encoding='utf-8') as file:
            data["review"].append(file.read())
    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"))
    pos_df["sentiment"] = 1
    neg_df["sentiment"] = 0
    return pd.concat([pos_df, neg_df])

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"))
```

IMDB데이터를 가져옴

예제 코드(Part 1)

```python
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(
        fname="imdb.tar.gz", #downloaded file name
        origin="http://ai.stanford.edu/~amaas/data/sentiment/aclImdb_v1.tar.gz",
        extract=True)

def directory_data(directory):
    data = {}
    data["review"] = []
    for file_path in os.listdir(directory):
        with open(os.path.join(directory, file_path), "r", encoding='utf-8') as file:
            data["review"].append(file.read())
    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"))
    pos_df["sentiment"] = 1
    neg_df["sentiment"] = 0
    return pd.concat([pos_df, neg_df])

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"))
```

모든 데이터가 디렉토리 안에 txt 파일 형태로 있어서 pandas의 데이터 프레임을 만들기 위해 변환작업을 진행해야 함
→ 아래의 두 개의 함수 필요

각 파일에서 리뷰 텍스트를 불러오는 함수

- 데이터를 가져올 디렉토리를 인자로 받음
- 디렉토리 안에 있는 파일들을 하나씩 가져와 data[“review”]에 하나씩 넣음
- pandas 데이터프레임으로 만들어서 변환함

예제 코드(Part 1)

```
1  import os
2  import re
3  import pandas as pd
4  import tensorflow as tf
5  from tensorflow.keras import utils
6
7  # IMDB 데이터다운로드
8  data_set = tf.keras.utils.get_file(
9          fname="imdb.tar.gz", #downloaded file name
10         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)
20
21 def data(directory):
22     pos_df = directory_data(os.path.join(directory, "pos"))
23     neg_df = directory_data(os.path.join(directory, "neg"))
24     pos_df["sentiment"] = 1
25     neg_df["sentiment"] = 0
26     return pd.concat([pos_df, neg_df])
27
28 train_df = data(os.path.join(os.path.dirname(
29 test_df = data(os.path.join(os.path.dirname(data_set), "aclImdb", "test"))
```

모든 데이터가 디렉토리 안에 txt 파일 형태로 있어서 pandas의 데이터 프레임을 만들기 위해 변환작업을 진행해야 함
→ 아래의 두 개의 함수 필요

각 리뷰에 해당하는 라벨값을 가져오는 함수

- 폴더 이름을 지정하면 directory_data함수를 호출하는데 이 때에 pos폴더(긍정데이터)에 접근할지 neg폴더(부정데이터)에 접근할지를 통해 각각의 데이터프레임을 얻음. 위 값은 각각 pos_df와 neg_df에 저장됨
- 라벨링 작업 → 긍정은 1, 부정은 0으로 만들고 데이터프레임을 통해 연동

예제 코드(Part 1)

```
1  import os
2  import re
3  import pandas as pd
4  import tensorflow as tf
5  from tensorflow.keras import utils
6
7  # IMDB 데이터다운로드
8  data_set = tf.keras.utils.get_file(
9          fname="imdb.tar.gz", #downloaded file name
10         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)
20
21 def data(directory):
22     pos_df = directory_data(os.path.join(directory, "pos"))
23     neg_df = directory_data(os.path.join(directory, "neg"))
24     pos_df["sentiment"] = 1
25     neg_df["sentiment"] = 0
26     return pd.concat([pos_df, neg_df])
27
28 train_df = data(os.path.join(os.path.dirname(data_set), "aclImdb", "train"))
29 test_df = data(os.path.join(os.path.dirname(data_set), "aclImdb", "test"))
```

앞에 설명한 두 함수를 사용하여 pandas 데이터프레임을 반환받는 구문

예제 코드(Part 2)

```
31  train_df.head()
32  reviews = list(train_df['review'])
33
34  # 문자열 문장 리스트를 토큰화
35  tokenized_reviews = [r.split() for r in reviews]
36  # 토큰화된 리스트에 대한 각 길이를 저장
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]
```

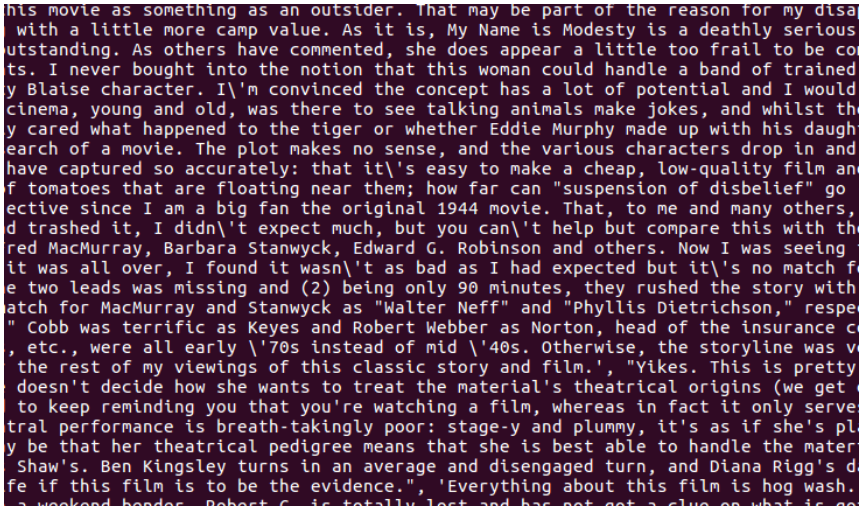
train_df에 review와 sentiment가 잘 들어와 있음을 확인

	review	sentiment
0	The film begins with a bunch of kids in reform...	1
1	My favorite "Imperialism" movie and one of the...	1
2	Two great comedians in a great Neil Simon movi...	1
3	I just thought it was excellent and I still do...	1
4	Who can watch a movie, look at Lucy Liu and no...	1

예제 코드(Part 2)

```
31  train_df.head()
32  reviews = list(train_df['review'])
33
34  # 문자열 문장 리스트를 토큰화
35  tokenized_reviews = [r.split() for r in reviews]
36  # 토큰화된 리스트에 대한 각 길이를 저장
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]
```

review 문장 리스트를 가져옴
review에는 각 문장들을 리스트로 가지고 있음



예제 코드(Part 2)

```
31  train_df.head()
32  reviews = list(train_df['review'])
33
34  # 문자열 문장 리스트를 토큰화
35  tokenized_reviews = [r.split() for r in reviews]
36  # 토큰화된 리스트에 대한 각 길이를 저장
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]
```

review 문장의 파싱

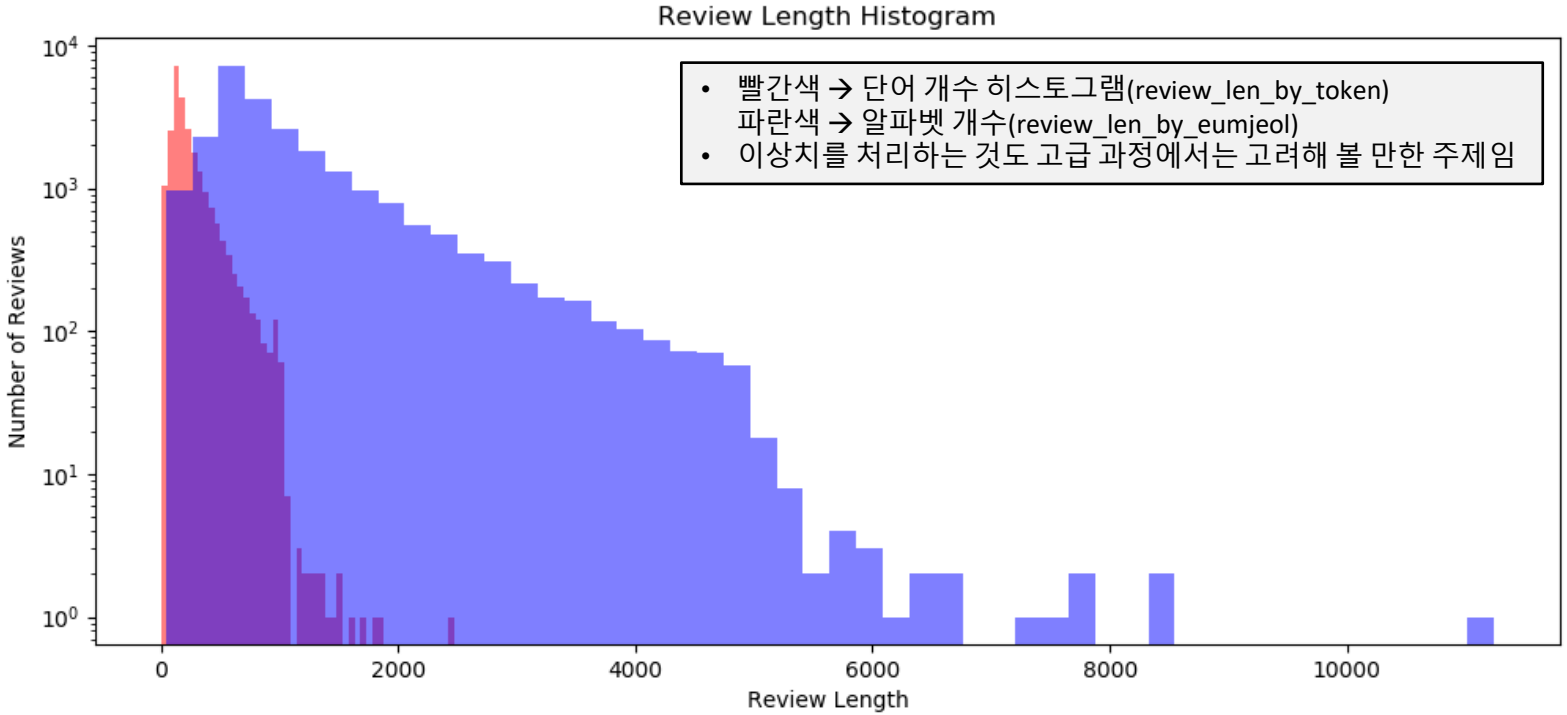
- 문장에 포함된 단어와 알파벳의 개수에 대한 데이터 분석을 수월하게 하기 위함
- 데이터 분석을 위한 사전작업 완료 → 데이터 분석 실질적인 시작 가능

예제 코드(Part 3)

히스토그램으로 문장을 구성하는 단어의 개수와 알파벳의 개수를 표현함

```
41  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
47  # 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')
51  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')
```

예제 코드(Part 3)


Review Length Histogram
Number of Reviews
Review Length
10^0
10^1
10^2
10^3
10^4
0
2000
4000
6000
8000
10000
• 빨간색 → 단어 개수 히스토그램(review_len_by_token)
파란색 → 알파벳 개수(review_len_by_eumjeol)
• 이상치를 처리하는 것도 고급 과정에서는 고려해 볼 만한 주제임

예제 코드(Part 4)

```
58  import numpy as np
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))
62  print('문장 길이 표준편차: ', np.std(review_len_by_token))
63  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))
```

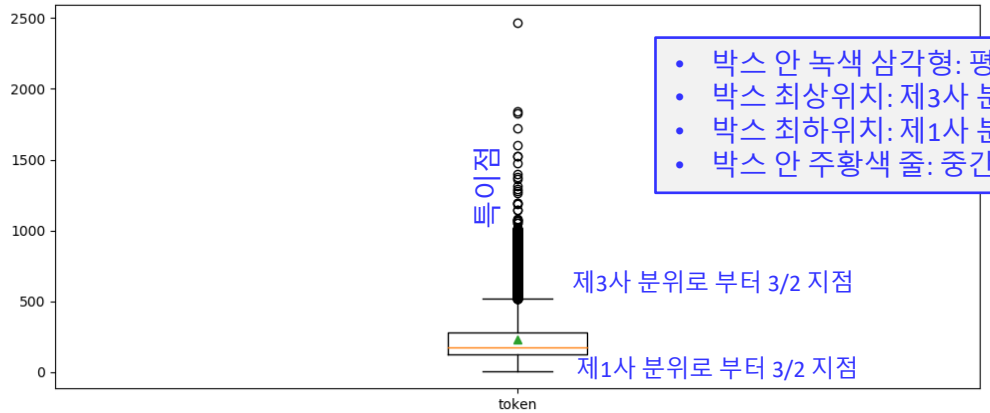


```
문장 최대길이:  2470
문장 최소길이:  10
문장 평균길이:  233.7872
문장 길이 표준편차:  173.72955740506563
문장 중간길이:  174.0
제 1 사분위 길이:  127.0
제 3 사분위 길이:  284.0
```

예제 코드(Part 5)

```
68  plt.figure(figsize=(12, 5))
69  # 박스플롯 생성
70  # 첫번째 파라메터: 여러 분포에 대한 데이터 리스트를 입력
71  # labels: 입력한 데이터에 대한 라벨
72  # showmeans: 평균값을 마크함
73  plt.boxplot([review_len_by_token],
74              labels=['token'],
75              showmeans=True)
76
77  plt.figure(figsize=(12, 5))
78  plt.boxplot([review_len_by_alphabet],
79              labels=['alphabet'],
80              showmeans=True)
```

문장 내 단어 수에 대한 히스토그램

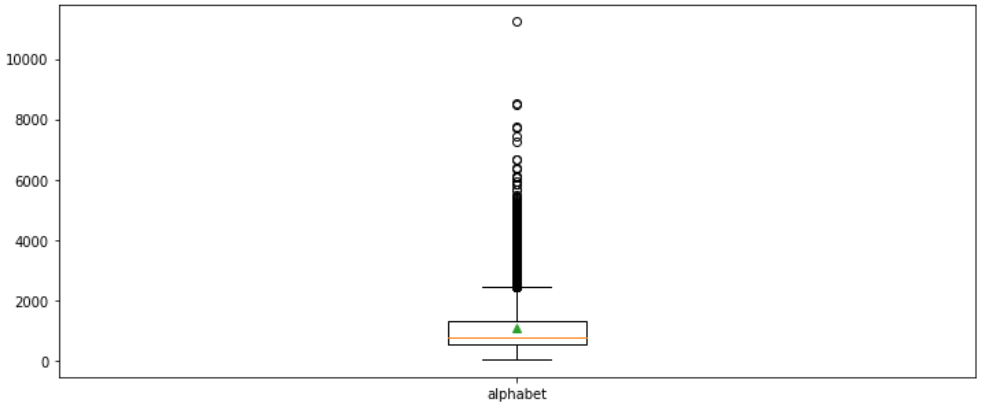


- 박스 안 녹색 삼각형: 평균값
- 박스 최상위치: 제3사 분위
- 박스 최하위치: 제1사 분위
- 박스 안 주황색 줄: 중간값

예제 코드(Part 5)

```
68  plt.figure(figsize=(12, 5))
69  # 박스플롯 생성
70  # 첫번째 파라메터: 여러 분포에 대한 데이터 리스트를 입력
71  # labels: 입력한 데이터에 대한 라벨
72  # showmeans: 평균값을 마크함
73  plt.boxplot([review_len_by_token],
74              labels=['token'],
75              showmeans=True)
76
77  plt.figure(figsize=(12, 5))
78  plt.boxplot([review_len_by_alphabet],
79              labels=['alphabet'],
80              showmeans=True)
```

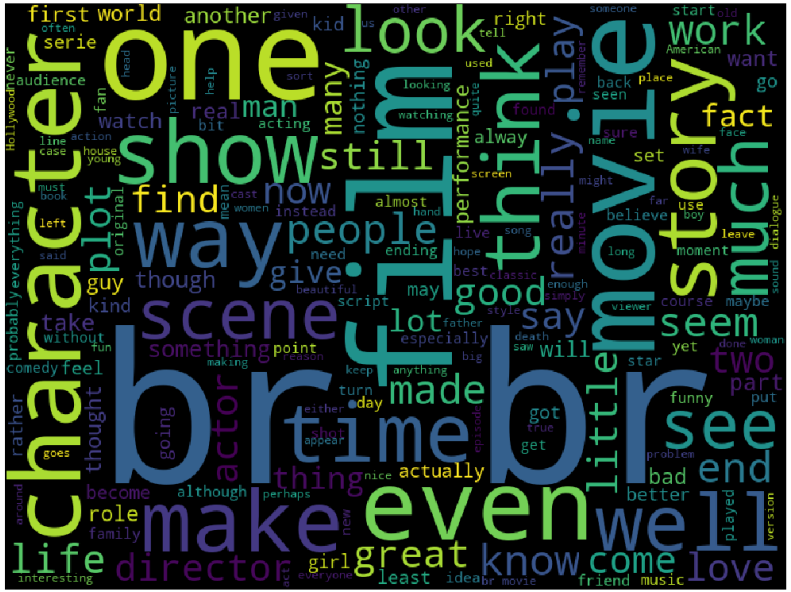
문장 내 알파벳 수에 대한 히스토그램



예제 코드(Part 6)

```
82  from wordcloud import WordCloud, STOPWORDS #
83  import matplotlib.pyplot as plt
84
85  wordcloud = WordCloud(stopwords = STOPWORDS, background_color = 'black', width = 800, height = 600).generate(' '.join(train_df['review']))
86
87  plt.figure(figsize = (15, 10))
88  plt.imshow(wordcloud)
89  plt.axis("off")
90  plt.show()
```

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



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

예제 코드(Part 7)

```
92  import seaborn as sns
93  import matplotlib.pyplot as plt
94
95  sentiment = train_df['sentiment'].value_counts()
96  fig, axe = plt.subplots(ncols=1)
97  fig.set_size_inches(6, 3)
98  sns.countplot(train_df['sentiment'])
99  plt.show()
```

긍정/부정의 분포 확인