```
import pandas as pd
import numpy as np
import urllib.request
from sklearn.feature_extraction.text import CountVectorizer
from sklearn.feature_extraction.text import TfidfVectorizer
import nltk
from nltk.corpus import stopwords
from nltk.stem import WordNetLemmatizer # 접사 (affix) ; 단어의 추가적의 의미를 주는 부분
```

```
# 접사 (affix) ; 단어의 추가적의 의미를 주는 부분
# 어간 (stem) ; 단어의 의미를 담고 있는 단어의 핵심 부분
# cats ; cat(어간) -s(접사)
# fox ;
```

```
nltk.download('wordnet')
```

```
[nltk_data] Downloading package wordnet to /root/nltk_data...
[nltk_data]   Unzipping corpora/wordnet.zip.
True
```

```
lemmatizer = WordNetLemmatizer()
words = ['policy', 'doing', 'organization', 'have', 'going', 'love', 'lives', 'f
print([lemmatizer.lemmatize(w) for w in words])
```

```
['policy', 'doing', 'organization', 'have', 'going', 'love', 'life', 'fly', 'c
```

```
lemmatizer.lemmatize('dies', 'v')
```

```
'die'
```

```
lemmatizer.lemmatize('watched', 'v')
```

```
'watch'
```

```
lemmatizer.lemmatize('has', 'v')
```

```
'have'
```

```
nltk.download('punkt')
nltk.download('wordnet')
nltk.download('stopwords')
```

```
[nltk_data] Downloading package punkt to /root/nltk_data...
[nltk_data]   Unzipping tokenizers/punkt.zip.
[nltk_data] Downloading package wordnet to /root/nltk_data...
[nltk_data]   Package wordnet is already up-to-date!
[nltk_data] Downloading package stopwords to /root/nltk_data...
[nltk_data]   Unzipping corpora/stopwords.zip.
True
```

```
urllib.request.urlretrieve("https://raw.githubusercontent.com/franciscadias/data
                           filename="/content/abcnews-data-text.csv")
```

```
('/content/abcnews-data-text.csv', <http.client.HTTPMessage at 0x7ff38b291fd0>
```

```
data = pd.read_csv('/content/abcnews-data-text.csv', error_bad_lines=False)
data
```

	publish_date	headline_text
0	20030219	aba decides against community broadcasting lic...
1	20030219	act fire witnesses must be aware of defamation
2	20030219	a g calls for infrastructure protection summit
3	20030219	air nz staff in aust strike for pay rise
4	20030219	air nz strike to affect australian travellers
...	...	...
1082163	20170630	when is it ok to compliment a womans smile a g...
1082164	20170630	white house defends trumps tweet
1082165	20170630	winter closes in on tasmania as snow ice falls
1082166	20170630	womens world cup australia wins despite atapat...
1082167	20170630	youtube stunt death foreshadowed by tweet

```
data.head() # 상위 5개 출력
```

	publish_date	headline_text
0	20030219	aba decides against community broadcasting lic...
1	20030219	act fire witnesses must be aware of defamation
2	20030219	a g calls for infrastructure protection summit
3	20030219	air nz staff in aust strike for pay rise
4	20030219	air nz strike to affect australian travellers

```
data.tail() # 하위 5개 출력
```

	publish_date	headline_text
1082163	20170630	when is it ok to compliment a womans smile a g...

```
text = data[['headline_text']]
```

1082165	20170630	winter closes in on tasmania as snow ice falls

```
text.nunique()
```

```
headline_text    1054983
dtype: int64
```

```
text.drop_duplicates(inplace=True) #중복된 요소를 제거
text = text.reset_index(drop=True)
print(len(text))
```

```
1054983
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:1: SettingWithCop
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/st
  """Entry point for launching an IPython kernel.
```

데이터 정제 및 정규화

```
text['headline_text'] = text.apply(lambda row:nltk.word_tokenize(row['headline_t
```

```
stop_words = stopwords.words('english')
text['headline_text'] = text['headline_text'].apply(lambda x : [word for word in
```

```
text.head()
```

	headline_text
0	[aba, decides, community, broadcasting, licence]
1	[act, fire, witnesses, must, aware, defamation]
2	[g, calls, infrastructure, protection, summit]
3	[air, nz, staff, aust, strike, pay, rise]
4	[air, nz, strike, affect, australian, travellers]

```
text['headline_text'] = text['headline_text'].apply(lambda x: [WordNetLemmatizer
```

```
text = text['headline_text'].apply(lambda x: [word for word in x if len(word)>2]
```

```
print(text[:5])
```

```
0    [aba, decide, community, broadcast, licence]
```

```
1    [act, fire, witness, must, aware, defamation]
2       [call, infrastructure, protection, summit]
3             [air, staff, aust, strike, pay, rise]
4    [air, strike, affect, australian, travellers]
Name: headline_text, dtype: object
```

```
detokenized_doc = []
for i in range(len(text)):
    t = ' '.join(text[i])
    detokenized_doc.append(t)

train_data = detokenized_doc
```

```
train_data[:5]
```

```
['aba decide community broadcast licence',
 'act fire witness must aware defamation',
 'call infrastructure protection summit',
 'air staff aust strike pay rise',
 'air strike affect australian travellers']
```

```
# DTM
c_vectorizer = CountVectorizer(stop_words='english', max_features = 5000)
document_term_matrix = c_vectorizer.fit_transform(train_data)
```

```
print('행렬의 크기 : ', document_term_matrix.shape)
```

```
행렬의 크기 :  (1054983, 5000)
```

```
tfidf_vectorizer = TfidfVectorizer(stop_words='english', max_features= 5000)
tf_idf_matrix = tfidf_vectorizer.fit_transform(train_data)
```

```
print('행렬의 크기 :', tf_idf_matrix.shape)
```

```
행렬의 크기 : (1054983, 5000)
```

여러가지 머신모델로 학습해보기

- KNN
- 나이브베이즈
- 베르누이 나이브 베이즈
- 랜덤포레스트
- SVM
- XGB (boosting기법)
- 결정트리

```
data[['publish_date']]
```

	publish_date
0	20030219
1	20030219
2	20030219
3	20030219
4	20030219
...	...
1082163	20170630
1082164	20170630
1082165	20170630
1082166	20170630
1082167	20170630

```
from sklearn.neighbors import KNeighborsClassifier
from sklearn.naive_bayes import MultinomialNB
from sklearn.naive_bayes import BernoulliNB
from sklearn.ensemble import RandomForestClassifier
from sklearn.svm import SVC
import xgboost as xgb
from xgboost.sklearn import XGBClassifier

models = []

```

abc 뉴스데이터로 word2vec

```
from nltk.corpus import abc
import nltk
nltk.download('abc')
nltk.download('punkt')
```

```
[nltk_data] Downloading package abc to /root/nltk_data...
[nltk_data]   Unzipping corpora/abc.zip.
[nltk_data] Downloading package punkt to /root/nltk_data...
[nltk_data]   Unzipping tokenizers/punkt.zip.
True
```

```
corpus = abc.sents()
```

```
print(corpus[:3])
```

```
[['PM', 'denies', 'knowledge', 'of', 'AWB', 'kickbacks', 'The', 'Prime', 'Mini
```

```
print('코퍼스의 크기 :', len(corpus))
```

```
코퍼스의 크기 : 29059
```

```
from gensim.models import Word2Vec

model = Word2Vec(sentences= corpus, size = 100, window=5, min_count=5, workers=4
```

```
model_result = model.wv.most_similar("man")
```

```
print(model_result)
```

```
[('woman', 0.9340240955352783), ('Bang', 0.9253141283988953), ('asteroid', 0.9
```

```
from gensim.models import KeyedVectors

model.wv.save_word2vec_format('./w2v')
loaded_model = KeyedVectors.load_word2vec_format("./w2v")
print("모델 load완료!")
```

```
모델 load완료!
```

```
model_result = loaded_model.wv.most_similar("man")
```

```
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:1: DeprecationWar
  """Entry point for launching an IPython kernel.
```

```
print(model_result)
```

```
[('woman', 0.9340240955352783), ('Bang', 0.9253141283988953), ('asteroid', 0.9
```

```
loaded_model.most_similar('overacting')
```

```
-------------------------------------------------------------------
```

```
1 loaded_model.most_similar('memory')
```

```
[('chasing', 0.9709212779998779),
 ('jolt', 0.9707803726196289),
 ('video', 0.9701134562492371),
 ('lifting', 0.9692025184631348),
 ('structures', 0.9688969254493713),
 ('display', 0.9687300324440002),
 ('infection', 0.968694806098938),
 ('semen', 0.9679538011550903),
 ('movie', 0.9677622318267822),
 ('shock', 0.9669978618621826)]
--> 452              raise KeyError("word '%s' not in vocabulary" %
```

```
1 loaded_model.most_similar('memorry')
```

```
-------------------------------------------------------------------
--------
KeyError                                  Traceback (most recent
call last)
<ipython-input-14-913270eb055a> in <module>()
----> 1 loaded_model.most_similar('memorry')

 ——————————————— 1 frames ———————————————
/usr/local/lib/python3.7/dist-
packages/gensim/models/keyedvectors.py in word_vec(self, word,
use_norm)
    450              return result
    451          else:
--> 452              raise KeyError("word '%s' not in vocabulary" %
word)
    453
```

한국어 word2vec 만들기

```
1 !pip install konlpy
```

```
Collecting konlpy
  Downloading konlpy-0.5.2-py2.py3-none-any.whl (19.4 MB)
     |████████████████████████████████| 19.4 MB 4.9 MB/s
Requirement already satisfied: numpy>=1.6 in /usr/local/lib/python3.7/dist-pac
Requirement already satisfied: lxml>=4.1.0 in /usr/local/lib/python3.7/dist-pa
Requirement already satisfied: tweepy>=3.7.0 in /usr/local/lib/python3.7/dist-
Collecting beautifulsoup4==4.6.0
  Downloading beautifulsoup4-4.6.0-py3-none-any.whl (86 kB)
     |████████████████████████████████| 86 kB 4.1 MB/s
Collecting JPype1>=0.7.0
  Downloading JPype1-1.3.0-cp37-cp37m-manylinux_2_5_x86_64.manylinux1_x86_64.w
     |████████████████████████████████| 448 kB 71.9 MB/s
Collecting colorama
  Downloading colorama-0.4.4-py2.py3-none-any.whl (16 kB)
Requirement already satisfied: typing-extensions in /usr/local/lib/python3.7/c
Requirement already satisfied: requests-oauthlib>=0.7.0 in /usr/local/lib/pyth
Requirement already satisfied: requests[socks]>=2.11.1 in /usr/local/lib/pytho
Requirement already satisfied: six>=1.10.0 in /usr/local/lib/python3.7/dist-pa
Requirement already satisfied: oauthlib>=3.0.0 in /usr/local/lib/python3.7/dis
Requirement already satisfied: certifi>=2017.4.17 in /usr/local/lib/python3.7/
Requirement already satisfied: urllib3!=1.25.0,!=1.25.1,<1.26,>=1.21.1 in /usr
Requirement already satisfied: idna<3,>=2.5 in /usr/local/lib/python3.7/dist-p
Requirement already satisfied: chardet<4,>=3.0.2 in /usr/local/lib/python3.7/c
Requirement already satisfied: PySocks!=1.5.7,>=1.5.6 in /usr/local/lib/pythor
Installing collected packages: JPype1, colorama, beautifulsoup4, konlpy
  Attempting uninstall: beautifulsoup4
    Found existing installation: beautifulsoup4 4.6.3
    Uninstalling beautifulsoup4-4.6.3:
      Successfully uninstalled beautifulsoup4-4.6.3
Successfully installed JPype1-1.3.0 beautifulsoup4-4.6.0 colorama-0.4.4 konlpy
```

```python
import pandas as pd
import matplotlib.pyplot as plt
import urllib.request
from gensim.models.word2vec import Word2Vec
from konlpy.tag import Okt
```

```python
urllib.request.urlretrieve("https://raw.githubusercontent.com/e9t/nsmc/master/ra
```

```
('ratings.txt', <http.client.HTTPMessage at 0x7f45fb09a450>)
```

```python
train_data = pd.read_table('ratings.txt')
```

```python
train_data[:5]
```

	id	document	label
0	8112052	어릴때보고 지금다시봐도 재밌어요ㅋㅋ	1
1	8132799	디자인을 배우는 학생으로, 외국디자이너와 그들이 일군 전통을 통해 발전해가는 문화산...	1
2	4655635	폴리스스토리 시리즈는 1부터 뉴까지 버릴께 하나도 없음.. 최고.	1
3	9251303	와.. 연기가 진짜 개쩔구나.. 지루할거라고 생각했는데 몰입해서 봤다.. 그래 이런	1

```python
print(len(train_data)) #리뷰 갯수 출력
```

```
200000
```

```python
# Null값 존재 유무
print(train_data.isnull().values.any())
```

```
True
```

```python
train_data = train_data.dropna(how='any') #null값이 존재하는 행 제거
print(train_data.isnull().values.any()) #Null값이 존재하는지 확인!
```

```
False
```

```
print(len(train_data))
```

```
199992
```

```
# 정규 표현식을 통한 한글 외 문자 제거
train_data['document'] = train_data['document'].str.replace("[^ㄱ-ㅎㅏ-ㅣ가-힣 ]",
```

```
train_data[:5]
```

	id	document	label
0	8112052	어릴때보고 지금다시봐도 재밌어요ㅋㅋ	1
1	8132799	디자인을 배우는 학생으로 외국디자이너와 그들이 일군 전통을 통해 발전해가는 문화산업...	1
2	4655635	폴리스스토리 시리즈는 부터 뉴까지 버릴께 하나도 없음 최고	1
3	9251303	와 연기가 진짜 개쩔구나 지루할거라고 생각했는데 몰입해서 봤다 그래 이런게 진짜 영화지	1

```
# 불용어 정의
stopwords = ['의','가','이','은','들','는','좀','잘','걍','과','도','를','으로','자','에'
```

```
okt = Okt()
tokenized_data = []
for sentence in train_data['document']:
    temp_x = okt.morphs(sentence, stem=True) # 토큰화
    temp_x = [word for word in temp_x if not word in stopwords] #불용어 제거
    tokenized_data.append(temp_x)
```

```
# 리뷰 길이 분포 확인
print('리뷰의 최대 길이 :', max(len(l) for l in tokenized_data))
print('리뷰의 평균 길이 :', sum(map(len, tokenized_data))/len(tokenized_data))
plt.hist([len(s) for s in tokenized_data], bins=50)
plt.xlabel('length of samples')
plt.ylabel('number of samples')
plt.show()
```

```
리뷰의 최대 길이 : 72
리뷰의 평균 길이 : 10.716703668146726
```

```
from gensim.models import Word2Vec
model = Word2Vec(sentences=tokenized_data, size=100, window=5, min_count=5, work
```

```
model.wv.vectors.shape
```

```
(16477, 100)
```

```
print(model.wv.most_similar("최민식"))
```

```
[('한석규', 0.8754636645317078), ('안성기', 0.8663073778152466), ('이민호', 0.86094
```

```
print(model.wv.most_similar("히어로"))
```

```
[('호러', 0.8675199747085571), ('느와르', 0.8628140687942505), ('슬래셔', 0.854195
```

사전 훈련된 워드 임베딩 (한국어)

```
import gensim
model = gensim.models.Word2Vec.load('/content/drive/MyDrive/Colab Notebooks/영우4
```

```
result = model.wv.most_similar("강아지")
print(result)
```

```
[('고양이', 0.7290452718734741), ('거위', 0.7185635566711426), ('토끼', 0.7056223
```

사전 훈련된 워드 임베딩 (영어)

```
import gensim

model = gensim.models.KeyedVectors.load_word2vec_format('/content/drive/MyDrive/
```

```
print(model.vectors.shape) # 3백만개의 단어와 각단어차원이 300
```

```
(3000000, 300)
```

```
print(model.similarity('this', 'is'))
print(model.similarity('post','book'))
```

```
0.40797037
0.057204384
```

```
print(model['book'])
```

```
[ 0.11279297 -0.02612305 -0.04492188  0.06982422  0.140625    0.03039551
 -0.04370117  0.24511719  0.08740234 -0.05053711  0.23144531 -0.07470703
  0.21875     0.03466797 -0.14550781  0.05761719  0.00671387 -0.00701904
  0.13183594 -0.25390625  0.14355469 -0.140625   -0.03564453 -0.21289062
 -0.24804688  0.04980469 -0.09082031  0.14453125  0.05712891 -0.10400391
 -0.19628906 -0.20507812 -0.27539062  0.03063965  0.20117188  0.17382812
  0.09130859 -0.10107422  0.22851562 -0.04077148  0.02709961 -0.00106049
  0.02709961  0.34179688 -0.13183594 -0.078125    0.02197266 -0.18847656
 -0.17480469 -0.05566406 -0.20898438  0.04858398 -0.07617188 -0.15625
 -0.05419922  0.01672363 -0.02722168 -0.11132812 -0.03588867 -0.18359375
  0.28710938  0.01757812  0.02185059 -0.05664062 -0.01251221  0.01708984
 -0.21777344 -0.06787109  0.04711914 -0.00668335  0.08544922 -0.02209473
  0.31835938  0.01794434 -0.02246094 -0.03051758 -0.09570312  0.24414062
  0.20507812  0.05419922  0.29101562  0.03637695  0.04956055 -0.06689453
  0.09277344 -0.10595703 -0.04370117  0.19726562 -0.03015137  0.05615234
  0.08544922 -0.09863281 -0.02392578 -0.08691406 -0.22460938 -0.16894531
  0.09521484 -0.0612793  -0.03015137 -0.265625   -0.13378906  0.00139618
  0.01794434  0.10107422  0.13964844  0.06445312 -0.09765625 -0.11376953
 -0.24511719 -0.15722656  0.00457764  0.12988281 -0.03540039 -0.08105469
  0.18652344  0.03125    -0.09326172 -0.04760742  0.23730469  0.11083984
  0.08691406  0.01916504  0.21386719 -0.0065918  -0.08984375 -0.02502441
 -0.09863281 -0.05639648 -0.26757812  0.19335938 -0.08886719 -0.25976562
  0.05957031 -0.10742188  0.09863281  0.1484375   0.04101562  0.00340271
 -0.06591797 -0.02941895  0.20019531 -0.00521851  0.02355957 -0.13671875
 -0.12597656 -0.10791016  0.0067749   0.15917969  0.0145874  -0.15136719
  0.07519531 -0.02905273  0.01843262  0.20800781  0.25195312 -0.11523438
 -0.23535156  0.04101562 -0.11035156  0.02905273  0.22460938 -0.04272461
  0.09667969  0.11865234  0.08007812  0.07958984  0.3125     -0.14941406
 -0.234375    0.06079102  0.06982422 -0.14355469 -0.05834961 -0.36914062
 -0.10595703  0.00738525  0.24023438 -0.10400391 -0.02124023  0.05712891
 -0.11621094 -0.16894531 -0.06396484 -0.12060547  0.08105469 -0.13769531
 -0.08447266  0.12792969 -0.15429688  0.17871094  0.2421875  -0.06884766
  0.03320312  0.04394531 -0.04589844  0.03686523 -0.07421875 -0.01635742
 -0.24121094 -0.08203125 -0.01733398  0.0291748   0.10742188  0.11279297
  0.12890625  0.01416016 -0.28710938  0.16503906 -0.25585938  0.2109375
 -0.19238281  0.22363281  0.04541016  0.00872803  0.11376953  0.375
  0.09765625  0.06201172  0.12109375 -0.24316406  0.203125    0.12158203
  0.08642578  0.01782227  0.17382812  0.01855469  0.03613281 -0.02124023
 -0.02905273 -0.04541016  0.1796875   0.06494141 -0.13378906 -0.09228516
  0.02172852  0.02099609  0.07226562  0.3046875  -0.27539062 -0.30078125
  0.08691406 -0.22949219  0.0546875  -0.34179688 -0.00680542 -0.0291748
 -0.03222656  0.16210938  0.01141357  0.23339844 -0.0859375  -0.06494141
  0.15039062  0.17675781  0.08251953 -0.26757812 -0.11669922  0.01330566
  0.01818848  0.10009766 -0.09570312  0.109375   -0.16992188 -0.23046875
 -0.22070312  0.0625      0.03662109 -0.125       0.05151367 -0.18847656
  0.22949219  0.26367188 -0.09814453  0.06176758  0.11669922  0.23046875
  0.32617188  0.02038574 -0.03735352 -0.12255859  0.296875   -0.25
 -0.08544922 -0.03149414  0.38085938  0.02929688 -0.265625    0.42382812
 -0.1484375   0.14355469 -0.03125     0.00717163 -0.16601562 -0.15820312
  0.03637695 -0.16796875 -0.01483154  0.09667969 -0.05761719 -0.00515747]
```

wikipedia word2Vec

```
!pip install wikiextractor
```

```
Collecting wikiextractor
  Downloading wikiextractor-3.0.6-py3-none-any.whl (46 kB)
     |████████████████████████████████| 46 kB 2.7 MB/s
Installing collected packages: wikiextractor
Successfully installed wikiextractor-3.0.6
```

```
# Colab에 Mecab 설치
!git clone https://github.com/SOMJANG/Mecab-ko-for-Google-Colab.git
%cd Mecab-ko-for-Google-Colab
!bash install_mecab-ko_on_colab190912.sh
```

```
Resolving bbuseruploads.s3.amazonaws.com (bbuseruploads.s3.amazonaws.com)... 5
Connecting to bbuseruploads.s3.amazonaws.com (bbuseruploads.s3.amazonaws.com)|
HTTP request sent, awaiting response... 200 OK
Length: 1414979 (1.3M) [application/x-tar]
Saving to: ‘mecab-0.996-ko-0.9.2.tar.gz.1’

mecab-0.996-ko-0.9. 100%[===================>]   1.35M  --.-KB/s    in 0.08s

2021-12-02 07:30:31 (16.4 MB/s) - ‘mecab-0.996-ko-0.9.2.tar.gz.1’ saved [14149

Done
Unpacking mecab-0.996-ko-0.9.2.tar.gz.......
Done
Change Directory to mecab-0.996-ko-0.9.2.......
installing mecab-0.996-ko-0.9.2.tar.gz........
configure
make
make check
make install
ldconfig
Done
Change Directory to /content
Downloading mecab-ko-dic-2.1.1-20180720.tar.gz.......
from https://bitbucket.org/eunjeon/mecab-ko-dic/downloads/mecab-ko-dic-2.1.1-2
--2021-12-02 07:30:49--  https://bitbucket.org/eunjeon/mecab-ko-dic/downloads/
Resolving bitbucket.org (bitbucket.org)... 104.192.141.1, 2406:da00:ff00::3403
Connecting to bitbucket.org (bitbucket.org)|104.192.141.1|:443... connected.
HTTP request sent, awaiting response... 302 Found
Location: https://bbuseruploads.s3.amazonaws.com/a4fcd83e-34f1-454e-a6ac-c242c
--2021-12-02 07:30:49--  https://bbuseruploads.s3.amazonaws.com/a4fcd83e-34f1-
Resolving bbuseruploads.s3.amazonaws.com (bbuseruploads.s3.amazonaws.com)... 5
Connecting to bbuseruploads.s3.amazonaws.com (bbuseruploads.s3.amazonaws.com)|
HTTP request sent, awaiting response... 200 OK
Length: 49775061 (47M) [application/x-tar]
Saving to: ‘mecab-ko-dic-2.1.1-20180720.tar.gz.1’

mecab-ko-dic-2.1.1- 100%[===================>]  47.47M  96.5MB/s    in 0.5s

2021-12-02 07:30:50 (96.5 MB/s) - ‘mecab-ko-dic-2.1.1-20180720.tar.gz.1’ saved

Done
Unpacking  mecab-ko-dic-2.1.1-20180720.tar.gz.......
Done
Change Directory to mecab-ko-dic-2.1.1-20180720
Done
installing........
```

```
configure
make
make install
apt-get update
apt-get upgrade
apt install curl
apt install git
bash <(curl -s https://raw.githubusercontent.com/konlpy/konlpy/master/scripts/
Done
Successfully Installed
Now you can use Mecab
from konlpy.tag import Mecab
```

```
!wget https://dumps.wikimedia.org/kowiki/latest/kowiki-latest-pages-articles.xml
```

```
--2021-12-02 06:42:12--  https://dumps.wikimedia.org/kowiki/latest/kowiki-late
Resolving dumps.wikimedia.org (dumps.wikimedia.org)... 208.80.154.7, 2620:0:86
Connecting to dumps.wikimedia.org (dumps.wikimedia.org)|208.80.154.7|:443... c
HTTP request sent, awaiting response... 200 OK
Length: 802741769 (766M) [application/octet-stream]
Saving to: ‘kowiki-latest-pages-articles.xml.bz2’

kowiki-latest-pages 100%[===================>] 765.55M  3.00MB/s    in 4m 15s

2021-12-02 06:46:28 (3.00 MB/s) - ‘kowiki-latest-pages-articles.xml.bz2’ savec
```

```
!python -m wikiextractor.WikiExtractor kowiki-latest-pages-articles.xml.bz2
# 위키익스트랙터를 사용하여 위키피디아 덤프를 파싱한다.
```

```
INFO: Preprocessing 'kowiki-latest-pages-articles.xml.bz2' to collect template
INFO: Preprocessed 100000 pages
INFO: Preprocessed 200000 pages
INFO: Preprocessed 300000 pages
INFO: Preprocessed 400000 pages
INFO: Preprocessed 500000 pages
INFO: Preprocessed 600000 pages
INFO: Preprocessed 700000 pages
INFO: Preprocessed 800000 pages
INFO: Preprocessed 900000 pages
INFO: Preprocessed 1000000 pages
INFO: Preprocessed 1100000 pages
INFO: Preprocessed 1200000 pages
INFO: Preprocessed 1300000 pages
INFO: Preprocessed 1400000 pages
INFO: Preprocessed 1500000 pages
INFO: Preprocessed 1600000 pages
INFO: Loaded 57894 templates in 344.3s
INFO: Starting page extraction from kowiki-latest-pages-articles.xml.bz2.
INFO: Using 1 extract processes.
INFO: Extracted 100000 articles (724.9 art/s)
INFO: Extracted 200000 articles (1015.1 art/s)
INFO: Extracted 300000 articles (1142.4 art/s)
INFO: Extracted 400000 articles (1218.0 art/s)
INFO: Extracted 500000 articles (1267.9 art/s)
INFO: Extracted 600000 articles (1226.1 art/s)
INFO: Extracted 700000 articles (1311.1 art/s)
INFO: Extracted 800000 articles (1384.8 art/s)
```

```
INFO: Extracted 900000 articles (2424.2 art/s)
INFO: Extracted 1000000 articles (1963.5 art/s)
INFO: Extracted 1100000 articles (1373.1 art/s)
INFO: Extracted 1200000 articles (1424.1 art/s)
INFO: Finished 1-process extraction of 1257505 articles in 991.1s (1268.8 art/
```

```
ls
```

```
images/                                     LICENSE
install_mecab-ko_on_colab190912.sh          README.md
install_mecab-ko_on_colab_light_210108.sh   text/
kowiki-latest-pages-articles.xml.bz2
```

```
ls text/AA
```

```
wiki_00  wiki_12  wiki_24  wiki_36  wiki_48  wiki_60  wiki_72  wiki_84  wiki_9
wiki_01  wiki_13  wiki_25  wiki_37  wiki_49  wiki_61  wiki_73  wiki_85  wiki_9
wiki_02  wiki_14  wiki_26  wiki_38  wiki_50  wiki_62  wiki_74  wiki_86  wiki_9
wiki_03  wiki_15  wiki_27  wiki_39  wiki_51  wiki_63  wiki_75  wiki_87  wiki_9
wiki_04  wiki_16  wiki_28  wiki_40  wiki_52  wiki_64  wiki_76  wiki_88
wiki_05  wiki_17  wiki_29  wiki_41  wiki_53  wiki_65  wiki_77  wiki_89
wiki_06  wiki_18  wiki_30  wiki_42  wiki_54  wiki_66  wiki_78  wiki_90
wiki_07  wiki_19  wiki_31  wiki_43  wiki_55  wiki_67  wiki_79  wiki_91
wiki_08  wiki_20  wiki_32  wiki_44  wiki_56  wiki_68  wiki_80  wiki_92
wiki_09  wiki_21  wiki_33  wiki_45  wiki_57  wiki_69  wiki_81  wiki_93
wiki_10  wiki_22  wiki_34  wiki_46  wiki_58  wiki_70  wiki_82  wiki_94
wiki_11  wiki_23  wiki_35  wiki_47  wiki_59  wiki_71  wiki_83  wiki_95
```

```
import os
import re
```

```
os.listdir('text')
```

```
['AD', 'AG', 'AA', 'AC', 'AB', 'AH', 'AE', 'AI', 'AF']
```

```
def list_wiki(dirname):
    filepaths = []
    filenames = os.listdir(dirname)
    for filename in filenames:
        filepath = os.path.join(dirname, filename)

        if os.path.isdir(filepath):
            # 재귀 함수
            filepaths.extend(list_wiki(filepath))
        else:
            find = re.findall(r"wiki_[0-9][0-9]", filepath)
            if 0 < len(find):
                filepaths.append(filepath)
    return sorted(filepaths)
```

```
filepaths = list_wiki('text')
```

```
len(filepaths)
```

```
862
```

```
with open("output_file.txt", "w") as outfile:
    for filename in filepaths:
        with open(filename) as infile:
            contents = infile.read()
            outfile.write(contents)
```

```
f = open('output_file.txt', encoding="utf8")

i=0
while True:
    line = f.readline()
    if line != '\n':
        i = i+1
        print("%d번째 줄 :" %i+line)
    if i==10:
        break
f.close()
```

```
1번째 줄 :<doc id="5" url="https://ko.wikipedia.org/wiki?curid=5" title="지미 카터

2번째 줄 :지미 카터

3번째 줄 :제임스 얼 카터 주니어(, 1924년 10월 1일 ~ )는 민주당 출신 미국 39대 대통령 (1977년

4번째 줄 :생애.

5번째 줄 :어린 시절.

6번째 줄 :지미 카터는 조지아주 섬터 카운티 플레인스 마을에서 태어났다.

7번째 줄 :조지아 공과대학교를 졸업하였다. 그 후 해군에 들어가 전함·원자력·잠수함의 승무원으로 일하였다

8번째 줄 :정계 입문.

9번째 줄 :1962년 조지아 주 상원 의원 선거에서 낙선하나 그 선거가 부정선거 였음을 입증하게 되어 당선

10번째 줄 :대통령 재임.
```

형태소 분석

```
from tqdm import tqdm
from konlpy.tag import Mecab
```

```
mecab = Mecab()
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
/usr/local/lib/python3.7/dist-packages/konlpy/tag/_mecab.py in __init__(self, dicpath)
    107         try:
--> 108             self.tagger = Tagger('-d %s' % dicpath)
    109             self.tagset = utils.read_json('%s/data/tagset/mecab.json' % utils.installpath)

NameError: name 'Tagger' is not defined

During handling of the above exception, another exception occurred:

Exception                                 Traceback (most recent call last)
                    1 frames
/usr/local/lib/python3.7/dist-packages/konlpy/tag/_mecab.py in __init__(self, dicpath)
    111             raise Exception('The MeCab dictionary does not exist at "%s". Is the dictionary correctly installed?\nYou can also try entering the dictionary path when initializing the Mecab class:
```

```
#!cd /content/Mecab-ko-for-Google-Colab/text
!pwd
```

```
/content/Mecab-ko-for-Google-Colab
```

```
f = open('output_file.txt', encoding='utf8')
lines = f.read().splitlines()
print(len(lines))
```

```
9877278
```

```
lines[:10]
```

```
['<doc id="5" url="https://ko.wikipedia.org/wiki?curid=5" title="지미 카터">',
 '지미 카터',
 '',
 '제임스 얼 카터 주니어(, 1924년 10월 1일 ~ )는 민주당 출신 미국 39대 대통령 (1977년 ~ 1981
 '생애.',
 '어린 시절.',
 '지미 카터는 조지아주 섬터 카운티 플레인스 마을에서 태어났다.',
 '조지아 공과대학교를 졸업하였다. 그 후 해군에 들어가 전함·원자력·잠수함의 승무원으로 일하였다. 1953
 '정계 입문.',
 '1962년 조지아 주 상원 의원 선거에서 낙선하나 그 선거가 부정선거 였음을 입증하게 되어 당선되고, 19
```

```
# 빈 문자열은 제외하고 형태소 분석을 진행.
result = []

for line in tqdm(lines):
    # 빈 문자열이 아닌 경우에만 수행
    if line:
        result.append(okt.morphs(line))
```

```
  1%|          | 105704/9877278 [39:57<45:55:24, 59.11it/s]
```

```
len(result)
```

Word2Vec 학습

```
from gensim.models import Word2Vec
model = Word2Vec(result, size=100, window=5, min_count=5, workers=4, sg=0)
```

```
model_result1 = model.wv.most_similar("대한민국")
print(model_result1)
```

```
model_result2 = model.wv.most_similar("어벤져스")
print(model_result2)
```

```
model_result3 = model.wv.most_similar("반도체")
print(model_result3)
```