Universal Sentence Encoder

In [1]:
```
import numpy as np
from nltk.corpus import stopwords
import warnings
warnings.filterwarnings("ignore")
import pandas as pd
import sqlite3
import csv
import matplotlib.pyplot as plt
import seaborn as sns
import numpy as np
import re
import os
from sqlalchemy import create_engine # database connection
import datetime as dt
from nltk.corpus import stopwords
from nltk.tokenize import word_tokenize
from nltk.stem.snowball import SnowballStemmer
from sklearn.feature_extraction.text import CountVectorizer
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.multiclass import OneVsRestClassifier
from sklearn.linear_model import SGDClassifier
from sklearn import metrics
from sklearn.metrics import f1_score,precision_score,recall_score
from sklearn import svm
from sklearn.linear_model import LogisticRegression
from sklearn.naive_bayes import GaussianNB
from datetime import datetime

from tqdm import tqdm
from nltk.corpus import stopwords
from sklearn.metrics.pairwise import cosine_similarity
from sklearn.metrics import pairwise_distances
```

In [2]:
```
data_main_clean_v5=pd.read_pickle('data_main_clean_v5.pickle')
```

In [3]:
```
data_main_clean_v5.head()
```

Out[3]:

	Title	Tokens	Cleaned_Title	Title_Length	Token_Space	L
0	implementing boundary value analysis of softwa...	[c++, testing]	implementing boundary value analysis software ...	74	c++ testing	value prog tes implemen softw
1	java.lang.noclassdeffounderror: javax/servlet/...	[java, jsp]	java lang noclassdeffounderror javax servlet j...	76	java jsp	java jsp l ser
2	java.sql.sqlexception: [microsoft][odbc driver ...	[java, sql]	java sql sqlexception microsoft odbc driver ma...	79	java sql	java index inv micro mana dr
3	better way to update feed on fb with php sdk	[php]	better way update feed fb php sdk	44	php	php update better f
4	"sql injection" issue preventing correct form ...	[php, sql]	sql injection issue preventing correct form su...	62	php sql	- php form is cor

Using Google API to find Universal Sentence Encoder embeddings

In [28]:
```
###TAKES A LONG TIME DONT RUN

#import tensorflow_hub as hub
#module_url = "https://tfhub.dev/google/universal-sentence-encoder-large/3"
#embed = hub.KerasLayer(module_url)
```

```
###Find embeddings using API
#import numpy as np
#import tensorflow as tf
#np_list = np.asarray(data_main_clean_v5['Cleaned_Title'])
#tensor_list = tf.convert_to_tensor(np_list)
#a=list(tensor_list)
#embedding = embed(tensor_list)
#z=np.array(embedding)

###Save embeddings to box
#from numpy import savez_compressed
#savez_compressed('USE_EMBEDDINGS', z)
```

In [4]:
```
from numpy import load
dict_data = load('USE_EMBEDDINGS.npz')
a1 = dict_data['arr_0']
#a1 contains the embedding vectors
```

In [5]:
```
a1.shape
```

Out[5]:
```
(1000000, 512)
```

Universal sentence encoder + Cosine Distance

In [13]:
```
def Recomend(string):
    stopwords_1 = stopwords.words("english")
    a=string
    sent_1=a.lower().strip()
    sent_1 = re.sub(r"won\'t", "will not", sent_1)
    sent_1 = re.sub(r"can\'t", "can not", sent_1)
    sent_1 = re.sub(r"n\'t", " not", sent_1)
    sent_1 = re.sub(r"\'re", " are", sent_1)
    sent_1 = re.sub(r"\'s", " is", sent_1)
    sent_1 = re.sub(r"\'d", " would", sent_1)
    sent_1 = re.sub(r"\'ll", " will", sent_1)
    sent_1 = re.sub(r"\'t", " not", sent_1)
    sent_1 = re.sub(r"\'ve", " have", sent_1)
```

```
    sent_1 = re.sub(r"\'m", " am", sent_1)
    sent_1 = re.sub('[^A-Za-z0-9-+]+', ' ', sent_1)
    sent_1 = ' '.join(e for e in sent_1.split() if e not in stopwords_1
)
    sent_1=sent_1.lower().strip()
    print('QUERY ENTERED BY THE USER')
    print(sent_1)
    print('\n')

    query=a1[0]
    distance  = pairwise_distances(a1, query.reshape(1,-1),metric='cosi
ne')
    indices = np.argsort(distance.flatten())[0:10]
    pdists  = np.sort(distance.flatten())[0:10]

    print('RECOMENDED SIMILAR QUESTIONS')
    g=0
    for i in indices:
        g=g+1
        print(g ,'th question','"',data_main_clean_v5['Cleaned_Title'][
i],'"')
        print(g ,'th question distance is ',round((float(distance[i])),
4))
        print('\n')
```

In [22]:
```
import time
start_time = time.time()
Recomend('implementing boundary value analysis software testing c++ pro
gram')
print('TIME TAKEN TO FETCH RESULTS')
print('5.656825304031372','seconds')
```

```
QUERY ENTERED BY THE USER
implementing boundary value analysis software testing c++ program


RECOMENDED SIMILAR QUESTIONS
1 th question " implementing boundary value analysis software testing c
++ program "
1 th question distance is  0.0
```

```
1 th question distance is  0.0

2 th question " implementing data structures algorithms c++ "
2 th question distance is  0.1083

3 th question " boundary value analysis c++ cppunit "
3 th question distance is  0.1114

4 th question " code metrics analysis unmanaged c++ code "
4 th question distance is  0.1211

5 th question " c++ code profiling analysis mac mpi "
5 th question distance is  0.1262

6 th question " calculating critical path dag c++ "
6 th question distance is  0.1282

7 th question " methods implementing using graphs nodes c++ "
7 th question distance is  0.1319

8 th question " complex data structures embedding extending python c++
"
8 th question distance is  0.1345

9 th question " obfuscate c++ variables functions "
9 th question distance is  0.1375

10 th question " c++ static global non-pod theory practice "
10 th question distance is  0.1376
```

```
TIME TAKEN TO FETCH RESULTS
5.656825304031372 seconds
```

Universal sentence encoder + Euclidean Distance

In [30]:

```
def Recomend(string):
    stopwords_1 = stopwords.words("english")
    a=string
    sent_1=a.lower().strip()
    sent_1 = re.sub(r"won\'t", "will not", sent_1)
    sent_1 = re.sub(r"can\'t", "can not", sent_1)
    sent_1 = re.sub(r"n\'t", " not", sent_1)
    sent_1 = re.sub(r"\'re", " are", sent_1)
    sent_1 = re.sub(r"\'s", " is", sent_1)
    sent_1 = re.sub(r"\'d", " would", sent_1)
    sent_1 = re.sub(r"\'ll", " will", sent_1)
    sent_1 = re.sub(r"\'t", " not", sent_1)
    sent_1 = re.sub(r"\'ve", " have", sent_1)
    sent_1 = re.sub(r"\'m", " am", sent_1)
    sent_1 = re.sub('[^A-Za-z0-9-+]+', ' ', sent_1)
    sent_1 = ' '.join(e for e in sent_1.split() if e not in stopwords_1
)
    sent_1=sent_1.lower().strip()
    print('QUERY ENTERED BY THE USER')
    print(sent_1)
    print('\n')

    query=a1[0]
    distance  = pairwise_distances(a1, query.reshape(1,-1),metric='eucl
idean')
    indices = np.argsort(distance.flatten())[0:10]
    pdists  = np.sort(distance.flatten())[0:10]

    print('RECOMENDED SIMILAR QUESTIONS')
    g=0
    for i in indices:
        g=g+1
```

```
        print(g ,'th question','"',data_main_clean_v5['Cleaned_Title'][
i],'"')
        print(g ,'th question distance is ',round((float(distance[i])),
4))
        print('\n')
```

In [31]:
```
import time
start_time = time.time()
Recomend('implementing boundary value analysis software testing c++ pro
gram')
print('TIME TAKEN TO FETCH RESULTS')
print(time.time()-start_time,'seconds')
```

```
QUERY ENTERED BY THE USER
implementing boundary value analysis software testing c++ program


RECOMENDED SIMILAR QUESTIONS
1 th question " implementing boundary value analysis software testing c
++ program "
1 th question distance is  0.0


2 th question " implementing data structures algorithms c++ "
2 th question distance is  0.4653


3 th question " boundary value analysis c++ cppunit "
3 th question distance is  0.472


4 th question " code metrics analysis unmanaged c++ code "
4 th question distance is  0.4922


5 th question " c++ code profiling analysis mac mpi "
5 th question distance is  0.5023


6 th question " calculating critical path dag c++ "
```

```
6 th question distance is  0.5064


7 th question " methods implementing using graphs nodes c++ "
7 th question distance is  0.5136


8 th question " complex data structures embedding extending python c++
"
8 th question distance is  0.5187


9 th question " obfuscate c++ variables functions "
9 th question distance is  0.5244


10 th question " c++ static global non-pod theory practice "
10 th question distance is  0.5245


TIME TAKEN TO FETCH RESULTS
4.870607376098633 seconds
```

Queries from Applied AI + Cosine Distance

In [6]:
```
from numpy import load
dict_data = load('query.npz')
a2 = dict_data['arr_0']
#a1 contains the embedding vectors
```

In [7]:
```
a2.shape
```

Out[7]:
```
(2, 512)
```

In [19]:
```
def Recomend(string,i):
```

```
    stopwords_1 = stopwords.words("english")
    a=string
    sent_1=a.lower().strip()
    sent_1 = re.sub(r"won\'t", "will not", sent_1)
    sent_1 = re.sub(r"can\'t", "can not", sent_1)
    sent_1 = re.sub(r"n\'t", " not", sent_1)
    sent_1 = re.sub(r"\'re", " are", sent_1)
    sent_1 = re.sub(r"\'s", " is", sent_1)
    sent_1 = re.sub(r"\'d", " would", sent_1)
    sent_1 = re.sub(r"\'ll", " will", sent_1)
    sent_1 = re.sub(r"\'t", " not", sent_1)
    sent_1 = re.sub(r"\'ve", " have", sent_1)
    sent_1 = re.sub(r"\'m", " am", sent_1)
    sent_1 = re.sub('[^A-Za-z0-9-+]+', ' ', sent_1)
    sent_1 = ' '.join(e for e in sent_1.split() if e not in stopwords_1
)
    sent_1=sent_1.lower().strip()
    print('QUERY ENTERED BY THE USER')
    print(a)
    print('\n')

    query=a2[i]
    distance  = pairwise_distances(a1, query.reshape(1,-1),metric='cosi
ne')
    indices = np.argsort(distance.flatten())[0:10]
    pdists  = np.sort(distance.flatten())[0:10]

    print('RECOMENDED SIMILAR QUESTIONS')
    g=0
    for i in indices:
        g=g+1
        print(g ,'th question','"',data_main_clean_v5['Cleaned_Title'][
i],'"')
        print(g ,'th question distance is ',round((float(distance[i])),
4))
        print('\n')
```

In [16]:
```
Recomend('how to create a linked list in python',0)
```

```
QUERY ENTERED BY THE USER
how to create a linked list in python


RECOMENDED SIMILAR QUESTIONS
1 th question " creating python list list tuples "
1 th question distance is  0.0485


2 th question " create list tuples list python "
2 th question distance is  0.0499


3 th question " python create list specific indexes list lists "
3 th question distance is  0.0501


4 th question " extend list within list python "
4 th question distance is  0.0508


5 th question " make python sublists list using seperator "
5 th question distance is  0.0533


6 th question " creating list methods executed python "
6 th question distance is  0.0539


7 th question " python create new list based existing list without cert
ain objects "
7 th question distance is  0.054


8 th question " convert python multiple list list "
8 th question distance is  0.0545


9 th question " create multidimensional list python two lists "
```

```
9 th question distance is  0.0548


10 th question " python make new tuple attaching info existing list "
10 th question distance is  0.0564
```

In [20]:
```
Recomend('LSTM with Keras',1)
```

```
QUERY ENTERED BY THE USER
LSTM with Keras


RECOMENDED SIMILAR QUESTIONS
1 th question " 2d convolution python similar matlab conv2 "
1 th question distance is  0.1341


2 th question " interpolation morphing image labview opencv "
2 th question distance is  0.1426


3 th question " interpolation subsampling 3d data python without vtk "
3 th question distance is  0.144


4 th question " multiplying matrix vector glm opengl "
4 th question distance is  0.1441


5 th question " opencv python bindings grabcut algorithm "
5 th question distance is  0.1483


6 th question " bilinear interpolation pil image python "
6 th question distance is  0.1496
```

```
7 th question " easiest way perform modular matrix inversion python "
7 th question distance is  0.1498


8 th question " calculate affine motion model coefficients using opencv
c++ "
8 th question distance is  0.1502


9 th question " perform bilinear interpolation python "
9 th question distance is  0.1508


10 th question " creating contour opencv using python "
10 th question distance is  0.1511
```

Queries from Applied AI + Euclidean Distance

In [21]:
```
def Recomend(string,i):
    stopwords_1 = stopwords.words("english")
    a=string
    sent_1=a.lower().strip()
    sent_1 = re.sub(r"won\'t", "will not", sent_1)
    sent_1 = re.sub(r"can\'t", "can not", sent_1)
    sent_1 = re.sub(r"n\'t", " not", sent_1)
    sent_1 = re.sub(r"\'re", " are", sent_1)
    sent_1 = re.sub(r"\'s", " is", sent_1)
    sent_1 = re.sub(r"\'d", " would", sent_1)
    sent_1 = re.sub(r"\'ll", " will", sent_1)
    sent_1 = re.sub(r"\'t", " not", sent_1)
    sent_1 = re.sub(r"\'ve", " have", sent_1)
    sent_1 = re.sub(r"\'m", " am", sent_1)
    sent_1 = re.sub('[^A-Za-z0-9-+]+', ' ', sent_1)
    sent_1 = ' '.join(e for e in sent_1.split() if e not in stopwords_1
```

```
)
    sent_1=sent_1.lower().strip()
    print('QUERY ENTERED BY THE USER')
    print(a)
    print('\n')

    query=a2[i]
    distance  = pairwise_distances(a1, query.reshape(1,-1),metric='eucl
idean')
    indices = np.argsort(distance.flatten())[0:10]
    pdists  = np.sort(distance.flatten())[0:10]

    print('RECOMENDED SIMILAR QUESTIONS')
    g=0
    for i in indices:
        g=g+1
        print(g ,'th question','"',data_main_clean_v5['Cleaned_Title'][
i],'"')
        print(g ,'th question distance is ',round((float(distance[i])),
4))
        print('\n')
```

In [22]:
```
Recomend('how to create a linked list in python',0)
```

```
QUERY ENTERED BY THE USER
how to create a linked list in python


RECOMENDED SIMILAR QUESTIONS
1 th question " creating python list list tuples "
1 th question distance is  0.3115


2 th question " create list tuples list python "
2 th question distance is  0.3158


3 th question " python create list specific indexes list lists "
3 th question distance is  0.3167
```

```
4 th question " extend list within list python "
4 th question distance is  0.3187


5 th question " make python sublists list using seperator "
5 th question distance is  0.3265


6 th question " creating list methods executed python "
6 th question distance is  0.3284


7 th question " python create new list based existing list without cert
ain objects "
7 th question distance is  0.3287


8 th question " convert python multiple list list "
8 th question distance is  0.3301


9 th question " create multidimensional list python two lists "
9 th question distance is  0.331


10 th question " python make new tuple attaching info existing list "
10 th question distance is  0.3358
```

In [23]:
```
Recomend('LSTM with Keras',1)
```

```
QUERY ENTERED BY THE USER
LSTM with Keras


RECOMENDED SIMILAR QUESTIONS
```

```
1 th question " 2d convolution python similar matlab conv2 "
1 th question distance is  0.518


2 th question " interpolation morphing image labview opencv "
2 th question distance is  0.534


3 th question " interpolation subsampling 3d data python without vtk "
3 th question distance is  0.5367


4 th question " multiplying matrix vector glm opengl "
4 th question distance is  0.5369


5 th question " opencv python bindings grabcut algorithm "
5 th question distance is  0.5446


6 th question " bilinear interpolation pil image python "
6 th question distance is  0.5469


7 th question " easiest way perform modular matrix inversion python "
7 th question distance is  0.5474


8 th question " calculate affine motion model coefficients using opencv
c++ "
8 th question distance is  0.5481


9 th question " perform bilinear interpolation python "
9 th question distance is  0.5492


10 th question " creating contour opencv using python "
10 th question distance is  0.5497
```

In [ ]: