In [1]:
```
import pandas as pd
import matplotlib.pyplot as plt
import re
import time
import warnings
import numpy as np
from nltk.corpus import stopwords
from sklearn.preprocessing import normalize
from sklearn.feature_extraction.text import CountVectorizer
from sklearn.feature_extraction.text import TfidfVectorizer
warnings.filterwarnings("ignore")
import sys
import os
import pandas as pd
import numpy as np
from tqdm import tqdm
import sqlite3
# exctract word2vec vectors
# https://github.com/explosion/spaCy/issues/1721
# http://landinghub.visualstudio.com/visual-cpp-build-tools
import spacy
```

In [2]:
```
# avoid decoding problems
df = pd.read_csv("train.csv")

# encode questions to unicode
# https://stackoverflow.com/a/6812069
# ----------------- python 2 ---------------------
# df['question1'] = df['question1'].apply(lambda x: unicode(str(x),"utf-8"))
# df['question2'] = df['question2'].apply(lambda x: unicode(str(x),"utf-8"))
# ----------------- python 3 ---------------------
df['question1'] = df['question1'].apply(lambda x: str(x))
df['question2'] = df['question2'].apply(lambda x: str(x))
```

In [7]:
```
df.head()
```

Out[7]:

	id	qid1	qid2	question1	question2	is_duplicate	q1_feats_m	q2_feats_m
0	0	1	2	What is the step by step guide to invest in sh...	What is the step by step guide to invest in sh...	0	[121.92992722988129, 100.0839056968689, 72.497...	[125.98330116271973, 95.63648426532745, 42.114...
1	1	3	4	What is the story of Kohinoor (Koh-i-Noor) Dia...	What would happen if the Indian government sto...	0	[-78.07093501091003, 54.843786865472794, 82.73...	[-106.8718991279602, 80.29034039378166, 79.066...
2	2	5	6	How can I increase the speed of my internet co...	How can Internet speed be increased by hacking...	0	[-5.3550145626068115, 73.6718100309372, 14.376...	[7.072874799370766, 15.51337805390358, 1.84691...
3	3	7	8	Why am I mentally very lonely? How can I solve...	Find the remainder when [math]23^{24} [/math] i...	0	[5.778358697891235, -34.71203848719597, 48.999...	[39.421539425849915, 44.136989906430244, -24.0...
4	4	9	10	Which one dissolve in water quikly sugar, salt...	Which fish would survive in salt water?	0	[51.13821983337402, 38.58731163293123, 123.639...	[31.950109004974365, 62.854101717472076, 1.778...

In [4]:
```
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.feature_extraction.text import CountVectorizer
# merge texts
questions = list(df['question1']) + list(df['question2'])

tfidf = TfidfVectorizer(lowercase=False, )
tfidf.fit_transform(questions)

# dict key:word and value:tf-idf score
word2tfidf = dict(zip(tfidf.get_feature_names(), tfidf.idf_))
```

In [5]:
```
import spacy

# en_vectors_web_lg, which includes over 1 million unique vectors.
nlp = spacy.load('en_core_web_sm')

vecs1 = []
# https://github.com/noamraph/tqdm
# tqdm is used to print the progress bar
for qu1 in tqdm(list(df['question1'])):
    doc1 = nlp(qu1)
    # 384 is the number of dimensions of vectors
    mean_vec1 = np.zeros([len(doc1), 384])
    for word1 in doc1:
        # word2vec
        vec1 = word1.vector
        # fetch df score
        try:
            idf = word2tfidf[str(word1)]
        except:
            idf = 0
        # compute final vec
        mean_vec1 += vec1 * idf
    mean_vec1 = mean_vec1.mean(axis=0)
    vecs1.append(mean_vec1)
df['q1_feats_m'] = list(vecs1)
```

```
100%|██████████████████████████████████████████████████████████████████████| 4
04290/404290 [1:20:47<00:00, 83.40it/s]
```

In [6]:
```
vecs2 = []
for qu2 in tqdm(list(df['question2'])):
    doc2 = nlp(qu2)
    mean_vec2 = np.zeros([len(doc2), 384])
    for word2 in doc2:
        # word2vec
        vec2 = word2.vector
        # fetch df score
        try:
            idf = word2tfidf[str(word2)]
        except:
            #print word
            idf = 0
        # compute final vec
        mean_vec2 += vec2 * idf
    mean_vec2 = mean_vec2.mean(axis=0)
    vecs2.append(mean_vec2)
df['q2_feats_m'] = list(vecs2)
```

```
100%|██████████████████████████████████████████████████████████████████████| 4
04290/404290 [1:18:32<00:00, 85.78it/s]
```

In [8]:
```
#prepro_features_train.csv (Simple Preprocessing Feartures)
#nlp_features_train.csv (NLP Features)
if os.path.isfile('nlp_features_train.csv'):
    dfnlp = pd.read_csv("nlp_features_train.csv",encoding='latin-1')
else:
    print("download nlp_features_train.csv from drive or run previous notebook")

if os.path.isfile('df_fe_without_preprocessing_train.csv'):
    dfppro = pd.read_csv("df_fe_without_preprocessing_train.csv",encoding='latin-1'
)
else:
    print("download df_fe_without_preprocessing_train.csv from drive or run previou
s notebook")
```

In [9]:
```
df1 = dfnlp.drop(['qid1','qid2','question1','question2'],axis=1)
df2 = dfppro.drop(['qid1','qid2','question1','question2','is_duplicate'],axis=1)
df3 = df.drop(['qid1','qid2','question1','question2','is_duplicate'],axis=1)
df3_q1 = pd.DataFrame(df3.q1_feats_m.values.tolist(), index= df3.index)
df3_q2 = pd.DataFrame(df3.q2_feats_m.values.tolist(), index= df3.index)
```

In [10]:
```
# dataframe of nlp features
df1.head()
```

Out[10]:

	id	is_duplicate	cwc_min	cwc_max	csc_min	csc_max	ctc_min	ctc_max	last_word_eq	first_wo
0	0	0	0.999980	0.833319	0.999983	0.999983	0.916659	0.785709	0.0	1.0
1	1	0	0.799984	0.399996	0.749981	0.599988	0.699993	0.466664	0.0	1.0
2	2	0	0.399992	0.333328	0.399992	0.249997	0.399996	0.285712	0.0	1.0
3	3	0	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.0	0.0
4	4	0	0.399992	0.199998	0.999950	0.666644	0.571420	0.307690	0.0	1.0

In [11]:
```
# data before preprocessing
df2.head()
```

Out[11]:

	id	freq_qid1	freq_qid2	q1len	q2len	q1_n_words	q2_n_words	word_Common	word_Total	word_
0	0	1	1	66	57	14	12	10.0	23.0	0.434
1	1	4	1	51	88	8	13	4.0	20.0	0.200
2	2	1	1	73	59	14	10	4.0	24.0	0.166
3	3	1	1	50	65	11	9	0.0	19.0	0.000
4	4	3	1	76	39	13	7	2.0	20.0	0.100

In [12]:
```
# Questions 1 tfidf weighted word2vec
df3_q1.head()
```

Out[12]:

	0	1	2	3	4	5	6	7	
0	121.929927	100.083906	72.497900	115.641795	-48.370865	34.619070	-172.057790	-92.502626	1
1	-78.070935	54.843787	82.738495	98.191855	-51.234840	55.013509	-39.140733	-82.692374	4
2	-5.355015	73.671810	14.376365	104.130241	1.433537	35.229116	-148.519385	-97.124595	4
3	5.778359	-34.712038	48.999631	59.699204	40.661263	-41.658731	-36.808594	24.170655	(
4	51.138220	38.587312	123.639488	53.333041	-47.062739	37.356212	-298.722753	-106.421119	1

5 rows × 384 columns

In [13]:
```
# Questions 2 tfidf weighted word2vec
df3_q2.head()
```

Out[13]:

	0	1	2	3	4	5	6	7	
0	125.983301	95.636484	42.114717	95.449986	-37.386301	39.400084	-148.116068	-87.851481	11
1	-106.871899	80.290340	79.066300	59.302100	-42.175332	117.616657	-144.364242	-127.131506	22
2	7.072875	15.513378	1.846914	85.937583	-33.808811	94.702337	-122.256856	-114.009530	53
3	39.421539	44.136990	-24.010927	85.265864	-0.339028	-9.323141	-60.499653	-37.044767	49
4	31.950109	62.854102	1.778147	36.218763	-45.130861	66.674880	-106.342344	-22.901031	59

5 rows × 384 columns

In [14]:
```
print("Number of features in nlp dataframe :", df1.shape[1])
print("Number of features in preprocessed dataframe :", df2.shape[1])
print("Number of features in question1 w2v  dataframe :", df3_q1.shape[1])
print("Number of features in question2 w2v  dataframe :", df3_q2.shape[1])
print("Number of features in final dataframe  :", df1.shape[1]+df2.shape[1]+df3_q1.
shape[1]+df3_q2.shape[1])
```

```
Number of features in nlp dataframe : 17
Number of features in preprocessed dataframe : 12
Number of features in question1 w2v  dataframe : 384
Number of features in question2 w2v  dataframe : 384
Number of features in final dataframe  : 797
```

In [16]:
```
# storing the final features to csv file
if not os.path.isfile('final_features.csv'):
    df3_q1['id']=df1['id']
    df3_q2['id']=df1['id']
    df1  = df1.merge(df2, on='id',how='left')
    df2  = df3_q1.merge(df3_q2, on='id',how='left')
    result  = df1.merge(df2, on='id',how='left')
    result.to_csv('final_features.csv')
```