3.6 Featurizing text data with tfidf weighted word-vectors

In [0]:

```
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
from sklearn.model_selection import train_test_split

# exctract word2vec vectors
# https://github.com/explosion/spaCy/issues/1721
# http://landinghub.visualstudio.com/visual-cpp-build-tools
import spacy
```

In [0]:

```
# avoid decoding problems
df = pd.read_csv("quora_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 [0]:

```
df.head()
```

Out[0]:

	id	qid1	qid2	question1	question2	is_duplicate
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
1	1	3	4	What is the story of Kohinoor (Koh-i-Noor) Dia...	What would happen if the Indian government sto...	0
2	2	5	6	How can I increase the speed of my internet co...	How can Internet speed be increased by hacking...	0
3	3	7	8	Why am I mentally very lonely? How can I solve...	Find the remainder when [math]23^{24}[/math] i...	0
4	4	9	10	Which one dissolve in water quikly sugar, salt...	Which fish would survive in salt water?	0

In [0]:

```
#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 previous notebook")
```

In [0]:

```
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','is_duplicate'],axis=1)
```

In [0]:

```
# dataframe of nlp features
df1.head()
```

Out[0]:

	id	is_duplicate	cwc_min	cwc_max	csc_min	csc_max	ctc_min	ctc_max	last_word_eq	first_word_eq	abs_len_diff	mean_len
0	0	0	0.999980	0.833319	0.999983	0.999983	0.916659	0.785709	0.0	1.0	2.0	13.0
1	1	0	0.799984	0.399996	0.749981	0.599988	0.699993	0.466664	0.0	1.0	5.0	12.5
2	2	0	0.399992	0.333328	0.399992	0.249997	0.399996	0.285712	0.0	1.0	4.0	12.0
3	3	0	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.0	0.0	2.0	12.0
4	4	0	0.399992	0.199998	0.999950	0.666644	0.571420	0.307690	0.0	1.0	6.0	10.0

In [0]:

```
# data before preprocessing
df2.head()
```

Out[0]:

	id	freq_qid1	freq_qid2	q1len	q2len	q1_n_words	q2_n_words	word_Common	word_Total	word_share	freq_q1+q2	freq_q1-q2
0	0	1	1	66	57	14	12	10.0	23.0	0.434783	2	0
1	1	4	1	51	88	8	13	4.0	20.0	0.200000	5	3
2	2	1	1	73	59	14	10	4.0	24.0	0.166667	2	0
3	3	1	1	50	65	11	9	0.0	19.0	0.000000	2	0
4	4	3	1	76	39	13	7	2.0	20.0	0.100000	4	2

In [0]:

```
df3.head()
```

Out[0]:

	id	question1	question2
0	0	What is the step by step guide to invest in sh...	What is the step by step guide to invest in sh...
1	1	What is the story of Kohinoor (Koh-i-Noor) Dia...	What would happen if the Indian government sto...
2	2	How can I increase the speed of my internet co...	How can Internet speed be increased by hacking...
3	3	Why am I mentally very lonely? How can I solve...	Find the remainder when [math]23^{24}[/math] i...
4	4	Which one dissolve in water quikly sugar, salt...	Which fish would survive in salt water?

In [0]:

```
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 questions dataframe :", df3.shape[1])
print("Number of features in final dataframe  :", df1.shape[1]+df2.shape[1]+df3.shape[1])
```

```
Number of features in nlp dataframe : 17
Number of features in preprocessed dataframe : 12
Number of features in questions dataframe : 3
Number of features in final dataframe  : 32
```

In [0]:

```
# storing the final features to csv file
if not os.path.isfile('final_data.csv'):
    df3['id']=df1['id']
    df1  = df1.merge(df2, on='id',how='left')
    result  = df1.merge(df3, on='id',how='left')
    result.to_csv('final_data.csv')
```

In [0]:

```
df=pd.read_csv('final_data.csv')
df.head
```

Out[0]:

```
<bound method NDFrame.head of         Unnamed: 0      id  is_duplicate   cwc_min   cwc_max  
csc_min  \
0                 0       0             0  0.999980  0.833319  0.999983
1                 1       1             0  0.799984  0.399996  0.749981
2                 2       2             0  0.399992  0.333328  0.399992
3                 3       3             0  0.000000  0.000000  0.000000
4                 4       4             0  0.399992  0.199998  0.999950
5                 5       5             1  0.666656  0.571420  0.888879
6                 6       6             0  0.000000  0.000000  0.000000
7                 7       7             1  0.499975  0.499975  0.599988
8                 8       8             0  0.999950  0.499988  0.999975
9                 9       9             0  0.599988  0.499992  0.333322
10               10      10             0  0.000000  0.000000  0.499975
11               11      11             1  0.666644  0.499988  0.599988
12               12      12             1  0.999975  0.999975  0.666644
13               13      13             1  0.999967  0.749981  0.999967
14               14      14             0  0.909083  0.909083  0.999991
15               15      15             1  0.374995  0.299997  0.166664
16               16      16             1  0.499975  0.499975  0.999950
17               17      17             0  0.000000  0.000000  0.249994
18               18      18             1  0.571420  0.499994  0.199996
19               19      19             0  0.599988  0.599988  0.999975
20               20      20             1  0.666644  0.499988  0.499975
21               21      21             0  0.666644  0.666644  0.499988
22               22      22             0  0.666644  0.666644  0.499988
23               23      23             0  0.000000  0.000000  0.000000
24               24      24             0  0.166664  0.166664  0.000000
25               25      25             0  0.833319  0.714276  0.999988
26               26      26             0  0.999950  0.666644  0.999950
27               27      27             0  0.666644  0.399992  0.000000
28               28      28             0  0.799984  0.799984  0.999967
29               29      29             1  0.749981  0.428565  0.749981
...             ...     ...           ...       ...       ...       ...
404260       404260  404260             0  0.666644  0.499988  0.333322
404261       404261  404261             1  0.799984  0.799984  0.999980
404262       404262  404262             0  0.666644  0.666644  0.749981
404263       404263  404263             0  0.499992  0.499992  0.399992
404264       404264  404264             0  0.599988  0.333330  0.199996
404265       404265  404265             1  0.666644  0.666644  0.249994
404266       404266  404266             0  0.499992  0.157894  0.799984
404267       404267  404267             1  0.999975  0.666656  0.749981
404268       404268  404268             0  0.399992  0.333328  0.199996
404269       404269  404269             0  0.749981  0.599988  0.000000
404270       404270  404270             0  0.999900  0.999900  0.999980
404271       404271  404271             0  0.333328  0.285710  0.166664
404272       404272  404272             1  0.285710  0.285710  0.874989
404273       404273  404273             1  0.799984  0.399996  0.599988
404274       404274  404274             1  0.499988  0.399992  0.599988
404275       404275  404275             0  0.199996  0.199996  0.599988
404276       404276  404276             0  0.499975  0.499975  0.999950
404277       404277  404277             0  0.000000  0.000000  0.499975
404278       404278  404278             0  0.499988  0.399992  0.999975
404279       404279  404279             0  0.333328  0.333328  0.571420
```

```
404280      404280  404280             1  0.999980  0.833319  0.999980
404281      404281  404281             1  0.666656  0.666656  0.999983
404282      404282  404282             1  0.499988  0.499988  0.499975
404283      404283  404283             0  0.571420  0.499994  0.499988
404284      404284  404284             1  0.999967  0.749981  0.999967
404285      404285  404285             0  0.857131  0.857131  0.999980
404286      404286  404286             1  0.666644  0.666644  0.599988
404287      404287  404287             0  0.999900  0.499975  0.999950
404288      404288  404288             0  0.000000  0.000000  0.124998
404289      404289  404289             0  0.999967  0.999967  0.999980

         csc_max   ctc_min   ctc_max  last_word_eq  ...  q2len  q1_n_words  \
0       0.999983  0.916659  0.785709           0.0  ...     57          14   
1       0.599988  0.699993  0.466664           0.0  ...     88           8   
2       0.249997  0.399996  0.285712           0.0  ...     59          14   
3       0.000000  0.000000  0.000000           0.0  ...     65          11   
4       0.666644  0.571420  0.307690           0.0  ...     39          13   
5       0.799992  0.705878  0.705878           1.0  ...     90          16   
6       0.000000  0.000000  0.000000           0.0  ...     62           4   
7       0.428565  0.571420  0.444440           1.0  ...     41           7   
8       0.799984  0.857131  0.749991           0.0  ...     37           8   
9       0.249994  0.444440  0.444440           0.0  ...     49           9   
10      0.111110  0.111110  0.052631           0.0  ...    116           9   
11      0.599988  0.624992  0.555549           1.0  ...     38           9   
12      0.499988  0.857131  0.749991           1.0  ...     39           7   
13      0.999967  0.999983  0.857131           0.0  ...     38           7   
14      0.999991  0.724135  0.724135           0.0  ...    140          29   
15      0.142855  0.249998  0.235293           0.0  ...     91          15   
16      0.999950  0.749981  0.749981           0.0  ...     29           4   
17      0.142855  0.124998  0.083333           0.0  ...     40          12   
18      0.166664  0.384612  0.357140           1.0  ...     73          14   
19      0.999975  0.777769  0.777769           0.0  ...     54           9   
20      0.333322  0.599988  0.428565           1.0  ...     43           5   
21      0.285710  0.571420  0.399996           0.0  ...     48           6   
22      0.333328  0.571420  0.444440           1.0  ...     37           9   
23      0.000000  0.000000  0.000000           0.0  ...     57           7   
24      0.000000  0.090908  0.058823           0.0  ...     48          17   
25      0.999988  0.928565  0.866661           0.0  ...     89          14   
26      0.666644  0.999975  0.666656           0.0  ...     38           4   
27      0.000000  0.285710  0.249997           0.0  ...     40           8   
28      0.749981  0.874989  0.777769           1.0  ...     41           8   
29      0.499992  0.749991  0.461535           0.0  ...     69           8   
...          ...       ...       ...           ...  ...    ...         ...   
404260  0.199996  0.499992  0.333330           0.0  ...     40           6   
404261  0.833319  0.899991  0.749994           0.0  ...     64          10   
404262  0.749981  0.714276  0.714276           1.0  ...     32           7   
404263  0.399992  0.454541  0.454541           0.0  ...     65          11   
404264  0.142855  0.399996  0.249998           0.0  ...     42          16   
404265  0.199996  0.428565  0.374995           0.0  ...     33           8   
404266  0.363633  0.583328  0.212121           0.0  ...    179          12   
404267  0.499992  0.636358  0.466664           1.0  ...     80          11   
404268  0.124998  0.299997  0.187499           0.0  ...     44          13   
404269  0.000000  0.428565  0.299997           1.0  ...     63           7   
404270  0.714276  0.857131  0.749991           0.0  ...     45           8   
404271  0.142855  0.230767  0.214284           1.0  ...     65          13   
404272  0.777769  0.562496  0.562496           1.0  ...     87          15   
404273  0.499992  0.699993  0.437497           1.0  ...     83          10   
404274  0.599988  0.555549  0.454541           0.0  ...     51          11   
404275  0.299997  0.399996  0.266665           0.0  ...     79          10   
404276  0.999950  0.749981  0.749981           1.0  ...     24           3   
404277  0.333322  0.249994  0.199996           0.0  ...     32           4   
404278  0.799984  0.749991  0.599994           0.0  ...     52           6   
404279  0.499994  0.461535  0.374998           0.0  ...     77          13   
404280  0.999980  0.909083  0.909083           1.0  ...     55          11   
404281  0.999983  0.769225  0.769225           1.0  ...     68          13   
404282  0.333322  0.499992  0.428565           0.0  ...     47           7   
404283  0.285710  0.499996  0.352939           0.0  ...     61          16   
404284  0.749981  0.999983  0.749991           1.0  ...     51           6   
404285  0.833319  0.846147  0.785709           1.0  ...     79          14   
404286  0.599988  0.624992  0.555549           1.0  ...     42           8   
404287  0.666644  0.749981  0.749981           1.0  ...     17           4   
404288  0.099999  0.058823  0.040000           0.0  ...    127          17   
404289  0.714276  0.999988  0.799992           1.0  ...     45           8   

        q2_n_words  word_Common  word_Total  word_share  freq_q1+q2  \
0               12         10.0        23.0    0.434783           2   
1               13          4.0        20.0    0.200000           5   
```

```
2               10          4.0         24.0    0.166667           2
3                9          0.0         19.0    0.000000           2
4                7          2.0         20.0    0.100000           4
5               16          8.0         31.0    0.258065           2
6               11          0.0         14.0    0.000000           2
7                9          4.0         16.0    0.250000           2
8                8          6.0         16.0    0.375000           3
9                9          3.0         18.0    0.166667           2
10              19          1.0         25.0    0.040000           2
11               8          5.0         17.0    0.294118           2
12               8          6.0         15.0    0.400000           3
13               6          5.0         13.0    0.384615           4
14              29         20.0         44.0    0.454545          10
15              17          4.0         31.0    0.129032          17
16               4          3.0          8.0    0.375000           2
17               8          1.0         20.0    0.050000           4
18              13          5.0         26.0    0.192308          70
19               9          7.0         18.0    0.388889           2
20               7          3.0         12.0    0.250000           4
21              10          1.0         16.0    0.062500           2
22               7          4.0         16.0    0.250000           8
23              11          0.0         18.0    0.000000           2
24              10          0.0         24.0    0.000000           6
25              15         13.0         29.0    0.448276           6
26               6          3.0         10.0    0.300000           4
27               7          0.0         15.0    0.000000           2
28               9          7.0         17.0    0.411765          21
29              12          6.0         20.0    0.300000          18
...            ...          ...          ...         ...         ...
404260           9          3.0         15.0    0.200000          42
404261          12          8.0         21.0    0.380952           3
404262           7          5.0         14.0    0.357143           2
404263          11          4.0         22.0    0.181818           2
404264          10          3.0         26.0    0.115385           2
404265           7          2.0         15.0    0.133333          21
404266          33          6.0         41.0    0.146341           8
404267          15          7.0         20.0    0.350000           6
404268           9          0.0         22.0    0.000000           2
404269          10          3.0         17.0    0.176471           2
404270           8          6.0         16.0    0.375000          18
404271          12          1.0         24.0    0.041667           2
404272          14          8.0         28.0    0.285714          11
404273          16          7.0         26.0    0.269231           2
404274           9          5.0         19.0    0.263158           8
404275          15          3.0         25.0    0.120000           4
404276           3          2.0          6.0    0.333333           9
404277           5          1.0          9.0    0.111111           2
404278          10          5.0         16.0    0.312500           3
404279          16          6.0         27.0    0.222222           2
404280          11         10.0         21.0    0.476190           2
404281          13         10.0         24.0    0.416667          10
404282           6          3.0         13.0    0.230769          40
404283          12          4.0         25.0    0.160000           2
404284           8          6.0         14.0    0.428571           2
404285          13         11.0         25.0    0.440000           4
404286           9          5.0         16.0    0.312500          13
404287           3          1.0          7.0    0.142857           2
404288          25          1.0         40.0    0.025000           2
404289          10          8.0         18.0    0.444444           2

        freq_q1-q2                                          question1  \
0                0  What is the step by step guide to invest in sh...
1                3  What is the story of Kohinoor (Koh-i-Noor) Dia...
2                0  How can I increase the speed of my internet co...
3                0  Why am I mentally very lonely? How can I solve...
4                2  Which one dissolve in water quikly sugar, salt...
5                0  Astrology: I am a Capricorn Sun Cap moon and c...
6                0                                Should I buy tiago?
7                0                     How can I be a good geologist?
8                1                    When do you use シ instead of し?
9                0  Motorola (company): Can I hack my Charter Moto...
10               0  Method to find separation of slits using fresn...
11               0        How do I read and find my YouTube comments?
12               1               What can make Physics easy to learn?
13               0        What was your first sexual experience like?
14               0  What are the laws to change your status from a...
15               3  What would a Trump presidency mean for current...
```

```
16                0                       What does manipulation mean?
17                2  Why do girls want to be friends with the guy t...
18               34  Why are so many Quora users posting questions ...
19                0  Which is the best digital marketing institutio...
20                2                         Why do rockets look white?
21                0              What's causing someone to be jealous?
22                0    What are the questions should not ask on Quora?
23                0                           How much is 30 kV in HP?
24                0  What does it mean that every time I look at th...
25                2  What are some tips on making it through the jo...
26                2                           What is web application?
27                0  Does society place too much importance on sports?
28               19             What is best way to make money online?
29                2             How should I prepare for CA final law?
...             ...                                                ...
404260           38                   Which phone is best under 12000?
404261            1  Who is the overall most popular Game of Throne...
404262            0          How do you troubleshoot a Toshiba laptop?
404263            0  How does the burning of fossil fuels contribut...
404264            0  Is it safe to store an external battery power ...
404265           17                  How can I gain weight on my body?
404266            0  What is the green dot next to the phone icon o...
404267            2  What are the causes of the fall of the Roman E...
404268            0  Why don't we still do great music like in the ...
404269            0  How do you diagnose antisocial personality dis...
404270           16        What is the difference between who and how?
404271            0  Does Stalin have any grandchildren that are st...
404272            1  What are the best new car products or inventio...
404273            0    What happens if you put milk in a coffee maker?
404274            0  Will the next generation of parenting change o...
404275            2  In accounting, why do we debit expenses and cr...
404276            5                         What is copilotsearch.com?
404277            0                            What does analytics do?
404278            1          How did you prepare for AIIMS/NEET/AIPMT?
404279            0  What is the minimum time required to build a f...
404280            0  What are some outfit ideas to wear to a frat p...
404281            4  Why is Manaphy childish in Pokémon Ranger and ...
404282           12        How does a long distance relationship work?
404283            0  What do you think of the removal of the MagSaf...
404284            0         What does Jainism say about homosexuality?
404285            0  How many keywords are there in the Racket prog...
404286           11          Do you believe there is life after death?
404287            0                                  What is one coin?
404288            0  What is the approx annual cost of living while...
404289            0              What is like to have sex with cousin?

                                                question2
0       What is the step by step guide to invest in sh...
1       What would happen if the Indian government sto...
2       How can Internet speed be increased by hacking...
3       Find the remainder when [math]23^{24}[/math] i...
4                 Which fish would survive in salt water?
5       I'm a triple Capricorn (Sun, Moon and ascendan...
6       What keeps childern active and far from phone ...
7               What should I do to be a great geologist?
8                   When do you use "&" instead of "and"?
9       How do I hack Motorola DCX3400 for free internet?
10      What are some of the things technicians can te...
11                 How can I see all my Youtube comments?
12                How can you make physics easy to learn?
13                 What was your first sexual experience?
14      What are the laws to change your status from a...
15      How will a Trump presidency affect the student...
16                          What does manipulation means?
17               How do guys feel after rejecting a girl?
18      Why do people ask Quora questions which can be...
19      Which is the best digital marketing institute ...
20            Why are rockets and boosters painted white?
21       What can I do to avoid being jealous of someone?
22                   Which question should I ask on Quora?
23      Where can I find a conversion chart for CC to ...
24       How many times a day do a clock’s hands overlap?
25      What are some tips on making it through the jo...
26                 What is the web application framework?
27               How do sports contribute to the society?
28              What is best way to ask for money online?
29      How one should know that he/she completely pre...
```

```
...                                                   ...
404260           What is the best phone to buy below 15k?
404261  Who is the most popular character in the Game ...
404262                  How do I reset a Toshiba laptop?
404263  Why does CO2 contribute more to global warming...
404264         How do I make a safe and cheap power bank?
404265                  What should I eat to gain weight?
404266  My boyfriend says he deleted his Facebook Mess...
404267  What were the most important causes and effect...
404268       Should I raise my young child on 80's music?
404269  What Does It Feel Like to have antisocial pers...
404270      What is the difference between "&" and "and"?
404271  What was Joseph Stalin's 5 year plan? How did ...
404272  What are some mind-blowing vehicles tools that...
404273  What would happen if I put milk instead of wat...
404274  What kind of parents will the next generation ...
404275  What is a utilities expense in accounting? How...
404276                           What is ContenVania.com?
404277                   What are analytical people like?
404278  How did you prepare for the AIIMS UG entrance ...
404279  What is a cheaper and quicker way to build an ...
404280  What are some outfit ideas wear to a frat them...
404281  Why is Manaphy annoying in Pokemon ranger and ...
404282    How are long distance relationships maintained?
404283  What will the CPU upgrade to the 2016 Apple Ma...
404284  What does Jainism say about Gays and Homosexua...
404285  How many keywords are there in PERL Programmin...
404286         Is it true that there is life after death?
404287                                  What's this coin?
404288  I am having little hairfall problem but I want...
404289      What is it like to have sex with your cousin?

[404290 rows x 31 columns]>
```

In [0]:

```
df.shape
```

Out[0]:

```
(404290, 31)
```

In [0]:

```
y_true = df['is_duplicate']
```

In [0]:

```
df=df.drop(['Unnamed: 0', 'id','is_duplicate'], axis=1, inplace=True)
```

In [0]:

```
X_train,X_test, y_train, y_test = train_test_split(df, y_true, stratify=y_true, test_size=0.3)
```

In [0]:

```
X_train["question1"].fillna(" ", inplace = True)
X_train["question2"].fillna(" ", inplace = True)
X_test["question1"].fillna(" ", inplace = True)
X_test["question2"].fillna(" ", inplace = True)
```

In [0]:

```
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.feature_extraction.text import CountVectorizer
# merge texts
train_questions = list(X_train['question1']) + list(X_train['question2'])
test_questions = list(X_test['question1']) + list(X_test['question2'])

tfidf = TfidfVectorizer(lowercase=False, )
tfidf.fit_transform(train_questions)
tfidf.fit_transform(train_questions)
tfidf.transform(test_questions)

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

- After we find TF-IDF scores, we convert each question to a weighted average of word2vec vectors by these scores.
- here we use a pre-trained GLOVE model which comes free with "Spacy". https://spacy.io/usage/vectors-similarity
- It is trained on Wikipedia and therefore, it is stronger in terms of word semantics.

In [0]:

```
# en_vectors_web_lg, which includes over 1 million unique vectors.
import en_core_web_sm
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(X_train['question1'])):
    doc1 = nlp(qu1)
    # 384 is the number of dimensions of vectors
    mean_vec1 = np.zeros([len(doc1), len(doc1[0].vector)])
    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)
X_train['q1_feats_m'] = list(vecs1)
```

In [0]:

```
vecs2 = []
for qu2 in tqdm(list(X_train['question2'])):
    doc2 = nlp(qu2)
    mean_vec2 = np.zeros([len(doc1), len(doc2[0].vector)])
    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)
X_train['q2_feats_m'] = list(vecs2)
```

In [0]:

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

vecs3 = []
# https://github.com/noamraph/tqdm
# tqdm is used to print the progress bar
for qu1 in tqdm(list(X_test['question1'])):
    doc1 = nlp(qu1)
    # 384 is the number of dimensions of vectors
    mean_vec1 = np.zeros([len(doc1), len(doc1[0].vector)])
    for word1 in doc1:
        # word2vec
        vec3 = word1.vector
        # fetch df score
        try:
            idf = word2tfidf[str(word1)]
        except:
            idf = 0
        # compute final vec
        mean_vec1 += vec3 * idf
    mean_vec1 = mean_vec1.mean(axis=0)
    vecs3.append(mean_vec1)
X_test['q1_feats_m'] = list(vecs3)
```

In [0]:

```
vecs4 = []
for qu2 in tqdm(list(X_test['question2'])):
    doc2 = nlp(qu2)
    mean_vec2 = np.zeros([len(doc1), len(doc2[0].vector)])
    for word2 in doc2:
        # word2vec
        vec4 = word2.vector
        # fetch df score
        try:
            idf = word2tfidf[str(word2)]
        except:
            #print word
            idf = 0
        # compute final vec
        mean_vec2 += vec4 * idf
    mean_vec2 = mean_vec2.mean(axis=0)
    vecs4.append(mean_vec2)
X_test['q2_feats_m'] = list(vecs4)
```

In [0]:

```
print(len(X_train['q2_feats_m'].iloc[0]))
```

```
96
```

In [0]:

```
# 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')
'''
```

In [0]:

```
X_train2=pd.DataFrame(X_train)
X_train2[['0_x','1_x','2_x','3_x','4_x','5_x','6_x','7_x','8_x','9_x','10_x','11_x','12_x','13_x','
14_x','15_x','16_x','17_x','18_x','19_x','20_x','21_x','22_x','23_x','24_x','25_x','26_x','27_x','2
8_x','29_x','30_x','31_x','32_x','33_x','34_x','35_x','36_x','37_x','38_x','39_x','40_x','41_x','42
_x','43_x','44_x','45_x','46_x','47_x','48_x','49_x','50_x','51_x','52_x','53_x','54_x','55_x','56_
x','57_x','58_x','59_x','60_x','61_x','62_x','63_x','64_x','65_x','66_x','67_x','68_x','69_x','70_x
','71_x','72_x','73_x','74_x','75_x','76_x','77_x','78_x','79_x','80_x','81_x','82_x','83_x','84_x'
,'85_x','86_x','87_x','88_x','89_x','90_x','91_x','92_x','93_x','94_x','95_x']] = pd.DataFrame(X_tr
ain2.q1_feats_m.tolist(), index= X_train2.index)
```

In [0]:

```
X_train2[['0_y','1_y','2_y','3_y','4_y','5_y','6_y','7_y','8_y','9_y','10_y','11_y','12_y','13_y','
14_y','15_y','16_y','17_y','18_y','19_y','20_y','21_y','22_y','23_y','24_y','25_y','26_y','27_y','2
8_y','29_y','30_y','31_y','32_y','33_y','34_y','35_y','36_y','37_y','38_y','39_y','40_y','41_y','42
_y','43_y','44_y','45_y','46_y','47_y','48_y','49_y','50_y','51_y','52_y','53_y','54_y','55_y','56_
y','57_y','58_y','59_y','60_y','61_y','62_y','63_y','64_y','65_y','66_y','67_y','68_y','69_y','70_y
','71_y','72_y','73_y','74_y','75_y','76_y','77_y','78_y','79_y','80_y','81_y','82_y','83_y','84_y'
,'85_y','86_y','87_y','88_y','89_y','90_y','91_y','92_y','93_y','94_y','95_y']] = pd.DataFrame(X_tr
ain2.q2_feats_m.tolist(), index= X_train2.index)
```

In [0]:

```
X_train=X_train2.drop(['question1','question2'],axis=1)
```

In [0]:

```
X_train.head()
```

Out[0]:

	cwc_min	cwc_max	csc_min	csc_max	ctc_min	ctc_max	last_word_eq	first_word_eq	abs_len_diff	mean_len	...	86_
101103	0.999950	0.666644	0.499988	0.499988	0.666656	0.571420	0.0	1.0	1.0	6.5	...	37.36559
49714	0.999975	0.999975	0.999980	0.714276	0.999989	0.818174	1.0	1.0	2.0	10.0	...	0.59574
33009	0.874989	0.874989	0.999986	0.999986	0.736838	0.736838	1.0	1.0	0.0	19.0	...	66.24353
273917	0.000000	0.000000	0.499988	0.133332	0.181817	0.068965	0.0	0.0	18.0	20.0	...	83.64601
50937	0.999967	0.374995	0.999967	0.374995	0.999983	0.374998	0.0	1.0	10.0	11.0	...	37.04170

5 rows × 220 columns

In [0]:

```
X_test2=pd.DataFrame(X_test)
X_test2[['0_x','1_x','2_x','3_x','4_x','5_x','6_x','7_x','8_x','9_x','10_x','11_x','12_x','13_x','1
4_x','15_x','16_x','17_x','18_x','19_x','20_x','21_x','22_x','23_x','24_x','25_x','26_x','27_x','28
_x','29_x','30_x','31_x','32_x','33_x','34_x','35_x','36_x','37_x','38_x','39_x','40_x','41_x','42_
x','43_x','44_x','45_x','46_x','47_x','48_x','49_x','50_x','51_x','52_x','53_x','54_x','55_x','56_x
','57_x','58_x','59_x','60_x','61_x','62_x','63_x','64_x','65_x','66_x','67_x','68_x','69_x','70_x'
,'71_x','72_x','73_x','74_x','75_x','76_x','77_x','78_x','79_x','80_x','81_x','82_x','83_x','84_x',
'85_x','86_x','87_x','88_x','89_x','90_x','91_x','92_x','93_x','94_x','95_x']] = pd.DataFrame(X_tes
t2.q1_feats_m.tolist(), index= X_test2.index)

X_test2[['0_y','1_y','2_y','3_y','4_y','5_y','6_y','7_y','8_y','9_y','10_y','11_y','12_y','13_y','1
4_y','15_y','16_y','17_y','18_y','19_y','20_y','21_y','22_y','23_y','24_y','25_y','26_y','27_y','28
_y','29_y','30_y','31_y','32_y','33_y','34_y','35_y','36_y','37_y','38_y','39_y','40_y','41_y','42_
y','43_y','44_y','45_y','46_y','47_y','48_y','49_y','50_y','51_y','52_y','53_y','54_y','55_y','56_y
','57_y','58_y','59_y','60_y','61_y','62_y','63_y','64_y','65_y','66_y','67_y','68_y','69_y','70_y'
,'71_y','72_y','73_y','74_y','75_y','76_y','77_y','78_y','79_y','80_y','81_y','82_y','83_y','84_y',
'85_y','86_y','87_y','88_y','89_y','90_y','91_y','92_y','93_y','94_y','95_y']] = pd.DataFrame(X_tes
t2.q2_feats_m.tolist(), index= X_test2.index)
```

In [0]:

```
X_test=X_test2.drop(['question1','question2'],axis=1)
```

In [0]:

```
X_test.head()
```

Out[0]:

	cwc_min	cwc_max	csc_min	csc_max	ctc_min	ctc_max	last_word_eq	first_word_eq	abs_len_diff	mean_len	...	86_
251863	0.000000	0.000000	0.249994	0.249994	0.111110	0.099999	0.0	0.0	1.0	9.5	...	43.63289
344339	0.599988	0.176470	0.666644	0.111110	0.624992	0.121951	0.0	0.0	33.0	24.5	...	-6.01040
386581	0.499988	0.499988	0.000000	0.000000	0.333328	0.181817	0.0	0.0	5.0	8.5	...	16.22251

	cwc_min	cwc_max	csc_min	csc_max	ctc_min	ctc_max	last_word_eq	first_word_eq	abs_len_diff	mean_len	...	86
190393	0.249994	0.199998	0.500000	0.000000	0.142855	0.124998	0.0	0.0	1.0	7.5	...	25.77763
24900	0.499988	0.399992	0.666644	0.285710	0.571420	0.333331	0.0	0.0	5.0	9.5	...	47.71261

5 rows × 220 columns

In [0]:

```
if not os.path.isfile('X_train.csv'):
    X_train.to_csv("X_train.csv")

if not os.path.isfile('X_test.csv'):
    X_test.to_csv("X_test.csv")
```

In [0]:

```
if not os.path.isfile('y_train.npy'):
    np.save("y_train.npy",y_train)
if not os.path.isfile('y_test.npy'):
    np.save("y_test.npy",y_test)
```

In [0]: