

DonorsChoose

DonorsChoose.org receives hundreds of thousands of project proposals each year for classroom projects in need of funding. Right now, a large number of volunteers is needed to manually screen each submission before it's approved to be posted on the DonorsChoose.org website.

Next year, DonorsChoose.org expects to receive close to 500,000 project proposals. As a result, there are three main problems they need to solve:

- How to scale current manual processes and resources to screen 500,000 projects so that they can be posted as quickly and as efficiently as possible
- How to increase the consistency of project vetting across different volunteers to improve the experience for teachers
- How to focus volunteer time on the applications that need the most assistance

The goal of the competition is to predict whether or not a DonorsChoose.org project proposal submitted by a teacher will be approved, using the text of project descriptions as well as additional metadata about the project, teacher, and school. DonorsChoose.org can then use this information to identify projects most likely to need further review before approval.

About the DonorsChoose Data Set

The `train.csv` data set provided by DonorsChoose contains the following features:

Feature		
<code>project_id</code>		A unique identifier for the proposed project. Example:
<code>project_title</code>	<ul style="list-style-type: none">••	Title of the project. Art Will Make Yc First C
<code>project_grade_category</code>	<ul style="list-style-type: none">••••	Grade level of students for which the project is targeted. One of the enumerated categories. Grade Gr Gr Gra

02/01/2020		11_DonorsChoose_TruncatedSVD		
Feature				
		One or more (comma-separated) subject categories for the project from the following enumerated list:		
project_subject_categories	<ul style="list-style-type: none">••••••••	Applied		
		Care		
		Health		
		History		
		Literacy & Language		
		Math & Science		
		Music & Arts		
		Special Education		
		<ul style="list-style-type: none">••		
		Music & Arts		
		Literacy & Language, Math & Science		
school_state		State where school is located (Two-letter U.S. state abbreviations) Ex		
		One or more (comma-separated) subject subcategories for the project from the following enumerated list:		
project_subject_subcategories		<ul style="list-style-type: none">••	Literature & Writing, Social Studies	
			An explanation of the resources needed for the project	
project_resource_summary		<ul style="list-style-type: none">• My students need hands on literacy materials in order to address sensory needs		
project_essay_1		First application essay		
project_essay_2		Second application essay		
project_essay_3		Third application essay		
project_essay_4		Fourth application essay		
project_submitted_datetime		Datetime when project application was submitted. Example: 2017-01-12T12:45:00		
teacher_id		A unique identifier for the teacher of the proposed project. Example: bdf8baa8fedef6bfeec7ae4f		
		Teacher's title. One of the following enumerated list:		
teacher_prefix		<ul style="list-style-type: none">••••••		
teacher_number_of_previously_posted_projects	Number of project applications previously submitted by the same teacher. Example: 5			

* See the section **Notes on the Essay Data** for more details about these features.

Additionally, the `resources.csv` data set provides more data about the resources required for each project. Each line in this file represents a resource required by a project:

Feature	Description
<code>id</code>	A <code>project_id</code> value from the <code>train.csv</code> file. Example: p036502
<code>description</code>	Description of the resource. Example: Tenor Saxophone Reeds, Box of 25

Feature	Description
quantity	Quantity of the resource required. Example: 3
price	Price of the resource required. Example: 9.95

Note: Many projects require multiple resources. The `id` value corresponds to a `project_id` in `train.csv`, so you use it as a key to retrieve all resources needed for a project:

The data set contains the following label (the value you will attempt to predict):

Label	Description
<code>project_is_approved</code>	A binary flag indicating whether DonorsChoose approved the project. A value of <code>0</code> indicates the project was not approved, and a value of <code>1</code> indicates the project was approved.



Notes on the Essay Data

Prior to May 17, 2016, the prompts for the essays were as follows:

- **project_essay_1:** "Introduce us to your classroom"
- **project_essay_2:** "Tell us more about your students"
- **project_essay_3:** "Describe how your students will use the materials you're requesting"
- **project_essay_3:** "Close by sharing why your project will make a difference"

Starting on May 17, 2016, the number of essays was reduced from 4 to 2, and the prompts for the first 2 essays were changed to the following:

- **project_essay_1:** "Describe your students: What makes your students special? Specific details about their background, your neighborhood, and your school are all helpful."
- **project_essay_2:** "About your project: How will these materials make a difference in your students' learning and improve their school lives?"

For all projects with `project_submitted_datetime` of 2016-05-17 and later, the values of `project_essay_3` and `project_essay_4` will be NaN.

In [1]:

```
%matplotlib inline
import warnings
warnings.filterwarnings("ignore")

import sqlite3
import pandas as pd
import numpy as np
import nltk
import string
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.feature_extraction.text import TfidfTransformer
from sklearn.feature_extraction.text import TfidfVectorizer

from sklearn.feature_extraction.text import CountVectorizer
from sklearn.metrics import confusion_matrix
from sklearn import metrics
from sklearn.metrics import roc_curve, auc
from nltk.stem.porter import PorterStemmer

import re
# Tutorial about Python regular expressions: https://pymotw.com/2/re/
import string
from nltk.corpus import stopwords
from nltk.stem import PorterStemmer
from nltk.stem.wordnet import WordNetLemmatizer

# from gensim.models import Word2Vec
# from gensim.models import KeyedVectors
import pickle

from tqdm import tqdm
import os

from plotly import plotly
import plotly.offline as offline
import plotly.graph_objs as go
offline.init_notebook_mode()
from collections import Counter
from scipy.sparse import hstack
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.linear_model import LogisticRegression
from sklearn.model_selection import RandomizedSearchCV
from sklearn.model_selection import GridSearchCV
from sklearn import preprocessing
from sklearn.metrics import confusion_matrix
from prettytable import PrettyTable
import nltk
from nltk.sentiment.vader import SentimentIntensityAnalyzer
```

In [109]:

```
nlTK.download('vader_lexicon')
```

```
[nlTK_data] Downloading package vader_lexicon to
[nltk_data] /home/neeraj_barthwal/nltk_data...
```

Out[109]:

True

1.1 Reading Data

In [2]:

```
project_data = pd.read_csv('train_data.csv')
resource_data = pd.read_csv('resources.csv')
```

Adding price attribute to project_data dataframe from resources using merge function

In [3]:

```
price_data = resource_data.groupby('id').agg({'price':'sum', 'quantity':'sum'}).res
project_data = pd.merge(project_data, price_data, on='id', how='left')
```

In [4]:

```
print("Number of data points in train data", project_data.shape)
print('-'*50)
print("The attributes of data :", project_data.columns.values)
```

```
Number of data points in train data (109248, 19)
```

```
-----
```

```
The attributes of data : ['Unnamed: 0' 'id' 'teacher_id' 'teacher_pref
ix' 'school_state'
```

```
'project_submitted_datetime' 'project_grade_category'
'project_subject_categories' 'project_subject_subcategories'
'project_title' 'project_essay_1' 'project_essay_2' 'project_essay_3'
'project_essay_4' 'project_resource_summary'
'teacher_number_of_previously_posted_projects' 'project_is_approved'
'price' 'quantity']
```

In [5]:

```
print("Number of data points in train data", resource_data.shape)
print(resource_data.columns.values)
resource_data.head(2)
```

Number of data points in train data (1541272, 4)
 ['id' 'description' 'quantity' 'price']

Out[5]:

	id	description	quantity	price
0	p233245	LC652 - Lakeshore Double-Space Mobile Drying Rack	1	149.00
1	p069063	Bouncy Bands for Desks (Blue support pipes)	3	14.95

1.2 preprocessing of project_subject_categories

In [6]:

```
categories = list(project_data['project_subject_categories'].values)
# remove special characters from list of strings python: https://stackoverflow.com/

# https://www.geeksforgeeks.org/removing-stop-words-nltk-python/
# https://stackoverflow.com/questions/23669024/how-to-strip-a-specific-word-from-a-
# https://stackoverflow.com/questions/8270092/remove-all-whitespace-in-a-string-in-
cat_list = []
for i in categories:
    temp = ""
    # consider we have text like this "Math & Science, Warmth, Care & Hunger"
    for j in i.split(','): # it will split it in three parts ["Math & Science", "Wa
        if 'The' in j.split(): # this will split each of the category based on spac
            j=j.replace('The','') # if we have the words "The" we are going to repl
        j = j.replace(' ', '') # we are placing all the ' '(space) with ''(empty) e
        temp+=j.strip()+" " # " abc ".strip() will return "abc", remove the trailing
        temp = temp.replace('&','_') # we are replacing the & value into
    cat_list.append(temp.strip())

project_data['clean_categories'] = cat_list
project_data.drop(['project_subject_categories'], axis=1, inplace=True)

from collections import Counter
my_counter = Counter()
for word in project_data['clean_categories'].values:
    my_counter.update(word.split())

cat_dict = dict(my_counter)
sorted_cat_dict = dict(sorted(cat_dict.items(), key=lambda kv: kv[1]))
```

1.3 preprocessing of project_subject_subcategories

In [7]:

```

sub_categories = list(project_data['project_subject_subcategories'].values)
# remove special characters from list of strings python: https://stackoverflow.com/

# https://www.geeksforgeeks.org/removing-stop-words-nltk-python/
# https://stackoverflow.com/questions/23669024/how-to-strip-a-specific-word-from-a-
# https://stackoverflow.com/questions/8270092/remove-all-whitespace-in-a-string-in-

sub_cat_list = []
for i in sub_categories:
    temp = ""
    # consider we have text like this "Math & Science, Warmth, Care & Hunger"
    for j in i.split(','): # it will split it in three parts ["Math & Science", "Wa
        if 'The' in j.split(): # this will split each of the category based on spac
            j=j.replace('The', '') # if we have the words "The" we are going to repl
            j = j.replace(' ', '') # we are placing all the ' '(space) with ''(empty) e
            temp +=j.strip()+" #" abc ".strip() will return "abc", remove the trailing
            temp = temp.replace('&', '_')
    sub_cat_list.append(temp.strip())

project_data['clean_subcategories'] = sub_cat_list
project_data.drop(['project_subject_subcategories'], axis=1, inplace=True)

# count of all the words in corpus python: https://stackoverflow.com/a/22898595/408
my_counter = Counter()
for word in project_data['clean_subcategories'].values:
    my_counter.update(word.split())

sub_cat_dict = dict(my_counter)
sorted_sub_cat_dict = dict(sorted(sub_cat_dict.items(), key=lambda kv: kv[1]))

```

1.3 Text preprocessing

In [8]:

```

# merge two column text dataframe:
project_data["essay"] = project_data["project_essay_1"].map(str) + \
    project_data["project_essay_2"].map(str) + \
    project_data["project_essay_3"].map(str) + \
    project_data["project_essay_4"].map(str)

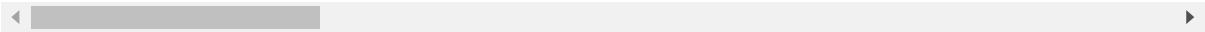
```

In [9]:

```
project_data.head(2)
```

Out[9]:

Unnamed: 0		id	teacher_id	teacher_prefix	school_state	project
0	160221	p253737	c90749f5d961ff158d4b4d1e7dc665fc	Mrs.	IN	
1	140945	p258326	897464ce9ddc600bced1151f324dd63a	Mr.	FL	



In [10]:

```
#### 1.4.2.3 Using Pretrained Models: TFIDF weighted W2V
```


In [11]:

```
# printing some random reviews
print(project_data['essay'].values[0])
print("="*50)
print(project_data['essay'].values[150])
print("="*50)
print(project_data['essay'].values[1000])
print("="*50)
print(project_data['essay'].values[20000])
print("="*50)
print(project_data['essay'].values[99999])
print("="*50)
```

My students are English learners that are working on English as their second or third languages. We are a melting pot of refugees, immigrants, and native-born Americans bringing the gift of language to our school. \r\n\r\n We have over 24 languages represented in our English Learner program with students at every level of mastery. We also have over 40 countries represented with the families within our school. Each student brings a wealth of knowledge and experiences to us that open our eyes to new cultures, beliefs, and respect.\r\n\r\nThe limits of your language are the limits of your world.\r\n\r\n-Ludwig Wittgenstein Our English learner's have a strong support system at home that begs for more resources. Many times our parents are learning to read and speak English along side of their children. Sometimes this creates barriers for parents to be able to help their child learn phonetics, letter recognition, and other reading skills.\r\n\r\n\r\nBy providing these dvd's and players, students are able to continue their mastery of the English language even if no one at home is able to assist. All families with students within the Level 1 proficiency status, will be offered to be a part of this program. These educational videos will be specially chosen by the English Learner Teacher and will be sent home regularly to watch. The videos are to help the child develop early reading skills.\r\n\r\n\r\nParents that do not have access to a dvd player will have the opportunity to check out a dvd player to use for the year. The plan is to use these videos and educational dvd's for the years to come for other EL students.\r\n\r\nnnnnnn

=====

The 51 fifth grade students that will cycle through my classroom this year all love learning, at least most of the time. At our school, 97.3% of the students receive free or reduced price lunch. Of the 560 students, 97.3% are minority students. \r\n\r\nThe school has a vibrant community that loves to get together and celebrate. Around Halloween there is a whole school parade to show off the beautiful costumes that students wear. On Cinco de Mayo we put on a big festival with crafts made by the students, dances, and games. At the end of the year the school hosts a carnival to celebrate the hard work put in during the school year, with a dunk tank being the most popular activity. My students will use these five brightly colored Hokki stools in place of regular, stationary, 4-legged chairs. As I will only have a total of ten in the classroom and not enough for each student to have an individual one, they will be used in a variety of ways. During independent reading time they will be used as special chairs students will each use on occasion. I will utilize them in place of chairs at my small group tables during math and reading times. The rest of the day they will be used by the students who need the highest amount of movement in their life in order to stay focused on school.\r\n\r\n\r\nWhenever asked what the classroom is missing, my students always say more Hokki Stools. They can't get their fill of the 5 stools we already have. When the students are sitting in group with me on the Hokki Stools, they are always moving, but at t

he same time doing their work. Anytime the students get to pick where they can sit, the Hokki Stools are the first to be taken. There are always students who head over to the kidney table to get one of the stools who are disappointed as there are not enough of them. \r\n\r\nWe ask a lot of students to sit for 7 hours a day. The Hokki stools will be a compromise that allow my students to do desk work and move at the same time. These stools will help students to meet their 60 minutes a day of movement by allowing them to activate their core muscles for balance while they sit. For many of my students, these chairs will take away the barrier that exists in schools for a child who can't sit still. nannan

=====

How do you remember your days of school? Was it in a sterile environment with plain walls, rows of desks, and a teacher in front of the room? A typical day in our room is nothing like that. I work hard to create a warm inviting themed room for my students look forward to coming to each day.\r\n\r\nMy class is made up of 28 wonderfully unique boys and girls of mixed races in Arkansas.\r\nThey attend a Title I school, which means there is a high enough percentage of free and reduced-price lunch to qualify. Our school is an "open classroom" concept, which is very unique as there are no walls separating the classrooms. These 9 and 10 year-old students are very eager learners; they are like sponges, absorbing all the information and experiences and keep on wanting more. With these resources such as the comfy red throw pillows and the whimsical nautical hanging decor and the blue fish nets, I will be able to help create the mood in our classroom setting to be one of a themed nautical environment. Creating a classroom environment is very important in the success in each and every child's education. The nautical photo props will be used with each child as they step foot into our classroom for the first time on Meet the Teacher evening. I'll take pictures of each child with them, have them developed, and then hung in our classroom ready for their first day of 4th grade. This kind gesture will set the tone before even the first day of school! The nautical thank you cards will be used throughout the year by the students as they create thank you cards to their team groups.\r\n\r\nYour generous donations will help me to help make our classroom a fun, inviting, learning environment from day one.\r\n\r\nIt costs lost of money out of my own pocket on resources to get our classroom ready. Please consider helping with this project to make our new school year a very successful one. Thank you!nannan

=====

My kindergarten students have varied disabilities ranging from speech and language delays, cognitive delays, gross/fine motor delays, to autism. They are eager beavers and always strive to work their hardest working past their limitations. \r\n\r\nThe materials we have are the ones I seek out for my students. I teach in a Title I school where most of the students receive free or reduced price lunch. Despite their disabilities and limitations, my students love coming to school and come eager to learn and explore. Have you ever felt like you had ants in your pants and you needed to groove and move as you were in a meeting? This is how my kids feel all the time. They want to be able to move as they learn or so they say. Wobble chairs are the answer and I love them because they develop their core, which enhances gross motor and in turn fine motor skills. \r\nThey also want to learn through games, my kids don't want to sit and do worksheets. They want to learn to count by jumping and playing. Physical engagement is the key to our success. The number toss and color and shape mats can make that happen. My students will forget they are doing work and just have the fun a 6 year old deserves. nannan

=====

The mediocre teacher tells. The good teacher explains. The superior te

acher demonstrates. The great teacher inspires. -William A. Ward\r\n\r\nMy school has 803 students which is makeup is 97.6% African-American, making up the largest segment of the student body. A typical school in Dallas is made up of 23.2% African-American students. Most of the students are on free or reduced lunch. We aren't receiving doctors, lawyers, or engineers children from rich backgrounds or neighborhoods. As an educator I am inspiring minds of young children and we focus not only on academics but one smart, effective, efficient, and disciplined students with good character. In our classroom we can utilize the Bluetooth for swift transitions during class. I use a speaker which doesn't amplify the sound enough to receive the message. Due to the volume of my speaker my students can't hear videos or books clearly and it isn't making the lessons as meaningful. But with the bluetooth speaker my students will be able to hear and I can stop, pause and replay it at any time.\r\nThe cart will allow me to have more room for storage of things that are needed for the day and has an extra part to it I can use. The table top chart has all of the letter, words and pictures for students to learn about different letters and it is more accessible.nannan
=====

In [12]:

```
# https://stackoverflow.com/a/47091490/4084039
import re

def decontracted(phrase):
    # specific
    phrase = re.sub(r"won't", "will not", phrase)
    phrase = re.sub(r"can't", "can not", phrase)

    # general
    phrase = re.sub(r"n't", " not", phrase)
    phrase = re.sub(r"\'re", " are", phrase)
    phrase = re.sub(r"\'s", " is", phrase)
    phrase = re.sub(r"\'d", " would", phrase)
    phrase = re.sub(r"\'ll", " will", phrase)
    phrase = re.sub(r"\'t", " not", phrase)
    phrase = re.sub(r"\'ve", " have", phrase)
    phrase = re.sub(r"\'m", " am", phrase)
    return phrase
```

In [13]:

```
sent = decontracted(project_data['essay'].values[20000])
print(sent)
print("="*50)
```

My kindergarten students have varied disabilities ranging from speech and language delays, cognitive delays, gross/fine motor delays, to autism. They are eager beavers and always strive to work their hardest working past their limitations. \r\n\r\nThe materials we have are the ones I seek out for my students. I teach in a Title I school where most of the students receive free or reduced price lunch. Despite their disabilities and limitations, my students love coming to school and come eager to learn and explore. Have you ever felt like you had ants in your pants and you needed to groove and move as you were in a meeting? This is how my kids feel all the time. They want to be able to move as they learn or so they say. Wobble chairs are the answer and I love them because they develop their core, which enhances gross motor and in turn fine motor skills. \r\nThey also want to learn through games, my kids do not want to sit and do worksheets. They want to learn to count by jumping and playing. Physical engagement is the key to our success. The number toss and color and shape mats can make that happen. My students will forget they are doing work and just have the fun a 6 year old deserves. nannan

=====

In [14]:

```
# \r \n \t remove from string python: http://texthandler.com/info/remove-line-break
sent = sent.replace('\r', ' ')
sent = sent.replace('\n', ' ')
sent = sent.replace('\t', ' ')
print(sent)
```

My kindergarten students have varied disabilities ranging from speech and language delays, cognitive delays, gross/fine motor delays, to autism. They are eager beavers and always strive to work their hardest working past their limitations. The materials we have are the ones I seek out for my students. I teach in a Title I school where most of the students receive free or reduced price lunch. Despite their disabilities and limitations, my students love coming to school and come eager to learn and explore. Have you ever felt like you had ants in your pants and you needed to groove and move as you were in a meeting? This is how my kids feel all the time. They want to be able to move as they learn or so they say. Wobble chairs are the answer and I love them because they develop their core, which enhances gross motor and in turn fine motor skills. They also want to learn through games, my kids do not want to sit and do worksheets. They want to learn to count by jumping and playing. Physical engagement is the key to our success. The number toss and color and shape mats can make that happen. My students will forget they are doing work and just have the fun a 6 year old deserves. nannan

In [15]:

```
#remove spacial character: https://stackoverflow.com/a/5843547/4084039
sent = re.sub('[^A-Za-z0-9]+', ' ', sent)
print(sent)
```

My kindergarten students have varied disabilities ranging from speech and language delays cognitive delays gross fine motor delays to autism They are eager beavers and always strive to work their hardest working past their limitations The materials we have are the ones I seek out for my students I teach in a Title I school where most of the students receive free or reduced price lunch Despite their disabilities and limitations my students love coming to school and come eager to learn and explore Have you ever felt like you had ants in your pants and you needed to groove and move as you were in a meeting This is how my kids feel all the time They want to be able to move as they learn or so they say Wobble chairs are the answer and I love them because they develop their core which enhances gross motor and in turn fine motor skills They also want to learn through games my kids do not want to sit and do worksheets They want to learn to count by jumping and playing Physical engagement is the key to our success The number toss and color and shape mats can make that happen My students will forget they are doing work and just have the fun a 6 year old deserves nannan

In [16]:

```
# https://gist.github.com/sebleier/554280
# we are removing the words from the stop words list: 'no', 'nor', 'not'
stopwords= {'i', 'me', 'my', 'myself', 'we', 'our', 'ours', 'ourselves', 'you', "you'll", "you'd", 'your', 'yours', 'yourself', 'yourselves', 'he', 'him', 'she', "she's", 'her', 'hers', 'herself', 'it', "it's", 'its', 'itself', 'theirs', 'themselves', 'what', 'which', 'who', 'whom', 'this', 'that', 'am', 'is', 'are', 'was', 'were', 'be', 'been', 'being', 'have', 'has', 'did', 'doing', 'a', 'an', 'the', 'and', 'but', 'if', 'or', 'because', 'at', 'by', 'for', 'with', 'about', 'against', 'between', 'into', 'through', 'above', 'below', 'to', 'from', 'up', 'down', 'in', 'out', 'on', 'off', 'then', 'once', 'here', 'there', 'when', 'where', 'why', 'how', 'all', 'most', 'other', 'some', 'such', 'only', 'own', 'same', 'so', 'than', 's', 't', 'can', 'will', 'just', 'don', "don't", 'should', "should've", 've', 'y', 'ain', 'aren', "aren't", 'couldn', "couldn't", 'didn', "didn't", 'hadn't', 'hasn', "hasn't", 'haven', "haven't", 'isn', "isn't", 'ma', "mustn't", 'needn', "needn't", 'shan', "shan't", 'shouldn', "shouldn't", 'won', "won't", 'wouldn', "wouldn't"}
```

In [17]:

```
# Combining all the above students
from tqdm import tqdm
preprocessed_essays = []
# tqdm is for printing the status bar
for sentence in tqdm(project_data['essay'].values):
    sent = sentence.lower().strip()
    sent = ' '.join(e for e in sent.split() if e not in stopwords)
    sent = decontracted(sent)
    sent = sent.replace('\r', ' ')
    sent = sent.replace('\n', ' ')
    sent = sent.replace('\n', ' ')
    sent = re.sub('[^A-Za-z0-9]+', ' ', sent)
    # https://gist.github.com/sebleier/554280
    preprocessed_essays.append(sent)
```

100%|██████████| 109248/109248 [00:14<00:00, 7329.00it/s]

In [18]:

```
# after preprocessing
preprocessed_essays[20000]
```

Out[18]:

'kindergarten students varied disabilities ranging speech language del
ays cognitive delays gross fine motor delays autism eager beavers alwa
ys strive work hardest working past limitations the materials ones see
k students teach title school students receive free reduced price lunc
h despite disabilities limitations students love coming school come ea
ger learn explore have ever felt like ants pants needed groove move me
eting kids feel time want able move learn say wobble chairs answer lov
e develop core enhances gross motor turn fine motor skills they also w
ant learn games kids want sit worksheets want learn count jumping play
ing physical engagement key success number toss color shape mats make
happen students forget work fun 6 year old deserves nannan'

In [19]:

```
project_data['clean_essay'] = preprocessed_essays
```

In [20]:

```
project_data.drop(['project_essay_1', 'project_essay_2', 'project_essay_3', 'project_e
```

1.4 Preprocessing of project_title

- Decontract project titles, remove line breaks and extra spaces, convert everything to lowercase and then remove all the stop words.

In [21]:

```

preprocessed_titles = []

for title in tqdm(project_data['project_title'].values):
    title = title.lower().strip()
    title = ' '.join(e for e in title.split() if e.lower() not in stopwords)
    title = decontracted(title)
    title = title.replace('\\r', ' ')
    title = title.replace('\\\"', ' ')
    title = title.replace('\\n', ' ')
    title = re.sub('[^A-Za-z0-9]+', ' ', title)
    preprocessed_titles.append(title)

```

100%|██████████| 109248/109248 [00:01<00:00, 69283.19it/s]

In [22]:

```

project_data['clean_title'] = preprocessed_titles
project_data.drop(['project_title'],axis=1,inplace=True)

```

Pre-processing teacher_prefix

In [23]:

```

#remove nan from teacher prefix:
#https://stackoverflow.com/questions/21011777/how-can-i-remove-nan-from-list-python
def remove_nan(prefix):
    if str(prefix)!='nan':
        pr = str(prefix)
        pr = re.sub("\\.", "", pr) #remove dot from the end of prefix
        return pr
    return "none"

cleaned_teacher_prefix = project_data['teacher_prefix'].map(remove_nan)
project_data['clean_teacher_prefix'] = cleaned_teacher_prefix

```

In [24]:

```

project_data.drop(['teacher_prefix'],axis=1,inplace=True)

```

Pre-process project_grade_category

- Clean the project grade categories:
 - Convert Grades 3-5 ==> Grades_3_5

In [25]:

```
def clean_project_grades(grade):
    grade = re.sub("\-", "_", grade)
    grade = re.sub(" ", "_", grade)
    return grade.strip()

clean_grades = project_data['project_grade_category'].map(clean_project_grades)
project_data['clean_grade_category'] = clean_grades
```

In [26]:

```
project_data.drop(['project_grade_category'], axis=1, inplace=True)
```

In [27]:

```
# Dropping all features we won't need going forward
project_data.drop(['project_resource_summary'], axis=1, inplace=True)
project_data.drop(['Unnamed: 0', 'teacher_id'], axis=1, inplace=True)
```

In [28]:

```
project_data.head(2)
```

Out[28]:

	id	school_state	project_submitted_datetime	teacher_number_of_previously_posted_proj
0	p253737	IN	2016-12-05 13:43:57	
1	p258326	FL	2016-10-25 09:22:10	

In [29]:

```
def count_words(col):
    wc = []
    for txt in tqdm(project_data[col]):
        w = 0
        for t in txt.split():
            w+=1
        wc.append(w)
    return wc
```

Counting number of words in essay and title

In [30]:

```
essay_wc = count_words('clean_essay')
title_wc = count_words('clean_title')
project_data['title_wc'] = title_wc
project_data['essay_wc'] = essay_wc
```

```
100%|██████████| 109248/109248 [00:01<00:00, 77013.28it/s]
100%|██████████| 109248/109248 [00:00<00:00, 1034117.03it/s]
```

Assignment 11: TruncatedSVD

- **step 1** Select the top 2k words from essay text and project_title (concatenate essay text with project title and then find the top 2k words) based on their `idf_` (https://scikit-learn.org/stable/modules/generated/sklearn.feature_extraction.text.TfidfVectorizer.html) values
- **step 2** Compute the co-occurrence matrix with these 2k words, with window size=5 ([ref](https://www.analyticsvidhya.com/blog/2017/06/word-embeddings-count-word2veec/) (<https://www.analyticsvidhya.com/blog/2017/06/word-embeddings-count-word2veec/>))



- **step 3** Use [TruncatedSVD](http://scikit-learn.org/stable/modules/generated/sklearn.decomposition.TruncatedSVD.html) (<http://scikit-learn.org/stable/modules/generated/sklearn.decomposition.TruncatedSVD.html>) on calculated co-occurrence matrix and reduce its dimensions, choose the number of components (`n_components`) using [elbow method](https://www.appliedaicourse.com/course/applied-ai-course-online/lessons/pca-code-example-using-non-visualization/) (<https://www.appliedaicourse.com/course/applied-ai-course-online/lessons/pca-code-example-using-non-visualization/>)

- The shape of the matrix after TruncatedSVD will be $2000 \times n$, i.e. each row represents a vector form of the corresponding word.
- Vectorize the essay text and project titles using these word vectors. (while vectorizing, do ignore all the words which are not in top 2k words)

- **step 4** Concatenate these truncatedSVD matrix, with the matrix with features
 - **school_state** : categorical data
 - **clean_categories** : categorical data
 - **clean_subcategories** : categorical data
 - **project_grade_category** : categorical data
 - **teacher_prefix** : categorical data
 - **quantity** : numerical data
 - **teacher_number_of_previously_posted_projects** : numerical data
 - **price** : numerical data
 - **sentiment score's of each of the essay** : numerical data
 - **number of words in the title** : numerical data
 - **number of words in the combine essays** : numerical data
 - **word vectors calculated in step 3** : numerical data
- **step 5:** Apply GBDT on matrix that was formed in **step 4** of this assignment, **DO REFER THIS BLOG: XGBOOST DMATRIX** (<https://www.kdnuggets.com/2017/03/simple-xgboost-tutorial-iris-dataset.html>)
- **step 6:**Hyper parameter tuning (Consider any two hyper parameters)
 - Find the best hyper parameter which will give the maximum [AUC](https://www.appliedaicourse.com/course/applied-ai-course-online/lessons/receiver-operating-characteristic-curve-roc-curve-and-auc-1/) (<https://www.appliedaicourse.com/course/applied-ai-course-online/lessons/receiver-operating-characteristic-curve-roc-curve-and-auc-1/>) value
 - Find the best hyper paramter using k-fold cross validation or simple cross validation data
 - Use gridsearch cv or randomsearch cv or you can also write your own for loops to do this task of hyperparameter tuning

Note: Data Leakage

1. There will be an issue of data-leakage if you vectorize the entire data and then split it into train/cv/test.
2. To avoid the issue of data-leakage, make sure to split your data first and then vectorize it.
3. While vectorizing your data, apply the method `fit_transform()` on you train data, and apply the method `transform()` on cv/test data.
4. For more details please go through this [link. \(https://soundcloud.com/applied-ai-course/leakage-bow-and-tfidf\)](https://soundcloud.com/applied-ai-course/leakage-bow-and-tfidf)

2. TruncatedSVD

2.1 Splitting data into Train and cross validation(or test): Stratified Sampling

In [31]:

```
#Sampling only 50,000 points
project_data = project_data.sample(n=50000)
```

In [32]:

```
#Separating features and label column
Y = project_data['project_is_approved']
X = project_data.drop(['project_is_approved', 'id'],axis=1)
print("Shape of X: ",X.shape)
print("Shape of Y: ",Y.shape)
```

```
Shape of X:  (50000, 14)
Shape of Y:  (50000,)
```

In [33]:

```
#separating data into train and test
X_train, X_test, Y_train, Y_test = train_test_split(X,Y,test_size=0.30,stratify=Y)
print("Shape of X_train: ", X_train.shape)
print("Shape of Y_train: ",Y_train.shape)
print("Shape of X_test: ",X_test.shape)
print("Shape of Y_test: ",Y_test.shape)
```

```
Shape of X_train:  (35000, 14)
Shape of Y_train:  (35000,)
Shape of X_test:   (15000, 14)
Shape of Y_test:   (15000,)
```

In [34]:

```
X_train.columns
```

Out[34]:

```
Index(['school_state', 'project_submitted_datetime',  
      'teacher_number_of_previously_posted_projects', 'price', 'quantity',  
      'clean_categories', 'clean_subcategories', 'essay', 'clean_essay',  
      'clean_title', 'clean_teacher_prefix', 'clean_grade_category',  
      'title_wc', 'essay_wc'],  
      dtype='object')
```

2.2 Make Data Model Ready: encoding numerical, categorical features

2.2.1 Encoding Categorical Features

One hot encoding: clean_categories

In [35]:

```
from collections import Counter  
my_counter = Counter()  
for word in X_train['clean_categories'].values:  
    my_counter.update(word.split())  
  
cat_dict = dict(my_counter)  
sorted_cat_dict = dict(sorted(cat_dict.items(), key=lambda kv: kv[1]))
```

In [36]:

```
# we use count vectorizer to convert the values into one  
vectorizer_category = CountVectorizer(vocabulary=list(sorted_cat_dict.keys()), lowercase=True)  
vectorizer_category.fit(X_train['clean_categories'].values)  
  
X_train_category_ohe = vectorizer_category.transform(X_train['clean_categories'].values)  
X_test_category_ohe = vectorizer_category.transform(X_test['clean_categories'].values)
```

In [37]:

```
print(vectorizer_category.get_feature_names())
print("Shape of X_train after one hot encoding ",X_train_category_ohe.shape)
print("Shape of X_test after one hot encoding ",X_test_category_ohe.shape)
print("Print some random encoded categories: ")
print(X_train_category_ohe[0].toarray())
print(X_test_category_ohe[15].toarray())
```

```
['Warmth', 'Care_Hunger', 'History_Civics', 'Music_Arts', 'AppliedLearning', 'SpecialNeeds', 'Health_Sports', 'Math_Science', 'Literacy_Language']
Shape of X_train after one hot encoding (35000, 9)
Shape of X_test after one hot encoding (15000, 9)
Print some random encoded categories:
[[0 0 0 0 0 1 0 0 0]]
[[0 0 1 0 0 0 0 0 0]]
```

One hot encoding: clean_subcategories

In [38]:

```
# count of all the words in corpus python: https://stackoverflow.com/a/22898595/408
my_counter = Counter()
for word in X_train['clean_subcategories'].values:
    my_counter.update(word.split())

sub_cat_dict = dict(my_counter)
sorted_sub_cat_dict = dict(sorted(sub_cat_dict.items(), key=lambda kv: kv[1]))
```

In [39]:

```
# we use count vectorizer to convert the values into one
vectorizer_subcategory = CountVectorizer(vocabulary=list(sorted_sub_cat_dict.keys()))
vectorizer_subcategory.fit(X_train['clean_subcategories'].values)

X_train_subcategory_ohe = vectorizer_subcategory.transform(X_train['clean_subcategories'])
X_test_subcategory_ohe = vectorizer_subcategory.transform(X_test['clean_subcategories'])
```

In [40]:

```
print(vectorizer_subcategory.get_feature_names())
print("Shape of X_train subcategory after one hot encoding ",X_train_subcategory_ohe.shape)
print("Shape of X_test subcategory after one hot encoding ",X_test_subcategory_ohe.shape)
print("Print some random encoded categories: ")
print(X_train_subcategory_ohe[0].toarray())
print(X_test_subcategory_ohe[10].toarray())
```

```
[ 'Economics', 'CommunityService', 'FinancialLiteracy', 'ParentInvolvement', 'Extracurricular', 'Civics_Government', 'ForeignLanguages', 'NutritionEducation', 'Warmth', 'Care_Hunger', 'SocialSciences', 'PerformingArts', 'CharacterEducation', 'TeamSports', 'Other', 'College_CareerPrep', 'History_Geography', 'Music', 'Health_LifeScience', 'EarlyDevelopment', 'ESL', 'Gym_Fitness', 'EnvironmentalScience', 'VisualArts', 'Health_Wellness', 'AppliedSciences', 'SpecialNeeds', 'Literature_Writing', 'Mathematics', 'Literacy']
```

```
Shape of X_train subcategory after one hot encoding (35000, 30)
```

```
Shape of X_test subcategory after one hot encoding (15000, 30)
```

Print some random encoded categories:

[illegible][illegible]

One hot encoding: school_state

In [41]:

```
# create a vocabulary for states
unique_states = np.unique(X_train['school_state'].values)

vectorizer_state = CountVectorizer(vocabulary=unique_states, lowercase=False, binary=True)
vectorizer_state.fit(X_train['school_state'].values)

X_train_school_state_ohe = vectorizer_state.transform(X_train['school_state'].values)
X_test_school_state_ohe = vectorizer_state.transform(X_test['school_state'].values)
```

In [42]:

```
print(vectorizer_state.get_feature_names())
print("Shape of X_train school_state after one hot encoding ",X_train_school_state_ohe.shape)
print("Shape of X_test school_state after one hot encoding ",X_test_school_state_ohe.shape)
print("Print some random encoded school_state: ")
print(X_train_school_state_ohe[0].toarray())
print(X_test_school_state_ohe[15].toarray())
```

```
[ 'AK', 'AL', 'AR', 'AZ', 'CA', 'CO', 'CT', 'DC', 'DE', 'FL', 'GA', 'HI', 'IA', 'ID', 'IL', 'IN', 'KS', 'KY', 'LA', 'MA', 'MD', 'ME', 'MI', 'MN', 'MO', 'MS', 'MT', 'NC', 'ND', 'NE', 'NH', 'NJ', 'NM', 'NV', 'NY', 'OH', 'OK', 'OR', 'PA', 'RI', 'SC', 'SD', 'TN', 'TX', 'UT', 'VA', 'VT', 'WA', 'WI', 'WV', 'WY']
```

```
Shape of X train school state after one hot encoding (35000, 51)
```

```
Shape of X_train school state after one hot encoding (52000, 51)
Shape of X test school state after one hot encoding (15000, 51)
```

Print some random encoded school state:

[illegible][illegible]

One hot encoding: teacher_prefix

In [43]:

```
unique_teacher_prefix = np.unique(X_train['clean_teacher_prefix'])

vectorizer_teacher_prefix = CountVectorizer(vocabulary=unique_teacher_prefix, lowercase=False)
vectorizer_teacher_prefix.fit(X_train['clean_teacher_prefix'].values)

X_train_teacher_prefix_ohe = vectorizer_teacher_prefix.transform(X_train['clean_teacher_prefix'])
X_test_teacher_prefix_ohe = vectorizer_teacher_prefix.transform(X_test['clean_teacher_prefix'])
```

In [44]:

```
print(vectorizer_teacher_prefix.get_feature_names())
print("Shape of X_train clean_teacher_prefix after one hot encoding ", X_train_teacher_prefix_ohe.shape)
print("Shape of X_test clean_teacher_prefix after one hot encoding ", X_test_teacher_prefix_ohe.shape)
print("Print some random encoded clean_teacher_prefix: ")
print(X_train_teacher_prefix_ohe[0].toarray())
print(X_test_teacher_prefix_ohe[15].toarray())
```

```
['Dr', 'Mr', 'Mrs', 'Ms', 'Teacher', 'none']
Shape of X_train clean_teacher_prefix after one hot encoding (35000, 6)
Shape of X_test clean_teacher_prefix after one hot encoding (15000, 6)
Print some random encoded clean_teacher_prefix:
[[0 0 0 1 0 0]]
[[0 0 1 0 0 0]]
```

One hot encoding: project_grade_category

In [45]:

```
unique_grades = np.unique(X_train['clean_grade_category'])

vectorizer_grade = CountVectorizer(vocabulary=unique_grades, lowercase=False, binary=True)
vectorizer_grade.fit(X_train['clean_grade_category'].values)

X_train_grade_category_ohe = vectorizer_grade.transform(X_train['clean_grade_category'])
X_test_grade_category_ohe = vectorizer_grade.transform(X_test['clean_grade_category'])
```

In [46]:

```
print(vectorizer_grade.get_feature_names())
print("Shape of X_train clean_grade_category after one hot encoding ", X_train_grade_category_ohe.shape)
print("Shape of X_test clean_grade_category after one hot encoding ", X_test_grade_category_ohe.shape)
print("Print some random encoded clean_grade_category: ")
print(X_train_grade_category_ohe[0].toarray())
print(X_test_grade_category_ohe[15].toarray())
```

```
['Grades_3_5', 'Grades_6_8', 'Grades_9_12', 'Grades_PreK_2']
Shape of X_train clean_grade_category after one hot encoding (35000, 4)
Shape of X_test clean_grade_category after one hot encoding (15000, 4)
Print some random encoded clean_grade_category:
[[0 0 0 1]]
[[0 0 1 0]]
```

2.2.2 Encoding Numerical features

Normalizing Price

In [47]:

```
price_vectorizer = preprocessing.Normalizer().fit(X_train['price'].values.reshape(1
```

In [48]:

```
X_train_price_normalized = price_vectorizer.transform(X_train['price'].values.resha
X_test_price_normalized = price_vectorizer.transform(X_test['price'].values.reshape
```

In [49]:

```
X_train_price_normalized
```

Out[49]:

```
array([[0.00326379, 0.00166269, 0.00067771, ..., 0.00688095, 0.0003785
8,
        0.00082507]])
```

In [50]:

```
X_test_price_normalized
```

Out[50]:

```
array([[0.01584354, 0.00964284, 0.00101259, ..., 0.00140745, 0.0018847
,
        0.00232991]])
```

Normalize teacher_number_of_previously_posted_projects

In [51]:

```
project_vectorizer = preprocessing.Normalizer().fit(X_train['teacher_number_of_prev
```

In [52]:

```
X_train_normal_previous_project = project_vectorizer.transform(X_train['teacher_num
X_test_normal_previous_project = project_vectorizer.transform(X_test['teacher_numbe
```

Selecting top 2k features

Concatinating essay and text

In [53]:

```
X_train['combined_text'] = X_train[['clean_essay', 'clean_title']].apply(lambda x:
```

In [54]:

```
tfidf_vectorizer = TfidfVectorizer(min_df = 10)
tfidf_values = tfidf_vectorizer.fit_transform(X_train['combined_text'].values)
```

In [55]:

```
print(tfidf_values.shape)
```

```
(35000, 10835)
```


In [56]:

```
sorted_indices = np.argsort(tfidf_vectorizer.idf_)[::-1]
features = tfidf_vectorizer.get_feature_names()
top_n = 2000

top_features = [features[i] for i in sorted_indices[:top_n]]
top_idf = [tfidf_vectorizer.idf_[i] for i in sorted_indices[:top_n]]

print(list(zip(top_features, top_idf)))

[('gardeners', 9.065236638693596), ('unlikely', 9.065236638693596),
 ('html', 9.065236638693596), ('reptiles', 9.065236638693596), ('deduct
ive', 9.065236638693596), ('humility', 9.065236638693596), ('replica',
9.065236638693596), ('replay', 9.065236638693596), ('constrained', 9.0
65236638693596), ('improvising', 9.065236638693596), ('impulses', 9.06
5236638693596), ('consonant', 9.065236638693596), ('incomparable', 9.0
65236638693596), ('stopwatches', 9.065236638693596), ('incorrect', 9.0
65236638693596), ('considerably', 9.065236638693596), ('indicating',
9.065236638693596), ('reinforcers', 9.065236638693596), ('industriou
s', 9.065236638693596), ('entails', 9.065236638693596), ('inequality',
9.065236638693596), ('rehabilitation', 9.065236638693596), ('infancy',
9.065236638693596), ('aka', 9.065236638693596), ('hose', 9.06523663869
3596), ('sweeping', 9.065236638693596), ('inferring', 9.06523663869359
6), ('amplified', 9.065236638693596), ('corporate', 9.06523663869359
6), ('reynolds', 9.065236638693596), ('sigh', 9.065236638693596), ('et
hiopian', 9.065236638693596), ('hayward', 9.065236638693596), ('declin
ing', 9.065236638693596), ('amusement', 9.065236638693596), ('supporte
r', 9.065236638693596), ('ergonomically', 9.065236638693596), ('coordi
nating', 9.065236638693596), ('beethoven', 9.065236638693596), ('decon
struct', 9.065236638693596), ('simpler', 9.065236638693596), ('restor
e', 9.065236638693596), ('surpassing', 9.065236638693596), ('heritage
s', 9.065236638693596), ('highlighter', 9.065236638693596), ('aluminu
m', 9.065236638693596), ('ho', 9.065236638693596), ('hola', 9.06523663
8693596), ('homogeneous', 9.065236638693596), ('convention', 9.0652366
38693596), ('airport', 9.065236638693596), ('atlantic', 9.065236638693
596), ('corridor', 9.065236638693596), ('java', 9.065236638693596),
 ('encountering', 9.065236638693596), ('bicycles', 9.065236638693596),
 ('tearing', 9.065236638693596), ('whites', 9.065236638693596), ('vesse
l', 9.065236638693596), ('communicates', 9.065236638693596), ('raps',
9.065236638693596), ('slp', 9.065236638693596), ('randomly', 9.0652366
38693596), ('ralph', 9.065236638693596), ('rainforest', 9.065236638693
596), ('bicultural', 9.065236638693596), ('comforted', 9.0652366386935
96), ('widens', 9.065236638693596), ('johnny', 9.065236638693596), ('a
dvertise', 9.065236638693596), ('rabbit', 9.065236638693596), ('joint
s', 9.065236638693596), ('ship', 9.065236638693596), ('comedians', 9.0
65236638693596), ('smith', 9.065236638693596), ('quietest', 9.06523663
8693596), ('ios', 9.065236638693596), ('affectionate', 9.0652366386935
96), ('enrolls', 9.065236638693596), ('instituted', 9.06523663869359
6), ('engines', 9.065236638693596), ('aig', 9.065236638693596), ('conf
irm', 9.065236638693596), ('refurbished', 9.065236638693596), ('enforc
ement', 9.065236638693596), ('refraction', 9.065236638693596), ('ski
t', 9.065236638693596), ('injustices', 9.065236638693596), ('westsid
e', 9.065236638693596), ('insecurities', 9.065236638693596), ('recruit
ing', 9.065236638693596), ('realistically', 9.065236638693596), ('inst
rumentalists', 9.065236638693596), ('stilts', 9.065236638693596), ('in
teractivity', 9.065236638693596), ('against', 9.065236638693596), ('re
citing', 9.065236638693596), ('interconnected', 9.065236638693596),
 ('interfering', 9.065236638693596), ('atoms', 9.065236638693596), ('in
terviewed', 9.065236638693596), ('slots', 9.065236638693596), ('correl
ations', 9.065236638693596), ('untangle', 9.065236638693596), ('quee
```

n', 9.065236638693596), ('forging', 9.065236638693596), ('arduous', 9.065236638693596), ('custodians', 9.065236638693596), ('fll', 9.065236638693596), ('flock', 9.065236638693596), ('floss', 9.065236638693596), ('cushy', 9.065236638693596), ('flute', 9.065236638693596), ('fog', 9.065236638693596), ('strumming', 9.065236638693596), ('folks', 9.065236638693596), ('assesses', 9.065236638693596), ('fingerprints', 9.065236638693596), ('shortened', 9.065236638693596), ('screening', 9.065236638693596), ('scrapbook', 9.065236638693596), ('fox', 9.065236638693596), ('cubelets', 9.065236638693596), ('scoot', 9.065236638693596), ('exceedingly', 9.065236638693596), ('scents', 9.065236638693596), ('sturdier', 9.065236638693596), ('scaled', 9.065236638693596), ('cyber', 9.065236638693596), ('sensor', 9.065236638693596), ('apprehensive', 9.065236638693596), ('sexual', 9.065236638693596), ('debt', 9.065236638693596), ('shootings', 9.065236638693596), ('virtues', 9.065236638693596), ('expending', 9.065236638693596), ('artsy', 9.065236638693596), ('visibly', 9.065236638693596), ('explosive', 9.065236638693596), ('exterior', 9.065236638693596), ('extraordinarily', 9.065236638693596), ('exuberance', 9.065236638693596), ('dean', 9.065236638693596), ('filtration', 9.065236638693596), ('barn', 9.065236638693596), ('assembling', 9.065236638693596), ('daughters', 9.065236638693596), ('dating', 9.065236638693596), ('exercisers', 9.065236638693596), ('shortages', 9.065236638693596), ('stretchy', 9.065236638693596), ('separating', 9.065236638693596), ('separately', 9.065236638693596), ('vinci', 9.065236638693596), ('storybook', 9.065236638693596), ('crystal', 9.065236638693596), ('summers', 9.065236638693596), ('evaluations', 9.065236638693596), ('grains', 9.065236638693596), ('grammatical', 9.065236638693596), ('warehouse', 9.065236638693596), ('warhol', 9.065236638693596), ('evenly', 9.065236638693596), ('greener', 9.065236638693596), ('sufficiency', 9.065236638693596), ('gritty', 9.065236638693596), ('groan', 9.065236638693596), ('beams', 9.065236638693596), ('guarantees', 9.065236638693596), ('routes', 9.065236638693596), ('gum', 9.065236638693596), ('rio', 9.065236638693596), ('anniversary', 9.065236638693596), ('hairspray', 9.065236638693596), ('sidelines', 9.065236638693596), ('hampered', 9.065236638693596), ('hamster', 9.065236638693596), ('counted', 9.065236638693596), ('rick', 9.065236638693596), ('summative', 9.065236638693596), ('critiques', 9.065236638693596), ('rsp', 9.065236638693596), ('bose', 9.065236638693596), ('subjected', 9.065236638693596), ('gary', 9.065236638693596), ('savers', 9.065236638693596), ('crown', 9.065236638693596), ('gene', 9.065236638693596), ('crowding', 9.065236638693596), ('sanitized', 9.065236638693596), ('generates', 9.065236638693596), ('apparatus', 9.065236638693596), ('samoa', 9.065236638693596), ('ghana', 9.065236638693596), ('walker', 9.065236638693596), ('googly', 9.065236638693596), ('salad', 9.065236638693596), ('apathetic', 9.065236638693596), ('safari', 9.065236638693596), ('globalization', 9.065236638693596), ('glory', 9.065236638693596), ('walmart', 9.065236638693596), ('glove', 9.065236638693596), ('goldberg', 9.065236638693596), ('ruled', 9.065236638693596), ('rube', 9.065236638693596), ('smoothie', 9.065236638693596), ('embedding', 9.065236638693596), ('collaborated', 9.065236638693596), ('specializing', 9.065236638693596), ('molds', 9.065236638693596), ('mommy', 9.065236638693596), ('cambridge', 9.065236638693596), ('sow', 9.065236638693596), ('dragging', 9.065236638693596), ('calculated', 9.065236638693596), ('wrestle', 9.065236638693596), ('digest', 9.065236638693596), ('pipes', 9.065236638693596), ('pinnies', 9.065236638693596), ('uil', 9.065236638693596), ('piggyp', 9.065236638693596), ('buzzes', 9.065236638693596), ('buzzer', 9.065236638693596), ('tragedies', 9.065236638693596), ('380', 9.065236638693596), ('aunt', 9.065236638693596), ('dork', 9.065236638693596), ('murder', 9.065236638693596), ('phys', 9.065236638693596), ('busily', 9.065236638693596), ('30am', 9.065236638693596), ('photographic', 9.065236638693596), ('modernize', 9.065236638693596), ('moderately', 9.065236638693596), ('tourism', 9.065236638693596), ('abl', 9.065236638693596)

6), ('metric', 9.065236638693596), ('snake', 9.065236638693596), ('portal', 9.065236638693596), ('midwestern', 9.065236638693596), ('abreast', 9.065236638693596), ('tongues', 9.065236638693596), ('attributed', 9.065236638693596), ('stash', 9.065236638693596), ('polygons', 9.065236638693596), ('capstone', 9.065236638693596), ('mimics', 9.065236638693596), ('mockingbird', 9.065236638693596), ('pokey', 9.065236638693596), ('worldview', 9.065236638693596), ('poke', 9.065236638693596), ('didn', 9.065236638693596), ('mines', 9.065236638693596), ('plugging', 9.065236638693596), ('mirroring', 9.065236638693596), ('miscellaneous', 9.065236638693596), ('950', 9.065236638693596), ('misconception', 9.065236638693596), ('nail', 9.065236638693596), ('nanakuli', 9.065236638693596), ('dicamillo', 9.065236638693596), ('diminishing', 9.065236638693596), ('patrons', 9.065236638693596), ('paterson', 9.065236638693596), ('ollie', 9.065236638693596), ('olympians', 9.065236638693596), ('operational', 9.065236638693596), ('15th', 9.065236638693596), ('opinionated', 9.065236638693596), ('opponents', 9.065236638693596), ('opportunities', 9.065236638693596), ('trimester', 9.065236638693596), ('diversities', 9.065236638693596), ('spines', 9.065236638693596), ('1300', 9.065236638693596), ('parade', 9.065236638693596), ('otterbox', 9.065236638693596), ('ziploc', 9.065236638693596), ('outings', 9.065236638693596), ('troublesome', 9.065236638693596), ('pamphlets', 9.065236638693596), ('stacks', 9.065236638693596), ('displacement', 9.065236638693596), ('stabilize', 9.065236638693596), ('oz', 9.065236638693596), ('patterson', 9.065236638693596), ('paws', 9.065236638693596), ('1999', 9.065236638693596), ('briefly', 9.065236638693596), ('philosophies', 9.065236638693596), ('transit', 9.065236638693596), ('pervasive', 9.065236638693596), ('pertains', 9.065236638693596), ('twisting', 9.065236638693596), ('personalizing', 9.065236638693596), ('dodgeball', 9.065236638693596), ('nest', 9.065236638693596), ('networks', 9.065236638693596), ('spelled', 9.065236638693596), ('nobel', 9.065236638693596), ('spheres', 9.065236638693596), ('noble', 9.065236638693596), ('nominees', 9.065236638693596), ('nonstop', 9.065236638693596), ('booked', 9.065236638693596), ('disappeared', 9.065236638693596), ('triangle', 9.065236638693596), ('pediatric', 9.065236638693596), ('nots', 9.065236638693596), ('tribal', 9.065236638693596), ('nuances', 9.065236638693596), ('positioning', 9.065236638693596), ('michelle', 9.065236638693596), ('dwindled', 9.065236638693596), ('activists', 9.065236638693596), ('vermont', 9.065236638693596), ('eip', 9.065236638693596), ('egyptian', 9.065236638693596), ('birmingham', 9.065236638693596), ('launcher', 9.065236638693596), ('adams', 9.065236638693596), ('prosperous', 9.065236638693596), ('launchers', 9.065236638693596), ('actor', 9.065236638693596), ('socioeconomics', 9.065236638693596), ('propped', 9.065236638693596), ('landfills', 9.065236638693596), ('attest', 9.065236638693596), ('proposals', 9.065236638693596), ('lawn', 9.065236638693596), ('proportions', 9.065236638693596), ('editors', 9.065236638693596), ('blacktop', 9.065236638693596), ('witch', 9.065236638693596), ('activates', 9.065236638693596), ('dysfunction', 9.065236638693596), ('editor', 9.065236638693596), ('laos', 9.065236638693596), ('deteriorating', 9.065236638693596), ('legends', 9.065236638693596), ('pulse', 9.065236638693596), ('verse', 9.065236638693596), ('keepers', 9.065236638693596), ('desiring', 9.065236638693596), ('kickoff', 9.065236638693596), ('purifier', 9.065236638693596), ('killing', 9.065236638693596), ('punishment', 9.065236638693596), ('admitted', 9.065236638693596), ('punching', 9.065236638693596), ('kinect', 9.065236638693596), ('admired', 9.065236638693596), ('versa', 9.065236638693596), ('clifford', 9.065236638693596), ('kings', 9.065236638693596), ('windmill', 9.065236638693596), ('clarification', 9.065236638693596), ('knocked', 9.065236638693596), ('ky', 9.065236638693596), ('laborious', 9.065236638693596), ('clamoring', 9.065236638693596), ('circular', 9.065236638693596), ('ladybug', 9.065236638693596), ('charlie', 9.065236638693596), ('pronounced', 9.065236638693596), ('exhibited', 9.065236638693596),

('solos', 9.065236638693596), ('toddler', 9.065236638693596), ('prairie', 9.065236638693596), ('toad', 9.065236638693596), ('wordless', 9.065236638693596), ('centrally', 9.065236638693596), ('prohibitive', 9.065236638693596), ('magnifiers', 9.065236638693596), ('solids', 9.065236638693596), ('census', 9.065236638693596), ('attraction', 9.065236638693596), ('mandate', 9.065236638693596), ('wonderment', 9.065236638693596), ('preservation', 9.065236638693596), ('manifest', 9.065236638693596), ('tipped', 9.065236638693596), ('attracted', 9.065236638693596), ('diamond', 9.065236638693596), ('masses', 9.065236638693596), ('title 1', 9.065236638693596), ('blink', 9.065236638693596), ('lure', 9.065236638693596), ('championships', 9.065236638693596), ('pound', 9.065236638693596), ('achiever', 9.065236638693596), ('lessening', 9.065236638693596), ('bleak', 9.065236638693596), ('verb', 9.065236638693596), ('charges', 9.065236638693596), ('accelerating', 9.065236638693596), ('producer', 9.065236638693596), ('proclaimed', 9.065236638693596), ('longevity', 9.065236638693596), ('earnings', 9.065236638693596), ('devouring', 9.065236638693596), ('chamber', 9.065236638693596), ('meat', 9.065236638693596), ('utensil', 9.065236638693596), ('cents', 9.065236638693596), ('predominant', 9.065236638693596), ('steep', 8.978225261703965), ('destroying', 8.978225261703965), ('decompose', 8.978225261703965), ('stanford', 8.978225261703965), ('designer', 8.978225261703965), ('deployments', 8.978225261703965), ('stix', 8.978225261703965), ('stereotype', 8.978225261703965), ('storyboard', 8.978225261703965), ('stir', 8.978225261703965), ('democracy', 8.978225261703965), ('densely', 8.978225261703965), ('stairs', 8.978225261703965), ('stare', 8.978225261703965), ('storyline', 8.978225261703965), ('developers', 8.978225261703965), ('standardize', 8.978225261703965), ('dies', 8.978225261703965), ('tailoring', 8.978225261703965), ('declined', 8.978225261703965), ('carrier', 8.978225261703965), ('tones', 8.978225261703965), ('toilet', 8.978225261703965), ('casual', 8.978225261703965), ('toddlers', 8.978225261703965), ('catchy', 8.978225261703965), ('tiring', 8.978225261703965), ('centralized', 8.978225261703965), ('tim', 8.978225261703965), ('ceo', 8.978225261703965), ('tic', 8.978225261703965), ('characterized', 8.978225261703965), ('thrived', 8.978225261703965), ('charm', 8.978225261703965), ('threshold', 8.978225261703965), ('threat', 8.978225261703965), ('chavez', 8.978225261703965), ('cheery', 8.978225261703965), ('childcare', 8.978225261703965), ('chrombooks', 8.978225261703965), ('theraputty', 8.978225261703965), ('chrysalis', 8.978225261703965), ('tongue', 8.978225261703965), ('toolkit', 8.978225261703965), ('strap', 8.978225261703965), ('cap', 8.978225261703965), ('bout', 8.978225261703965), ('trucks', 8.978225261703965), ('truck', 8.978225261703965), ('troopers', 8.978225261703965), ('tripled', 8.978225261703965), ('bravest', 8.978225261703965), ('breakouts', 8.978225261703965), ('brightness', 8.978225261703965), ('bucks', 8.978225261703965), ('transparent', 8.978225261703965), ('buried', 8.978225261703965), ('burlington', 8.978225261703965), ('trainings', 8.978225261703965), ('bust', 8.978225261703965), ('trails', 8.978225261703965), ('busting', 8.978225261703965), ('trading', 8.978225261703965), ('cakes', 8.978225261703965), ('calculation', 8.978225261703965), ('towel', 8.978225261703965), ('canceling', 8.978225261703965), ('cloth', 8.978225261703965), ('clouds', 8.978225261703965), ('terrarium', 8.978225261703965), ('colts', 8.978225261703965), ('continuity', 8.978225261703965), ('conveniently', 8.978225261703965), ('cookbooks', 8.978225261703965), ('cooling', 8.978225261703965), ('copper', 8.978225261703965), ('cordless', 8.978225261703965), ('cornell', 8.978225261703965), ('supplemented', 8.978225261703965), ('suites', 8.978225261703965), ('suicide', 8.978225261703965), ('criminal', 8.978225261703965), ('criticism', 8.978225261703965), ('stuttering', 8.978225261703965), ('stunning', 8.978225261703965), ('cte', 8.978225261703965), ('studnets', 8.978225261703965), ('cue', 8.978225261703965), ('strikes', 8.978225261703965), ('damages', 8.978225261703965), ('darling', 8.978225261703965), ('streamlined', 8.97

8225261703965), ('swept', 8.978225261703965), ('swipe', 8.978225261703965), ('symphony', 8.978225261703965), ('complications', 8.978225261703965), ('collapse', 8.978225261703965), ('colonies', 8.978225261703965), ('teks', 8.978225261703965), ('teeter', 8.978225261703965), ('commonalities', 8.978225261703965), ('technologists', 8.978225261703965), ('compartments', 8.978225261703965), ('competed', 8.978225261703965), ('compiling', 8.978225261703965), ('taping', 8.978225261703965), ('synthesis', 8.978225261703965), ('tantrums', 8.978225261703965), ('conceptualize', 8.978225261703965), ('conditioned', 8.978225261703965), ('conductive', 8.978225261703965), ('cone', 8.978225261703965), ('confirm', 8.978225261703965), ('discouraging', 8.978225261703965), ('connectors', 8.978225261703965), ('consensus', 8.978225261703965), ('disconnect', 8.978225261703965), ('zumba', 8.978225261703965), ('squat', 8.978225261703965), ('leer', 8.978225261703965), ('providence', 8.978225261703965), ('lakota', 8.978225261703965), ('latch', 8.978225261703965), ('protocols', 8.978225261703965), ('latter', 8.978225261703965), ('laughed', 8.978225261703965), ('proprioceptive', 8.978225261703965), ('propose', 8.978225261703965), ('lazy', 8.978225261703965), ('leak', 8.978225261703965), ('leather', 8.978225261703965), ('profiles', 8.978225261703965), ('mankind', 8.978225261703965), ('lifted', 8.978225261703965), ('limitation', 8.978225261703965), ('lions', 8.978225261703965), ('logistics', 8.978225261703965), ('lookout', 8.978225261703965), ('printables', 8.978225261703965), ('principled', 8.978225261703965), ('lunings', 8.978225261703965), ('magically', 8.978225261703965), ('mall', 8.978225261703965), ('manager', 8.978225261703965), ('ladders', 8.978225261703965), ('psychologist', 8.978225261703965), ('knex', 8.978225261703965), ('pullout', 8.978225261703965), ('interference', 8.978225261703965), ('internship', 8.978225261703965), ('reassured', 8.978225261703965), ('intricacies', 8.978225261703965), ('inundated', 8.978225261703965), ('iran', 8.978225261703965), ('irreplaceable', 8.978225261703965), ('italy', 8.978225261703965), ('iv', 8.978225261703965), ('ivan', 8.978225261703965), ('rapids', 8.978225261703965), ('ranged', 8.978225261703965), ('railroad', 8.978225261703965), ('rockets', 8.978225261703965), ('quo', 8.978225261703965), ('quilt', 8.978225261703965), ('joseph', 8.978225261703965), ('judicial', 8.978225261703965), ('quarters', 8.978225261703965), ('pyramids', 8.978225261703965), ('kahoots', 8.978225261703965), ('keepsakes', 8.978225261703965), ('kevin', 8.978225261703965), ('managers', 8.978225261703965), ('preparations', 8.978225261703965), ('discrimination', 8.978225261703965), ('nicest', 8.978225261703965), ('pictorial', 8.978225261703965), ('phrasing', 8.978225261703965), ('mustangs', 8.978225261703965), ('photographer', 8.978225261703965), ('naming', 8.978225261703965), ('narrators', 8.978225261703965), ('peter', 8.978225261703965), ('necklace', 8.978225261703965), ('nerves', 8.978225261703965), ('performs', 8.978225261703965), ('performer', 8.978225261703965), ('pendulum', 8.978225261703965), ('masking', 8.978225261703965), ('notices', 8.978225261703965), ('notorious', 8.978225261703965), ('pbs', 8.978225261703965), ('paychecks', 8.978225261703965), ('opposing', 8.978225261703965), ('orientated', 8.978225261703965), ('parameters', 8.978225261703965), ('panhandle', 8.978225261703965), ('outlooks', 8.978225261703965), ('outnumber', 8.978225261703965), ('oversized', 8.978225261703965), ('multiculturalism', 8.978225261703965), ('pie', 8.978225261703965), ('mt', 8.978225261703965), ('person', 8.978225261703965), ('mater', 8.978225261703965), ('matisse', 8.978225261703965), ('mayor', 8.978225261703965), ('ppcd', 8.978225261703965), ('powerfully', 8.978225261703965), ('medal', 8.978225261703965), ('potty', 8.978225261703965), ('postures', 8.978225261703965), ('memorial', 8.978225261703965), ('pops', 8.978225261703965), ('migrate', 8.978225261703965), ('migration', 8.978225261703965), ('poly', 8.978225261703965), ('poetic', 8.978225261703965), ('modal', 8.978225261703965), ('playback', 8.978225261703965), ('planks', 8.978225261703965), ('monkeys', 8.978225261703965), ('monopoly', 8.978225261703965), ('moo

dy', 8.978225261703965), ('moore', 8.978225261703965), ('pirates', 8.978225261703965), ('pine', 8.978225261703965), ('intercity', 8.978225261703965), ('insurance', 8.978225261703965), ('inhibits', 8.978225261703965), ('shield', 8.978225261703965), ('entrepreneurial', 8.978225261703965), ('sis', 8.978225261703965), ('envisioned', 8.978225261703965), ('episode', 8.978225261703965), ('epitome', 8.978225261703965), ('simplified', 8.978225261703965), ('errands', 8.978225261703965), ('shuffle', 8.978225261703965), ('shredder', 8.978225261703965), ('showcases', 8.978225261703965), ('expects', 8.978225261703965), ('sherman', 8.978225261703965), ('inhabit', 8.978225261703965), ('fade', 8.978225261703965), ('falcons', 8.978225261703965), ('farmer', 8.978225261703965), ('farsi', 8.978225261703965), ('setters', 8.978225261703965), ('feathers', 8.978225261703965), ('feats', 8.978225261703965), ('figuratively', 8.978225261703965), ('filtered', 8.978225261703965), ('finisher', 8.978225261703965), ('flexibly', 8.978225261703965), ('entrepreneur', 8.978225261703965), ('skyrocket', 8.978225261703965), ('slam', 8.978225261703965), ('slice', 8.978225261703965), ('disinfectant', 8.978225261703965), ('spoon', 8.978225261703965), ('dixie', 8.978225261703965), ('docking', 8.978225261703965), ('doe', 8.978225261703965), ('dollhouse', 8.978225261703965), ('dolphins', 8.978225261703965), ('downfall', 8.978225261703965), ('downtime', 8.978225261703965), ('drew', 8.978225261703965), ('soup', 8.978225261703965), ('dslr', 8.978225261703965), ('soprano', 8.978225261703965), ('ect', 8.978225261703965), ('soldiers', 8.978225261703965), ('soldering', 8.978225261703965), ('editions', 8.978225261703965), ('electrifying', 8.978225261703965), ('snug', 8.978225261703965), ('snakes', 8.978225261703965), ('smarts', 8.978225261703965), ('slowed', 8.978225261703965), ('slot', 8.978225261703965), ('fluffy', 8.978225261703965), ('securing', 8.978225261703965), ('fork', 8.978225261703965), ('revere', 8.978225261703965), ('healthiest', 8.978225261703965), ('hears', 8.978225261703965), ('heidi', 8.978225261703965), ('restrict', 8.978225261703965), ('resting', 8.978225261703965), ('hiking', 8.978225261703965), ('reserve', 8.978225261703965), ('hoot', 8.978225261703965), ('hoped', 8.978225261703965), ('hormones', 8.978225261703965), ('humbling', 8.978225261703965), ('rented', 8.978225261703965), ('idioms', 8.978225261703965), ('remembers', 8.978225261703965), ('relive', 8.978225261703965), ('inactivity', 8.978225261703965), ('incapable', 8.978225261703965), ('indescribable', 8.978225261703965), ('ineffective', 8.978225261703965), ('reggio', 8.978225261703965), ('influencing', 8.978225261703965), ('infographics', 8.978225261703965), ('informing', 8.978225261703965), ('retire', 8.978225261703965), ('revitalize', 8.978225261703965), ('fortitude', 8.978225261703965), ('hatched', 8.978225261703965), ('france', 8.978225261703965), ('freed', 8.978225261703965), ('fro', 8.978225261703965), ('fulfills', 8.978225261703965), ('gamer', 8.978225261703965), ('gamut', 8.978225261703965), ('satisfies', 8.978225261703965), ('generalization', 8.978225261703965), ('salinas', 8.978225261703965), ('roughest', 8.978225261703965), ('graham', 8.978225261703965), ('roping', 8.978225261703965), ('rollers', 8.978225261703965), ('rochester', 8.978225261703965), ('grumbling', 8.978225261703965), ('ham', 8.978225261703965), ('rickety', 8.978225261703965), ('richly', 8.978225261703965), ('richard', 8.978225261703965), ('hangouts', 8.978225261703965), ('hangs', 8.978225261703965), ('rhode', 8.978225261703965), ('rhetoric', 8.978225261703965), ('boss', 8.978225261703965), ('slashed', 8.978225261703965), ('underestimate', 8.978225261703965), ('belts', 8.978225261703965), ('youths', 8.978225261703965), ('vernier', 8.978225261703965), ('adversely', 8.978225261703965), ('billion', 8.978225261703965), ('apron', 8.978225261703965), ('bernardino', 8.978225261703965), ('acknowledging', 8.978225261703965), ('weary', 8.978225261703965), ('backpatter', 8.978225261703965), ('wiki', 8.978225261703965), ('asus', 8.978225261703965), ('yards', 8.978225261703965), ('affection', 8.978225261703965), ('backing', 8.978225261703965), ('alphabetter', 8.978225261703965), ('affirmatio

n', 8.978225261703965), ('veterans', 8.978225261703965), ('useless', 8.978225261703965), ('240', 8.978225261703965), ('auggie', 8.978225261703965), ('weed', 8.978225261703965), ('2020', 8.978225261703965), ('woes', 8.978225261703965), ('wand', 8.978225261703965), ('uninterested', 8.978225261703965), ('baked', 8.978225261703965), ('2005', 8.978225261703965), ('verses', 8.978225261703965), ('aps', 8.978225261703965), ('affiliated', 8.978225261703965), ('blurry', 8.978225261703965), ('yours', 8.978225261703965), ('bilitery', 8.978225261703965), ('blockly', 8.978225261703965), ('banana', 8.978225261703965), ('boisterous', 8.978225261703965), ('airplane', 8.978225261703965), ('ventures', 8.978225261703965), ('asperger', 8.978225261703965), ('asians', 8.978225261703965), ('viola', 8.978225261703965), ('unlocks', 8.978225261703965), ('480', 8.978225261703965), ('apathy', 8.978225261703965), ('venturing', 8.978225261703965), ('walkers', 8.978225261703965), ('uploading', 8.978225261703965), ('viruses', 8.978225261703965), ('welfare', 8.978225261703965), ('biologists', 8.978225261703965), ('zooming', 8.978225261703965), ('ugly', 8.978225261703965), ('uke', 8.978225261703965), ('altering', 8.978225261703965), ('biodiversity', 8.978225261703965), ('2008', 8.978225261703965), ('watershed', 8.978225261703965), ('aba', 8.978225261703965), ('wildcats', 8.978225261703965), ('indigenous', 8.89818255403043), ('indispensable', 8.89818255403043), ('reliability', 8.89818255403043), ('deepens', 8.89818255403043), ('circulating', 8.89818255403043), ('workings', 8.89818255403043), ('individualizing', 8.89818255403043), ('mediocre', 8.89818255403043), ('accommodated', 8.89818255403043), ('acclimate', 8.89818255403043), ('cleanup', 8.89818255403043), ('upside', 8.89818255403043), ('measurable', 8.89818255403043), ('wore', 8.89818255403043), ('upwards', 8.89818255403043), ('allergies', 8.89818255403043), ('classifying', 8.89818255403043), ('precisely', 8.89818255403043), ('meantime', 8.89818255403043), ('precocious', 8.89818255403043), ('inattentive', 8.89818255403043), ('rele ntless', 8.89818255403043), ('weeds', 8.89818255403043), ('medals', 8.89818255403043), ('theatrical', 8.89818255403043), ('praying', 8.89818255403043), ('potting', 8.89818255403043), ('stranger', 8.89818255403043), ('medically', 8.89818255403043), ('theodore', 8.89818255403043), ('chimes', 8.89818255403043), ('defense', 8.89818255403043), ('polar', 8.89818255403043), ('minimized', 8.89818255403043), ('hollywood', 8.89818255403043), ('pltw', 8.89818255403043), ('misbehavior', 8.89818255403043), ('aac', 8.89818255403043), ('pleasures', 8.89818255403043), ('hints', 8.89818255403043), ('channeled', 8.89818255403043), ('highlighted', 8.89818255403043), ('diminish', 8.89818255403043), ('playdoh', 8.89818255403043), ('diminishes', 8.89818255403043), ('modifying', 8.89818255403043), ('stamping', 8.89818255403043), ('amenities', 8.89818255403043), ('tiered', 8.89818255403043), ('staffed', 8.89818255403043), ('heartwarming', 8.89818255403043), ('tightly', 8.89818255403043), ('5k', 8.89818255403043), ('undergo', 8.89818255403043), ('validate', 8.89818255403043), ('discreetly', 8.89818255403043), ('discusses', 8.89818255403043), ('timelines', 8.89818255403043), ('differentiates', 8.89818255403043), ('resemble', 8.89818255403043), ('underwear', 8.89818255403043), ('alpha', 8.89818255403043), ('remedy', 8.89818255403043), ('illustrates', 8.89818255403043), ('delzer', 8.89818255403043), ('removable', 8.89818255403043), ('dentist', 8.89818255403043), ('removes', 8.89818255403043), ('ie', 8.89818255403043), ('thief', 8.89818255403043), ('identical', 8.89818255403043), ('allowance', 8.89818255403043), ('positioned', 8.89818255403043), ('positional', 8.89818255403043), ('metallophones', 8.89818255403043), ('methodology', 8.89818255403043), ('humidity', 8.89818255403043), ('thirsting', 8.89818255403043), ('hum', 8.89818255403043), ('reproducible', 8.89818255403043), ('mike', 8.89818255403043), ('determines', 8.89818255403043), ('checkers', 8.89818255403043), ('aboard', 8.89818255403043), ('horn', 8.89818255403043), ('devastation', 8.89818255403043), ('chatting', 8.89818255403043), ('weighing', 8.89818255403043), ('cps', 8.89818255403043), ('dazzl

ing', 8.89818255403043), ('tapped', 8.89818255403043), ('quaint', 8.89818255403043), ('coral', 8.89818255403043), ('ld', 8.89818255403043), ('wit', 8.89818255403043), ('jv', 8.89818255403043), ('junie', 8.89818255403043), ('beavers', 8.89818255403043), ('judy', 8.89818255403043), ('conform', 8.89818255403043), ('configurations', 8.89818255403043), ('confidential', 8.89818255403043), ('confidant', 8.89818255403043), ('learnings', 8.89818255403043), ('superpower', 8.89818255403043), ('adventurers', 8.89818255403043), ('concretely', 8.89818255403043), ('promises', 8.89818255403043), ('widespread', 8.89818255403043), ('prom', 8.89818255403043), ('summit', 8.89818255403043), ('achieves', 8.89818255403043), ('unthinkable', 8.89818255403043), ('lethargic', 8.89818255403043), ('compromised', 8.89818255403043), ('jensen', 8.89818255403043), ('python', 8.89818255403043), ('consisted', 8.89818255403043), ('consoles', 8.89818255403043), ('lagging', 8.89818255403043), ('swahili', 8.89818255403043), ('publisher', 8.89818255403043), ('continuum', 8.89818255403043), ('adjacent', 8.89818255403043), ('pulleys', 8.89818255403043), ('kindly', 8.89818255403043), ('kindling', 8.89818255403043), ('ladybugs', 8.89818255403043), ('williams', 8.89818255403043), ('unmatched', 8.89818255403043), ('unpack', 8.89818255403043), ('addicted', 8.89818255403043), ('belorussia', 8.89818255403043), ('conversion', 8.89818255403043), ('constructively', 8.89818255403043), ('keynote', 8.89818255403043), ('surpassed', 8.89818255403043), ('constructions', 8.89818255403043), ('synergize', 8.89818255403043), ('wiping', 8.89818255403043), ('kenya', 8.89818255403043), ('protagonists', 8.89818255403043), ('protagonist', 8.89818255403043), ('prose', 8.89818255403043), ('jennifer', 8.89818255403043), ('raisers', 8.89818255403043), ('ineed', 8.89818255403043), ('cpr', 8.89818255403043), ('wherein', 8.89818255403043), ('culmination', 8.89818255403043), ('combs', 8.89818255403043), ('cure', 8.89818255403043), ('magformers', 8.89818255403043), ('currents', 8.89818255403043), ('recyclable', 8.89818255403043), ('redesigned', 8.89818255403043), ('accounting', 8.89818255403043), ('strife', 8.89818255403043), ('refuel', 8.89818255403043), ('ingredient', 8.89818255403043), ('wonderland', 8.89818255403043), ('disguised', 8.89818255403043), ('prescribed', 8.89818255403043), ('preschools', 8.89818255403043), ('coincide', 8.89818255403043), ('dancer', 8.89818255403043), ('dangers', 8.89818255403043), ('prepping', 8.89818255403043), ('tents', 8.89818255403043), ('database', 8.89818255403043), ('inflection', 8.89818255403043), ('registered', 8.89818255403043), ('inferencing', 8.89818255403043), ('cuba', 8.89818255403043), ('stuffy', 8.89818255403043), ('prioritize', 8.89818255403043), ('loathe', 8.89818255403043), ('jams', 8.89818255403043), ('aches', 8.89818255403043), ('lifts', 8.89818255403043), ('aerial', 8.89818255403043), ('liking', 8.89818255403043), ('lion', 8.89818255403043), ('crammed', 8.89818255403043), ('crashing', 8.89818255403043), ('crawling', 8.89818255403043), ('succeeded', 8.89818255403043), ('credible', 8.89818255403043), ('invigorate', 8.89818255403043), ('lorax', 8.89818255403043), ('proceeds', 8.89818255403043), ('investigated', 8.89818255403043), ('intro', 8.89818255403043), ('comparable', 8.89818255403043), ('wolves', 8.89818255403043), ('rearranging', 8.89818255403043), ('communicated', 8.89818255403043), ('ct', 8.89818255403043), ('loosing', 8.89818255403043), ('reassurance', 8.89818255403043), ('commodity', 8.89818255403043), ('manipulates', 8.89818255403043), ('dim', 8.89818255403043), ('frustrate', 8.89818255403043), ('dribble', 8.89818255403043), ('nit', 8.89818255403043), ('funniest', 8.89818255403043), ('approval', 8.89818255403043), ('fresno', 8.89818255403043), ('transferable', 8.89818255403043), ('freeing', 8.89818255403043), ('buoyancy', 8.89818255403043), ('notable', 8.89818255403043), ('peg', 8.89818255403043), ('yields', 8.89818255403043), ('socials', 8.89818255403043), ('yo', 8.89818255403043), ('elastic', 8.89818255403043), ('snowy', 8.89818255403043), ('peacefully', 8.89818255403043), ('nourishing', 8.89818255403043), ('elicit', 8.89818255403043), ('scraps', 8.89818255403043), ('screws', 8.89818255403043),

8255403043), ('175', 8.89818255403043), ('forests', 8.89818255403043), ('smelling', 8.89818255403043), ('secretly', 8.89818255403043), ('perceptions', 8.89818255403043), ('fuzzy', 8.89818255403043), ('ventured', 8.89818255403043), ('calkins', 8.89818255403043), ('candidates', 8.89818255403043), ('yearbooks', 8.89818255403043), ('glassware', 8.89818255403043), ('negatives', 8.89818255403043), ('gizmos', 8.89818255403043), ('drought', 8.89818255403043), ('nerve', 8.89818255403043), ('drumsticks', 8.89818255403043), ('persevered', 8.89818255403043), ('netbooks', 8.89818255403043), ('gingerbread', 8.89818255403043), ('perfecting', 8.89818255403043), ('calendars', 8.89818255403043), ('sam', 8.89818255403043), ('nfl', 8.89818255403043), ('cad', 8.89818255403043), ('applaud', 8.89818255403043), ('saxophones', 8.89818255403043), ('garageband', 8.89818255403043), ('bustle', 8.89818255403043), ('eaters', 8.89818255403043), ('solver', 8.89818255403043), ('nys', 8.89818255403043), ('employs', 8.89818255403043), ('attained', 8.89818255403043), ('booster', 8.89818255403043), ('facilitators', 8.89818255403043), ('panthers', 8.89818255403043), ('pan', 8.89818255403043), ('extravaganza', 8.89818255403043), ('boomwhackers', 8.89818255403043), ('shark', 8.89818255403043), ('1100', 8.89818255403043), ('explorer', 8.89818255403043), ('palpable', 8.89818255403043), ('ethical', 8.89818255403043), ('pal', 8.89818255403043), ('envy', 8.89818255403043), ('pakistani', 8.89818255403043), ('overhaul', 8.89818255403043), ('evaluated', 8.89818255403043), ('asleep', 8.89818255403043), ('shutting', 8.89818255403043), ('evolves', 8.89818255403043), ('excels', 8.89818255403043), ('owe', 8.89818255403043), ('owns', 8.89818255403043), ('tunnels', 8.89818255403043), ('outline', 8.89818255403043), ('environment', 8.89818255403043), ('observational', 8.89818255403043), ('yousafzai', 8.89818255403043), ('sel', 8.89818255403043), ('encounters', 8.89818255403043), ('offs', 8.89818255403043), ('offset', 8.89818255403043), ('endearing', 8.89818255403043), ('partly', 8.89818255403043), ('vocab', 8.89818255403043), ('slave', 8.89818255403043), ('atlases', 8.89818255403043), ('onward', 8.89818255403043), ('filmed', 8.89818255403043), ('vice', 8.89818255403043), ('separation', 8.89818255403043), ('sequences', 8.89818255403043), ('engine', 8.89818255403043), ('enlightened', 8.89818255403043), ('settling', 8.89818255403043), ('vga', 8.89818255403043), ('entail', 8.89818255403043), ('123', 8.89818255403043), ('bracket', 8.89818255403043), ('entrusted', 8.89818255403043), ('sacred', 8.89818255403043), ('contrary', 8.89818255403043), ('necessitates', 8.89818255403043), ('yeah', 8.89818255403043), ('carrots', 8.89818255403043), ('handmade', 8.89818255403043), ('carriers', 8.89818255403043), ('nashville', 8.89818255403043), ('430', 8.89818255403043), ('nate', 8.89818255403043), ('harbor', 8.89818255403043), ('carnival', 8.89818255403043), ('anchorage', 8.89818255403043), ('navajo', 8.89818255403043), ('carnegie', 8.89818255403043), ('auxiliary', 8.89818255403043), ('graciously', 8.89818255403043), ('spoons', 8.89818255403043), ('gorgeous', 8.89818255403043), ('rome', 8.89818255403043), ('tlc', 8.89818255403043), ('anne', 8.89818255403043), ('spine', 8.89818255403043), ('automaticity', 8.89818255403043), ('persuade', 8.89818255403043), ('guru', 8.89818255403043), ('divider', 8.89818255403043), ('roaring', 8.89818255403043), ('280', 8.89818255403043), ('tomatoes', 8.89818255403043), ('roman', 8.89818255403043), ('annie', 8.89818255403043), ('uncontrollable', 8.89818255403043), ('specifics', 8.89818255403043), ('hamilton', 8.89818255403043), ('grey', 8.89818255403043), ('greenville', 8.89818255403043), ('ruby', 8.89818255403043), ('disservice', 8.89818255403043), ('announce', 8.89818255403043), ('hartford', 8.89818255403043), ('pirate', 8.89818255403043), ('tossing', 8.89818255403043), ('cats', 8.89818255403043), ('glockenspiels', 8.89818255403043), ('runny', 8.89818255403043), ('haves', 8.89818255403043), ('spreads', 8.89818255403043), ('goalie', 8.89818255403043), ('unattainable', 8.89818255403043), ('fillers', 8.824074581876708), ('babysitting', 8.824074581876708), ('gyms', 8.824074581876708), ('cotton', 8.824074581876708),

('untapped', 8.824074581876708), ('quoted', 8.824074581876708), ('skel
etal', 8.824074581876708), ('analog', 8.824074581876708), ('az', 8.824
074581876708), ('shortcomings', 8.824074581876708), ('resonate', 8.824
074581876708), ('serviced', 8.824074581876708), ('heartbeat', 8.824074
581876708), ('hazards', 8.824074581876708), ('excelled', 8.82407458187
6708), ('controllers', 8.824074581876708), ('ramp', 8.82407458187670
8), ('jacksonville', 8.824074581876708), ('backdrop', 8.82407458187670
8), ('died', 8.824074581876708), ('ranch', 8.824074581876708), ('rampa
nt', 8.824074581876708), ('firefighters', 8.824074581876708), ('sugges
ting', 8.824074581876708), ('coworkers', 8.824074581876708), ('hirin
g', 8.824074581876708), ('sensation', 8.824074581876708), ('homeroom
s', 8.824074581876708), ('spreadsheet', 8.824074581876708), ('jellyfis
h', 8.824074581876708), ('gun', 8.824074581876708), ('returns', 8.8240
74581876708), ('rage', 8.824074581876708), ('retains', 8.8240745818767
08), ('amidst', 8.824074581876708), ('sierra', 8.824074581876708), ('h
esitation', 8.824074581876708), ('etiquette', 8.824074581876708), ('fa
cebook', 8.824074581876708), ('ashamed', 8.824074581876708), ('copie
d', 8.824074581876708), ('equalizer', 8.824074581876708), ('dispense
r', 8.824074581876708), ('unrealistic', 8.824074581876708), ('heatin
g', 8.824074581876708), ('kindhearted', 8.824074581876708), ('arthur',
8.824074581876708), ('supreme', 8.824074581876708), ('shells', 8.82407
4581876708), ('sporadic', 8.824074581876708), ('stagnant', 8.824074581
876708), ('kate', 8.824074581876708), ('kayla', 8.824074581876708),
('expansive', 8.824074581876708), ('purely', 8.824074581876708), ('dic
tation', 8.824074581876708), ('epidemic', 8.824074581876708), ('exhibi
ts', 8.824074581876708), ('arranging', 8.824074581876708), ('enthusias
ts', 8.824074581876708), ('vibe', 8.824074581876708), ('himself', 8.82
4074581876708), ('killed', 8.824074581876708), ('supervised', 8.824074
581876708), ('korea', 8.824074581876708), ('angels', 8.82407458187670
8), ('hated', 8.824074581876708), ('supplementary', 8.82407458187670
8), ('exhilarating', 8.824074581876708), ('converting', 8.824074581876
708), ('kicks', 8.824074581876708), ('cork', 8.824074581876708), ('qui
ets', 8.824074581876708), ('iphones', 8.824074581876708), ('whoever',
8.824074581876708), ('sbac', 8.824074581876708), ('ruin', 8.8240745818
76708), ('stifled', 8.824074581876708), ('wages', 8.824074581876708),
('genders', 8.824074581876708), ('gems', 8.824074581876708), ('sauce
r', 8.824074581876708), ('dysgraphia', 8.824074581876708), ('inequitie
s', 8.824074581876708), ('gandhi', 8.824074581876708), ('rubrics', 8.8
24074581876708), ('socratic', 8.824074581876708), ('welding', 8.82407
4581876708), ('velocity', 8.824074581876708), ('doubles', 8.8240745818
76708), ('anti', 8.824074581876708), ('hustle', 8.824074581876708),
('aural', 8.824074581876708), ('approached', 8.824074581876708), ('wel
lbeing', 8.824074581876708), ('springfield', 8.824074581876708), ('rem
odeled', 8.824074581876708), ('banking', 8.824074581876708), ('sandy',
8.824074581876708), ('indicators', 8.824074581876708), ('southside',
8.824074581876708), ('audiobook', 8.824074581876708), ('relied', 8.824
074581876708), ('balm', 8.824074581876708), ('gladly', 8.8240745818767
08), ('soviet', 8.824074581876708), ('defensive', 8.824074581876708),
('deficiency', 8.824074581876708), ('decor', 8.824074581876708), ('wed
nesdays', 8.824074581876708), ('implements', 8.824074581876708), ('imp
edes', 8.824074581876708), ('sage', 8.824074581876708), ('inconsisten
t', 8.824074581876708), ('sail', 8.824074581876708), ('saint', 8.82407
4581876708), ('giggling', 8.824074581876708), ('stove', 8.824074581876
708), ('rumbling', 8.824074581876708), ('husband', 8.824074581876708),
('inputs', 8.824074581876708), ('arctic', 8.824074581876708), ('crime
s', 8.824074581876708), ('upbringing', 8.824074581876708), ('intramura
l', 8.824074581876708), ('secretary', 8.824074581876708), ('affectiv
e', 8.824074581876708), ('warranty', 8.824074581876708), ('foil', 8.82
4074581876708), ('florence', 8.824074581876708), ('slows', 8.824074581
876708), ('whistles', 8.824074581876708), ('floods', 8.82407458187670
8), ('inquisitiveness', 8.824074581876708), ('encompassing', 8.8240745

81876708), ('aesthetics', 8.824074581876708), ('readable', 8.824074581876708), ('variable', 8.824074581876708), ('sei', 8.824074581876708), ('vested', 8.824074581876708), ('diaries', 8.824074581876708), ('suck', 8.824074581876708), ('sudan', 8.824074581876708), ('upbringings', 8.824074581876708), ('interviewing', 8.824074581876708), ('smelly', 8.824074581876708), ('documentary', 8.824074581876708), ('replicas', 8.824074581876708), ('cuties', 8.824074581876708), ('recommendation', 8.824074581876708), ('designate', 8.824074581876708), ('verify', 8.824074581876708), ('agencies', 8.824074581876708), ('elapsed', 8.824074581876708), ('afloat', 8.824074581876708), ('interior', 8.824074581876708), ('despair', 8.824074581876708), ('voluntary', 8.824074581876708), ('fourths', 8.824074581876708), ('eleventh', 8.824074581876708), ('housekeeping', 8.824074581876708), ('groans', 8.824074581876708), ('screaming', 8.824074581876708), ('fought', 8.824074581876708), ('groovin', 8.824074581876708), ('sd', 8.824074581876708), ('revamp', 8.824074581876708), ('vigor', 8.824074581876708), ('monotonous', 8.824074581876708), ('booklets', 8.824074581876708), ('underline', 8.824074581876708), ('thumbs', 8.824074581876708), ('thumb', 8.824074581876708), ('technicians', 8.824074581876708), ('looms', 8.824074581876708), ('channels', 8.824074581876708), ('loops', 8.824074581876708), ('bilinguals', 8.824074581876708), ('charities', 8.824074581876708), ('miraculous', 8.824074581876708), ('podium', 8.824074581876708), ('bolts', 8.824074581876708), ('bite', 8.824074581876708), ('bravery', 8.824074581876708), ('nominated', 8.824074581876708), ('prince', 8.824074581876708), ('peoples', 8.824074581876708), ('world', 8.824074581876708), ('underground', 8.824074581876708), ('processed', 8.824074581876708), ('monkey', 8.824074581876708), ('troubling', 8.824074581876708), ('monuments', 8.824074581876708), ('yeats', 8.824074581876708), ('comply', 8.824074581876708), ('zippers', 8.824074581876708), ('wraparound', 8.824074581876708), ('tilt', 8.824074581876708), ('trombone', 8.824074581876708), ('boast', 8.824074581876708), ('triumphs', 8.824074581876708), ('linoleum', 8.824074581876708), ('tiggly', 8.824074581876708), ('opted', 8.824074581876708), ('loaned', 8.824074581876708), ('mondays', 8.824074581876708), ('molecular', 8.824074581876708), ('openly', 8.824074581876708), ('chasing', 8.824074581876708), ('cello', 8.824074581876708), ('omaha', 8.824074581876708), ('bulldogs', 8.824074581876708), ('bulldog', 8.824074581876708), ('pregnancy', 8.824074581876708), ('2003', 8.824074581876708), ('posses', 8.824074581876708), ('1800', 8.824074581876708), ('mason', 8.824074581876708), ('numerals', 8.824074581876708), ('1600', 8.824074581876708), ('therefor', 8.824074581876708), ('texting', 8.824074581876708), ('brochure', 8.824074581876708), ('broom', 8.824074581876708), ('brushing', 8.824074581876708), ('carle', 8.824074581876708), ('postural', 8.824074581876708), ('bone', 8.824074581876708), ('marketable', 8.824074581876708), ('pedagogy', 8.824074581876708), ('prerequisite', 8.824074581876708), ('abled', 8.824074581876708), ('bonded', 8.824074581876708), ('magnatiles', 8.824074581876708), ('mills', 8.824074581876708), ('preview', 8.824074581876708), ('bread', 8.824074581876708), ('colonial', 8.824074581876708), ('breadth', 8.824074581876708), ('ol', 8.824074581876708), ('nouns', 8.824074581876708), ('poorer', 8.824074581876708), ('notoriously', 8.824074581876708), ('mics', 8.824074581876708), ('2006', 8.824074581876708), ('bully', 8.824074581876708), ('absenteeism', 8.824074581876708), ('tenor', 8.824074581876708), ('centerpiece', 8.824074581876708), ('patrol', 8.824074581876708), ('tulsa', 8.824074581876708), ('catapult', 8.824074581876708), ('outspoken', 8.824074581876708), ('consumables', 8.824074581876708), ('toothbrush', 8.824074581876708), ('booth', 8.824074581876708), ('neil', 8.824074581876708), ('legal', 8.824074581876708), ('leftovers', 8.824074581876708), ('bother', 8.824074581876708), ('blossomed', 8.824074581876708), ('adapters', 8.824074581876708), ('overview', 8.824074581876708), ('confronted', 8.824074581876708), ('canyon', 8.824074581876708), ('carve', 8.824074581876708), ('tots', 8.824074581876708), ('myt

h', 8.824074581876708), ('unanimously', 8.824074581876708), ('carefre
e', 8.824074581876708), ('wiped', 8.824074581876708), ('carving', 8.82
4074581876708), ('symbolism', 8.824074581876708), ('overarching', 8.82
4074581876708), ('tubano', 8.824074581876708), ('provoke', 8.824074581
876708), ('multiplying', 8.824074581876708), ('networking', 8.82407458
1876708), ('comprises', 8.824074581876708), ('leverage', 8.82407458187
6708), ('actives', 8.824074581876708), ('unbearable', 8.82407458187670
8), ('wv', 8.824074581876708), ('zoned', 8.824074581876708), ('permit
s', 8.824074581876708), ('peru', 8.824074581876708), ('nap', 8.8240745
81876708), ('unbelievably', 8.824074581876708), ('permitted', 8.824074
581876708), ('caps', 8.755081710389756), ('weathering', 8.755081710389
756), ('musicality', 8.755081710389756), ('elected', 8.75508171038975
6), ('photosynthesis', 8.755081710389756), ('desmos', 8.75508171038975
6), ('badges', 8.755081710389756), ('dramas', 8.755081710389756), ('mu
slim', 8.755081710389756), ('howard', 8.755081710389756), ('divorce',
8.755081710389756), ('alongs', 8.755081710389756), ('2007', 8.75508171
0389756), ('bumpy', 8.755081710389756), ('destiny', 8.75508171038975
6), ('castle', 8.755081710389756), ('rita', 8.755081710389756), ('humb
led', 8.755081710389756), ('disheartening', 8.755081710389756), ('bull
ied', 8.755081710389756), ('memorized', 8.755081710389756), ('melodi
c', 8.755081710389756), ('romeo', 8.755081710389756), ('graphical', 8.
755081710389756), ('cincinnati', 8.755081710389756), ('budgetary', 8.7
55081710389756), ('foresee', 8.755081710389756), ('memoir', 8.75508171
0389756), ('abstractly', 8.755081710389756), ('improvisation', 8.75508
1710389756), ('formally', 8.755081710389756), ('navigation', 8.7550817
10389756), ('nursing', 8.755081710389756), ('chronically', 8.755081710
389756), ('balloon', 8.755081710389756), ('possessions', 8.75508171038
9756), ('warning', 8.755081710389756), ('transplant', 8.75508171038975
6), ('rounds', 8.755081710389756), ('demanded', 8.755081710389756),
('caretakers', 8.755081710389756), ('rockstars', 8.755081710389756),
('scratched', 8.755081710389756), ('nancy', 8.755081710389756), ('mer
e', 8.755081710389756), ('snapshot', 8.755081710389756), ('choirs', 8.
755081710389756), ('peninsula', 8.755081710389756), ('caregiver', 8.75
5081710389756), ('scotch', 8.755081710389756), ('hydroponic', 8.755081
710389756), ('chinatown', 8.755081710389756), ('verbalize', 8.75508171
0389756), ('pianos', 8.755081710389756), ('resumes', 8.75508171038975
6), ('tragic', 8.755081710389756), ('butler', 8.755081710389756), ('cl
ap', 8.755081710389756), ('herself', 8.755081710389756), ('pillars',
8.755081710389756), ('gardner', 8.755081710389756), ('umbrella', 8.755
081710389756), ('disagree', 8.755081710389756), ('butter', 8.755081710
389756), ('planner', 8.755081710389756), ('gearing', 8.75508171038975
6), ('disappointment', 8.755081710389756), ('dynamath', 8.755081710389
756), ('nepali', 8.755081710389756), ('caution', 8.755081710389756),
('picks', 8.755081710389756), ('trades', 8.755081710389756), ('revisio
ns', 8.755081710389756), ('heated', 8.755081710389756), ('cam', 8.7550
81710389756), ('59', 8.755081710389756), ('sprawled', 8.75508171038975
6), ('yell', 8.755081710389756), ('validated', 8.755081710389756), ('m
osaics', 8.755081710389756), ('publicly', 8.755081710389756), ('timele
ss', 8.755081710389756), ('samoan', 8.755081710389756), ('appalachia',
8.755081710389756), ('headstart', 8.755081710389756), ('module', 8.755
081710389756), ('niche', 8.755081710389756), ('harnessing', 8.75508171
0389756), ('diminished', 8.755081710389756), ('gusto', 8.7550817103897
56), ('approve', 8.755081710389756), ('polished', 8.755081710389756),
('wearable', 8.755081710389756), ('transcends', 8.755081710389756),
('caliber', 8.755081710389756), ('diagnostic', 8.755081710389756), ('h
onoring', 8.755081710389756), ('hampshire', 8.755081710389756), ('dist
ributing', 8.755081710389756), ('2010', 8.755081710389756), ('fundrais
e', 8.755081710389756), ('365', 8.755081710389756), ('pointed', 8.7550
81710389756), ('mineral', 8.755081710389756), ('distress', 8.755081710
389756), ('handing', 8.755081710389756), ('wastes', 8.75508171038975
6), ('charitable', 8.755081710389756), ('plickers', 8.75508171038975

6), ('multiage', 8.755081710389756), ('hive', 8.755081710389756), ('to
tes', 8.755081710389756), ('disrupted', 8.755081710389756), ('categori
cal', 8.755081710389756), ('diligence', 8.755081710389756), ('boasts',
8.755081710389756), ('scaffolded', 8.755081710389756), ('gamers', 8.75
5081710389756), ('dioramas', 8.755081710389756), ('secrets', 8.7550817
10389756), ('patiently', 8.755081710389756), ('teachings', 8.755081710
389756), ('craze', 8.755081710389756), ('vista', 8.755081710389756),
('parenting', 8.755081710389756), ('orientation', 8.755081710389756),
('rays', 8.755081710389756), ('lining', 8.755081710389756), ('factorie
s', 8.755081710389756), ('cracking', 8.755081710389756), ('bows', 8.75
5081710389756), ('sufficiently', 8.755081710389756), ('shall', 8.75508
1710389756), ('lice', 8.755081710389756), ('profoundly', 8.75508171038
9756), ('cow', 8.755081710389756), ('jar', 8.755081710389756), ('rall
y', 8.755081710389756), ('extrinsic', 8.755081710389756), ('procure',
8.755081710389756), ('teammate', 8.755081710389756), ('levers', 8.7550
81710389756), ('med', 8.755081710389756), ('intertwined', 8.7550817103
89756), ('ffa', 8.755081710389756), ('ted', 8.755081710389756), ('beac
on', 8.755081710389756), ('skillful', 8.755081710389756), ('engulfed',
8.755081710389756), ('ferguson', 8.755081710389756), ('fence', 8.75508
1710389756), ('partially', 8.755081710389756), ('feminine', 8.75508171
0389756), ('critiquing', 8.755081710389756), ('fearlessly', 8.75508171
0389756), ('subsequently', 8.755081710389756), ('opt', 8.7550817103897
56), ('probes', 8.755081710389756), ('lofty', 8.755081710389756), ('un
wind', 8.755081710389756), ('jean', 8.755081710389756), ('sumdog', 8.7
55081710389756), ('rebuilt', 8.755081710389756), ('proportion', 8.7550
81710389756), ('exemplar', 8.755081710389756), ('surge', 8.75508171038
9756), ('05', 8.755081710389756), ('unprecedented', 8.75508171038975
6), ('kicked', 8.755081710389756), ('kiddies', 8.755081710389756), ('l
apboards', 8.755081710389756), ('viewpoint', 8.755081710389756), ('exc
laim', 8.755081710389756), ('contend', 8.755081710389756), ('pupil',
8.755081710389756), ('exceptions', 8.755081710389756), ('adhere', 8.75
5081710389756), ('shovels', 8.755081710389756), ('controls', 8.7550817
10389756), ('behold', 8.755081710389756), ('knocking', 8.7550817103897
56), ('exhibiting', 8.755081710389756), ('overlook', 8.75508171038975
6), ('radius', 8.755081710389756), ('tuck', 8.755081710389756), ('bo
w', 8.755081710389756), ('lesser', 8.755081710389756), ('trumpet', 8.7
55081710389756), ('tangle', 8.755081710389756), ('extensively', 8.7550
81710389756), ('jet', 8.755081710389756), ('leftover', 8.7550817103897
56), ('lee', 8.755081710389756), ('advantaged', 8.755081710389756),
('acoustic', 8.755081710389756), ('astronaut', 8.755081710389756), ('e
xpresses', 8.755081710389756), ('bounced', 8.755081710389756), ('she
d', 8.755081710389756), ('estimate', 8.755081710389756), ('juliet', 8.
755081710389756), ('astonishing', 8.755081710389756), ('sequential',
8.755081710389756), ('ron', 8.755081710389756), ('flies', 8.7550817103
89756), ('referrals', 8.755081710389756), ('biomedical', 8.75508171038
9756), ('preparedness', 8.755081710389756), ('voiced', 8.7550817103897
56), ('vet', 8.755081710389756), ('slim', 8.755081710389756), ('floori
ng', 8.755081710389756), ('flair', 8.755081710389756), ('vertical', 8.
755081710389756), ('injured', 8.755081710389756), ('tending', 8.755081
710389756), ('clickers', 8.755081710389756), ('prevented', 8.755081710
389756), ('occupation', 8.755081710389756), ('passport', 8.75508171038
9756), ('unfinished', 8.755081710389756), ('selfless', 8.7550817103897
56), ('pastel', 8.755081710389756), ('broadens', 8.755081710389756),
('tender', 8.755081710389756), ('accompaniment', 8.755081710389756),
('refreshed', 8.755081710389756), ('flimsy', 8.755081710389756), ('pre
teens', 8.755081710389756), ('stringed', 8.755081710389756), ('darknes
s', 8.755081710389756), ('clusters', 8.755081710389756), ('securely',
8.755081710389756), ('fights', 8.755081710389756), ('princesses', 8.75
5081710389756), ('clarify', 8.755081710389756), ('meanwhile', 8.755081
710389756), ('preforming', 8.755081710389756), ('interdependent', 8.75
5081710389756), ('alleviating', 8.755081710389756), ('infer', 8.755081

710389756), ('martian', 8.755081710389756), ('reef', 8.755081710389756), ('decomposing', 8.755081710389756), ('slope', 8.755081710389756), ('recreating', 8.755081710389756), ('prey', 8.755081710389756), ('installation', 8.755081710389756), ('intact', 8.755081710389756), ('meager', 8.755081710389756), ('premium', 8.755081710389756), ('incidence', 8.755081710389756), ('bio', 8.755081710389756), ('slogan', 8.755081710389756), ('shell', 8.690543189252185), ('trusted', 8.690543189252185), ('seize', 8.690543189252185), ('silver', 8.690543189252185), ('cape', 8.690543189252185), ('experimented', 8.690543189252185), ('snare', 8.690543189252185), ('brim', 8.690543189252185), ('tuba', 8.690543189252185), ('burma', 8.690543189252185), ('slowing', 8.690543189252185), ('foldable', 8.690543189252185), ('paddles', 8.690543189252185), ('tuned', 8.690543189252185), ('dominoes', 8.690543189252185), ('dont', 8.690543189252185), ('grandchildren', 8.690543189252185), ('smarties', 8.690543189252185), ('forge', 8.690543189252185), ('volcano', 8.690543189252185), ('downloading', 8.690543189252185), ('evolved', 8.690543189252185), ('searches', 8.690543189252185), ('navy', 8.690543189252185), ('gonna', 8.690543189252185), ('obama', 8.690543189252185), ('evidence', 8.690543189252185), ('shops', 8.690543189252185), ('assimilate', 8.690543189252185), ('boots', 8.690543189252185), ('paintbrushes', 8.690543189252185), ('painters', 8.690543189252185), ('snacking', 8.690543189252185), ('spain', 8.690543189252185), ('exhibition', 8.690543189252185), ('volumes', 8.690543189252185), ('canvases', 8.690543189252185), ('perks', 8.690543189252185), ('canon', 8.690543189252185), ('slates', 8.690543189252185), ('visitor', 8.690543189252185), ('organically', 8.690543189252185), ('verge', 8.690543189252185), ('triggers', 8.690543189252185), ('fist', 8.690543189252185), ('verbs', 8.690543189252185), ('eachother', 8.690543189252185), ('frenzy', 8.690543189252185), ('enlarged', 8.690543189252185), ('sld', 8.690543189252185), ('aristotle', 8.690543189252185), ('filtering', 8.690543189252185), ('bye', 8.690543189252185), ('nigeria', 8.690543189252185), ('arguably', 8.690543189252185), ('ebd', 8.690543189252185), ('voting', 8.690543189252185), ('england', 8.690543189252185), ('tribulations', 8.690543189252185), ('solace', 8.690543189252185), ('tricks', 8.690543189252185), ('yummy', 8.690543189252185), ('trainers', 8.690543189252185), ('venue', 8.690543189252185), ('yelling', 8.690543189252185), ('paradigm', 8.690543189252185), ('voluntarily', 8.690543189252185), ('veggie', 8.690543189252185), ('astronomy', 8.690543189252185), ('victory', 8.690543189252185), ('elevated', 8.690543189252185), ('simulated', 8.690543189252185), ('1500', 8.690543189252185), ('soreness', 8.690543189252185), ('soars', 8.690543189252185), ('elderly', 8.690543189252185), ('sorely', 8.690543189252185), ('sore', 8.690543189252185), ('elbow', 8.690543189252185), ('arrivals', 8.690543189252185), ('triumph', 8.690543189252185), ('panels', 8.690543189252185), ('tweezers', 8.690543189252185), ('troubleshoot', 8.690543189252185), ('cage', 8.690543189252185), ('ottomans', 8.690543189252185), ('sanitizing', 8.690543189252185), ('nh', 8.690543189252185), ('bumping', 8.690543189252185), ('singled', 8.690543189252185), ('environment', 8.690543189252185), ('selfies', 8.690543189252185), ('armed', 8.690543189252185), ('puberty', 8.690543189252185), ('raffle', 8.690543189252185), ('absorbed', 8.690543189252185), ('tastes', 8.690543189252185), ('rapport', 8.690543189252185), ('childrens', 8.690543189252185), ('tasting', 8.690543189252185), ('hydration', 8.690543189252185), ('suess', 8.690543189252185), ('ideals', 8.690543189252185), ('chill', 8.690543189252185), ('chip', 8.690543189252185), ('webquests', 8.690543189252185), ('sudden', 8.690543189252185), ('stifle', 8.690543189252185), ('mercy', 8.690543189252185), ('illiteracy', 8.690543189252185), ('delving', 8.690543189252185), ('delicate', 8.690543189252185), ('complaints', 8.690543189252185), ('israel', 8.690543189252185), ('portray', 8.690543189252185), ('rank', 8.690543189252185), ('impressions', 8.690543189252185), ('lettuce', 8.690543189252185), ('joe', 8.690543189252185), ('backed', 8.690543189252185), ('pola

roid', 8.690543189252185), ('prohibits', 8.690543189252185), ('jeopardy', 8.690543189252185), ('thoughtfulness', 8.690543189252185), ('aforementioned', 8.690543189252185), ('rereading', 8.690543189252185), ('horrific', 8.690543189252185), ('composer', 8.690543189252185), ('hotel', 8.690543189252185), ('unused', 8.690543189252185), ('lids', 8.690543189252185), ('workspaces', 8.690543189252185), ('popping', 8.690543189252185), ('wizards', 8.690543189252185), ('steal', 8.690543189252185), ('weave', 8.690543189252185), ('unemployed', 8.690543189252185), ('wedo', 8.690543189252185), ('utility', 8.690543189252185), ('lounging', 8.690543189252185), ('beakers', 8.690543189252185), ('bead', 8.690543189252185), ('inherent', 8.690543189252185), ('dads', 8.690543189252185), ('privately', 8.690543189252185), ('stride', 8.690543189252185), ('initiated', 8.690543189252185), ('agreement', 8.690543189252185), ('da', 8.690543189252185), ('realms', 8.690543189252185), ('cymbals', 8.690543189252185), ('luck', 8.690543189252185), ('cutters', 8.690543189252185), ('lunchroom', 8.690543189252185), ('uphold', 8.690543189252185), ('colony', 8.690543189252185), ('cultivates', 8.690543189252185), ('madison', 8.690543189252185), ('lonely', 8.690543189252185), ('dawn', 8.690543189252185), ('listener', 8.690543189252185), ('upward', 8.690543189252185), ('gratifying', 8.690543189252185), ('circulate', 8.690543189252185), ('decorative', 8.690543189252185), ('clarinets', 8.690543189252185), ('biased', 8.690543189252185), ('incomplete', 8.690543189252185), ('md', 8.690543189252185), ('processor', 8.690543189252185), ('substances', 8.690543189252185), ('alarm', 8.690543189252185), ('rekenrek', 8.690543189252185), ('decimal', 8.690543189252185), ('upbeat', 8.690543189252185), ('accolades', 8.690543189252185), ('strapped', 8.690543189252185), ('comparisons', 8.690543189252185), ('rehearsing', 8.690543189252185), ('mask', 8.690543189252185), ('charting', 8.690543189252185), ('tellers', 8.690543189252185), ('ceases', 8.690543189252185), ('hdmi', 8.690543189252185), ('bedtime', 8.690543189252185), ('morals', 8.690543189252185), ('authority', 8.690543189252185), ('growling', 8.690543189252185), ('wrapped', 8.690543189252185), ('placements', 8.690543189252185), ('spruce', 8.690543189252185), ('dividing', 8.690543189252185), ('conquering', 8.690543189252185), ('unseen', 8.690543189252185), ('activated', 8.690543189252185), ('retirement', 8.690543189252185), ('punjabi', 8.690543189252185), ('categorize', 8.690543189252185), ('gulf', 8.690543189252185), ('sprawl', 8.690543189252185), ('squirming', 8.690543189252185), ('pigeon', 8.690543189252185), ('rewriting', 8.690543189252185), ('rewrite', 8.690543189252185), ('andy', 8.690543189252185), ('zoos', 8.690543189252185), ('keller', 8.690543189252185), ('constitution', 8.690543189252185), ('tab', 8.690543189252185), ('unconditional', 8.690543189252185), ('pinch', 8.690543189252185), ('kiddo', 8.690543189252185), ('lavender', 8.690543189252185), ('spinners', 8.690543189252185), ('consciousness', 8.690543189252185), ('pitching', 8.690543189252185), ('rods', 8.690543189252185), ('leapfrog', 8.690543189252185), ('uv', 8.690543189252185), ('beneath', 8.690543189252185), ('mit', 8.690543189252185), ('misunderstood', 8.690543189252185), ('channeling', 8.690543189252185), ('confront', 8.690543189252185), ('dignity', 8.690543189252185), ('bending', 8.690543189252185), ('resistable', 8.690543189252185), ('stampers', 8.690543189252185), ('cartoon', 8.690543189252185), ('yay', 8.690543189252185), ('authentically', 8.690543189252185), ('abcya', 8.690543189252185), ('homestead', 8.690543189252185), ('residing', 8.690543189252185), ('uzbekistan', 8.690543189252185), ('keva', 8.690543189252185), ('narrator', 8.690543189252185), ('diplomas', 8.690543189252185), ('watson', 8.690543189252185), ('unnatural', 8.690543189252185), ('beautify', 8.690543189252185), ('adopting', 8.690543189252185), ('monies', 8.690543189252185), ('providers', 8.690543189252185), ('restroom', 8.690543189252185), ('authenticity', 8.690543189252185), ('disagreements', 8.690543189252185), ('vacations', 8.690543189252185), ('lacked', 8.690543189252185), ('lowering', 8.629918567435752), ('realization', 8.6299185674357


```
52), ('pumpkin', 8.629918567435752), ('lamp', 8.629918567435752), ('ad
olescence', 8.629918567435752), ('adept', 8.629918567435752), ('protot
yping', 8.629918567435752), ('logo', 8.629918567435752), ('rechargeabl
e', 8.629918567435752), ('excerpts', 8.629918567435752), ('lover', 8.6
29918567435752), ('promotion', 8.629918567435752), ('launched', 8.6299
18567435752), ('lighthouse', 8.629918567435752), ('artificial', 8.6299
18567435752), ('advertisements', 8.629918567435752), ('lengthy', 8.629
918567435752), ('librarians', 8.629918567435752), ('juggling', 8.62991
8567435752), ('showcased', 8.629918567435752), ('proficiently', 8.6299
18567435752), ('shadows', 8.629918567435752), ('wildcat', 8.6299185674
35752), ('palacio', 8.629918567435752), ('prompted', 8.62991856743575
2), ('expel', 8.629918567435752), ('leaping', 8.629918567435752), ('ar
rays', 8.629918567435752), ('whichever', 8.629918567435752), ('rating
s', 8.629918567435752), ('lip', 8.629918567435752), ('factual', 8.6299
18567435752), ('paraprofessionals', 8.629918567435752), ('quests', 8.6
29918567435752), ('wishlist', 8.629918567435752), ('iron', 8.629918567
435752), ('familial', 8.629918567435752), ('loan', 8.629918567435752),
('wildest', 8.629918567435752), ('readability', 8.629918567435752),
('activist', 8.629918567435752), ('paintbrush', 8.629918567435752),
('grasped', 8.629918567435752), ('membership', 8.629918567435752), ('i
ntentions', 8.629918567435752), ('researcher', 8.629918567435752), ('s
alary', 8.629918567435752)]
```

Creating co-occurrence matrix

In [57]:

```
class CoocuranceMaxtrix:

    def __init__(self, vocab, context_window=5):
        self.vocab = vocab
        self.context_window = context_window

    def fit(self, data):
        """
        input: data - should be a pandas dataframe.
        """
        rows = len(self.vocab)
        cols = len(self.vocab)
        c_mat = pd.DataFrame(np.zeros((rows, cols)), index = self.vocab, columns=self.vocab)

        for text in data.values:
            words = text.split()
            for idx in range(len(words)):
                if c_mat.get(words[idx]) is None:
                    continue
                for cw in range(1, self.context_window+1):
                    if (idx+cw) < len(words):
                        if c_mat.get(words[idx+cw]) is not None:
                            c_mat[words[idx+cw]].loc[words[idx]] = c_mat.get(words[idx]) + 1
                    if (idx-cw) >= 0:
                        if c_mat.get(words[idx-cw]) is not None:
                            c_mat[words[idx-cw]].loc[words[idx]] = c_mat.get(words[idx]) + 1

        np.fill_diagonal(c_mat.values, 0)
        return c_mat
```


Unit testing code with given example

In [58]:

```

vocab = ["abc", "pqr", "def"]
data = [["abc def ijk pqr"], ["pqr klm opq"], ["lmn pqr xyz abc def pqr abc"]]
df = pd.DataFrame(data, columns=['text'])
cvec = CooccurrenceMaxtrix(vocab, 2)
a = cvec.fit(df['text'])
a

```

Out[58]:

	abc	pqr	def
abc	0.0	3.0	3.0
pqr	3.0	0.0	2.0
def	3.0	2.0	0.0

In [59]:

```

coo_vectorizer = CooccurrenceMaxtrix(top_features)
co_mat = coo_vectorizer.fit(X_train['combined_text'])

```

In [60]:

```

cnt = 0
for i in co_mat.index:
    if cnt == 20:
        break
    for j in co_mat.index:
        if co_mat.loc[j][i] != 0:
            print(i, j, "====>", co_mat.loc[j][i])
            cnt += 1
            if cnt == 20:
                break

```

```

unlikely yards ==> 1.0
html java ==> 1.0
html abcy a ==> 1.0
reptiles prey ==> 1.0
humility wit ==> 1.0
humility memoir ==> 1.0
replica medals ==> 1.0
replica ruby ==> 1.0
replica broom ==> 1.0
replica lavender ==> 1.0
improvising glockenspiels ==> 3.0
impulses poke ==> 1.0
considerably shortened ==> 1.0
considerably slowed ==> 1.0
entails carving ==> 1.0
entails formally ==> 1.0
inequality wages ==> 1.0
inequality confronted ==> 1.0
inequality lee ==> 1.0
rehabilitation handmade ==> 1.0

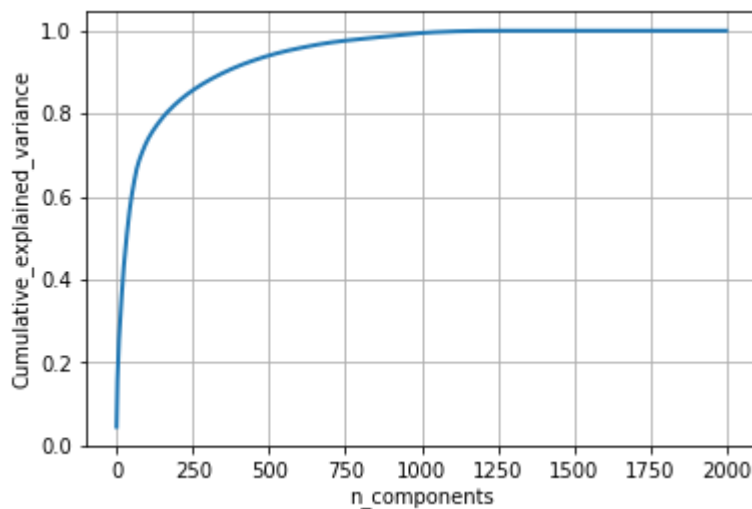
```

2.3 Applying TruncatedSVD and Calculating Vectors for `essay` and `project_title`

In [61]:

```
# Dim-reduction using Truncated SVD and get the Top Singular Values explaining the
from sklearn.decomposition import TruncatedSVD
svd = TruncatedSVD(n_components=1999, random_state=42) #n_components must be < n_fe
svd_matrix = svd.fit_transform(co_mat)
cum_var_explained = np.cumsum(svd.explained_variance_ratio_)
# Plot the Truncated SVD spectrum
plt.figure(1, figsize=(6, 4))

plt.clf()
plt.plot(cum_var_explained, linewidth=2)
plt.axis('tight')
plt.grid()
plt.xlabel('n_components')
plt.ylabel('Cumulative_explained_variance')
plt.show()
```



Observation: From the above plot we can see that `n_components = 600` explain more than 95% variance

In [113]:

```
svd = TruncatedSVD(n_components=600)
svd.fit(co_mat)
trsvd = svd.transform(co_mat)
```

In [121]:

```
print("variance explained by 600 components: ",svd.explained_variance_ratio_.sum())
variance explained by 600 components: 0.9575896104494178
```

In [115]:

```
print(top_features[:20]) # top 20 features
```

```
['gardeners', 'unlikely', 'html', 'reptiles', 'deductive', 'humility',
'replica', 'replay', 'constrained', 'improvising', 'impulses', 'conson-
ant', 'incomparable', 'stopwatches', 'incorrect', 'considerably', 'ind-
icating', 'reinforcers', 'industrious', 'entails']
```

In [116]:

```
tdf = pd.DataFrame(trsvd, index = co_mat.index)
tdf.head() # printing tr svd values after reducing dimentionality to 500dim
```

Out[116]:

	0	1	2	3	4	
gardeners	0.000000e+00	0.000000e+00	0.000000e+00	0.000000e+00	0.000000e+00	0.000000e+00
unlikely	1.546017e-12	9.935938e-18	2.109296e-14	3.853573e-15	-3.243056e-15	-9.649963e-11
html	6.820674e-11	2.282953e-17	2.455548e-08	-8.018255e-09	1.077368e-15	6.345220e-11
reptiles	3.041319e-10	-2.161996e-17	1.938832e-10	-7.996296e-11	2.529968e-15	-1.988162e-11
deductive	0.000000e+00	0.000000e+00	0.000000e+00	0.000000e+00	0.000000e+00	0.000000e+00

5 rows × 600 columns

Creating a dictionary of word to its trsvd value

In [117]:

```
word2trsvd = dict()
i=0
for vec in trsvd:
    word2trsvd[co_mat.index[i]] = vec
    i+=1
```

In [118]:

```
print(word2trsvd['dysgraphia'])
```

```
[ 3.94602885e-07 -8.54285446e-17 1.24886181e-12 5.71307359e-15
-5.58843236e-15 4.47262841e-15 3.21110915e-11 9.76500185e-12
 9.29764852e-09 2.96240226e-14 -1.10499041e-13 -1.22745227e-08
-4.52048262e-10 1.88105421e-09 -6.84685859e-08 7.54947881e-10
 9.02102210e-05 -6.30775236e-05 -6.38853982e-11 3.70830021e-14
 2.24965450e-13 1.03714738e-09 7.73969168e-11 -3.19173048e-11
-1.11595284e-12 4.37154881e-12 3.87755634e-08 1.10253699e-07
 7.27424871e-08 -3.13830225e-12 -8.02038583e-12 -3.02871296e-12
-4.69451886e-12 -2.55861976e-12 4.57022680e-11 -9.95571100e-10
 1.50584410e-08 -8.10854917e-08 6.69520215e-12 1.43329673e-10
-7.25471864e-12 1.02022765e-11 1.79068305e-11 4.31476751e-12
-4.62310601e-08 1.25100033e-08 -4.56916384e-08 -2.10323387e-09
-5.26819745e-07 8.80101523e-09 -9.55282153e-08 -2.47301345e-09
-1.59470984e-09 7.61835490e-10 9.05533987e-08 -6.70936984e-10
 2.42422455e-08 2.62309412e-10 -3.85591920e-11 4.89222759e-09
 8.03643923e-08 6.95183620e-10 5.96893301e-09 5.54889926e-08
 2.31960677e-09 2.65558582e-07 -9.30887262e-08 3.03242894e-09
 1.49413591e-03 -3.29191428e-10 -3.34994506e-10 2.12767823e-03
 1.30462208e-05 -5.61648492e-06 1.38414114e-06 -2.03859573e-07
 2.37680245e-04 -1.79848826e-05 2.98049879e-05 -1.21984839e-08
-5.23491122e-09 2.32560960e-09 3.65271985e-09 -2.29244540e-08
 1.59257901e-07 1.61002257e-08 -5.61564406e-05 1.78862909e-03
 3.96868186e-04 -2.04792359e-07 1.20833803e-05 5.46924291e-06
 2.98941191e-05 1.96498490e-06 4.39619809e-05 5.63200867e-03
 3.97115816e-06 2.49437716e-06 1.40397797e-05 7.09444392e-07
-2.43889182e-04 4.85290625e-04 3.49829551e-04 7.96522693e-06
 6.50690350e-06 -1.12307419e-05 3.19684234e-05 -2.24923447e-06
-4.78495136e-09 -1.96611559e-07 -2.15466895e-07 -1.03685548e-09
-1.55177530e-05 -1.05886594e-03 1.53702103e-02 1.53578438e-04
 1.06575801e-05 9.40272663e-03 3.32896454e-04 -9.89651814e-05
-1.98954628e-04 6.70905248e-04 -4.46736151e-07 1.81291984e-07
 1.65552759e-02 5.72574499e-04 -1.37947514e-03 4.18737199e-03
-1.10501875e-04 -9.19226959e-04 -2.99438113e-04 -4.00824821e-04
 3.61683856e-03 2.89856373e-02 -4.10999797e-04 1.56602132e-02
 1.45094519e-02 1.14470123e-02 9.76861209e-04 -1.92892026e-02
 2.57688105e-02 -2.65438640e-02 -2.30309969e-02 1.62621600e-01
-8.59231286e-04 -3.91435337e-02 -5.86339725e-02 -3.81860375e-02
-1.43726047e-04 -5.20869883e-03 1.21840598e-04 -5.75529802e-02
 9.73566681e-07 -9.48151849e-07 -1.31492141e-02 -7.34422989e-04
-2.50278580e-06 -4.39692342e-05 -8.98966309e-04 -7.08302720e-04
 1.08691069e-03 -9.65438177e-04 -5.32691506e-03 9.84140991e-05
 2.13685603e-04 -1.98435542e-03 1.39655031e-06 7.10745859e-07
 3.03918331e-03 1.66294942e-04 -1.23773725e-03 1.76392029e-07
-1.04655917e-06 -4.14691170e-03 -8.05284481e-04 -1.41539859e-06
 4.63178724e-06 -8.48424399e-04 7.95553121e-04 3.01804921e-04
-6.64331877e-04 1.15630570e-03 -2.59064267e-02 1.53467421e-06
 8.25269300e-03 3.03658984e-03 2.62078685e-02 9.92560767e-07
 2.47141478e-06 -2.83307530e-06 -4.75764184e-06 6.09819956e-06
 7.16429824e-06 -3.47079684e-02 -2.50384749e-02 3.12169853e-02
 4.74753244e-03 -2.94671982e-02 -3.52276700e-02 3.18420169e-03
-2.29039257e-02 -9.32257050e-02 -1.12862502e-02 -1.74987378e-02
 1.33871730e-02 7.89281233e-03 -7.24751212e-04 -4.54935368e-02
-4.81240196e-04 -1.42397818e-02 -2.32686440e-02 -1.26371834e-02
 5.37921877e-02 4.07120486e-03 3.67498918e-03 -4.39560651e-02
-3.20792060e-03 -1.92700793e-02 2.22197815e-03 1.39823307e-03
 2.07825774e-03 7.37338598e-05 -2.92410556e-03 1.79203974e-03
 1.56841378e-01 1.39606122e-03 1.31334819e-01 5.92162021e-04]
```

-2.16188245e-03	2.15156210e-01	3.44061131e-03	4.38766234e-04
-4.79944473e-02	-3.99214818e-03	1.62886534e-03	-1.66453041e-03
2.01039688e-04	-1.43743637e-01	1.84156439e-02	7.16561344e-02
-8.97781790e-03	1.01547649e-02	-1.81323783e-02	3.73258628e-02
5.09539963e-03	-3.94117722e-02	-8.81595600e-03	-3.90876765e-03
2.91772029e-02	8.98293022e-04	3.95856848e-05	1.35801394e-03
-1.30316393e-02	7.06669907e-04	4.30349340e-03	4.07759832e-03
-1.86023251e-03	-8.58403456e-03	5.65711125e-03	-4.41530884e-04
2.68937718e-02	-5.71500377e-03	1.74023770e-02	-7.06340816e-02
9.05151801e-03	-4.31467154e-03	2.90996023e-03	3.08621259e-03
-8.44255645e-02	6.59099576e-05	-1.75301306e-02	3.71618206e-03
1.01414462e-02	2.41516695e-02	-1.00193656e-01	-1.85133814e-02
8.74302592e-03	1.20523819e-02	-8.97091567e-03	-1.53748541e-02
-7.13887941e-04	-2.77139729e-02	-4.40262346e-02	-3.00258773e-05
-4.53932920e-06	-3.72551203e-03	5.08827818e-02	1.26252906e-04
1.01971719e-02	-2.84237196e-02	-2.42771662e-03	-2.27936690e-02
-5.38023826e-03	-3.09341017e-03	5.03716551e-04	1.09537398e-02
5.57944593e-03	-2.81076289e-02	5.05049889e-03	7.00991617e-03
1.04789004e-02	-3.64792926e-03	1.56384440e-03	-1.42232141e-02
-1.36070318e-02	1.55834609e-02	7.55491381e-02	8.40051529e-03
1.02658551e-02	-3.11947931e-03	-5.05960451e-03	4.25954544e-03
3.60984097e-03	1.72774288e-04	2.14926308e-04	5.44429114e-05
1.71126781e-04	6.61850797e-03	-1.80238818e-02	-8.87997454e-03
-1.29777556e-03	9.63569340e-03	6.55048260e-02	2.78963254e-03
-1.84447539e-02	1.50970648e-02	-3.38395508e-02	-2.04805717e-04
1.34994523e-04	-3.95156795e-04	-2.72890981e-04	3.69374245e-02
1.38658926e-03	1.22074286e-02	1.43441570e-02	-3.81292804e-04
-1.19419664e-02	2.46606781e-03	-6.46269505e-02	2.45273443e-03
-4.26641955e-03	1.50795765e-02	-2.75039493e-02	-9.81166043e-05
2.55197482e-04	9.70265405e-03	1.44109293e-02	1.07166656e-03
-1.25565887e-04	1.95461884e-02	5.45416653e-03	-1.94904949e-02
-7.85059378e-05	3.63961252e-05	4.59120469e-02	-2.59739313e-02
-1.43976200e-01	4.39206025e-02	-9.14050791e-03	2.03748241e-01
8.85274447e-03	6.93982093e-04	1.15046689e-01	-1.50391304e-02
2.66742797e-02	-4.26011837e-04	-2.45688398e-03	8.57142466e-03
5.17006877e-02	-1.15572188e-01	-3.09089026e-02	2.76865649e-02
8.78564248e-02	-2.00090588e-01	2.79783800e-02	5.49567270e-02
2.28627392e-02	3.64688605e-03	-1.05739335e-02	-1.88780094e-01
-2.15154576e-01	8.30400428e-03	1.73930061e-02	-6.69631225e-03
2.28616737e-02	-2.35438659e-03	1.91192610e-02	-5.50463008e-03
-4.89035956e-04	9.39112448e-04	-5.41514671e-04	5.78025429e-04
-2.86327428e-04	-1.03118470e-04	8.01055772e-04	2.36000417e-04
1.19344776e-04	-3.35408966e-04	-3.39745001e-04	9.28098722e-04
3.31359272e-05	-9.45142847e-04	3.62938310e-03	-2.24652967e-03
-1.82390448e-02	4.26319055e-03	-2.96973231e-02	-2.26199385e-02
-1.67720141e-02	-5.46288357e-03	-2.27427345e-02	-1.80156225e-02
1.29977324e-01	-5.86951729e-02	5.26362284e-02	2.24511409e-03
8.30827919e-02	-7.98443027e-02	-3.29258010e-02	1.95761139e-01
-3.32389809e-02	-1.03553835e-03	-5.93930540e-03	3.84221759e-03
-2.47502337e-02	-1.80175032e-01	-6.99016560e-02	2.21429847e-03
1.86065423e-03	-3.44486586e-02	-1.11342770e-01	4.05146113e-02
1.20872441e-02	2.28135571e-02	2.92572537e-02	-9.76890732e-02
-1.41022236e-03	1.94129434e-02	1.00825764e-01	-5.55923336e-02
3.80648571e-02	-5.56236686e-02	-8.89811356e-02	-4.20009107e-02
-1.70756582e-01	2.98555198e-02	-2.02388207e-02	-9.59565985e-02
3.72760178e-02	-1.25348281e-01	3.96913498e-02	-7.53028915e-02
1.05256934e-01	-1.27968731e-01	-1.95041697e-01	1.16533568e-02
-4.46922969e-03	-9.81109631e-03	7.15583896e-03	-2.82007245e-02
7.09182351e-03	-1.36677134e-01	2.49800208e-02	-3.38618848e-02
-5.36314246e-04	1.15991470e-01	6.92275027e-02	-8.80940371e-02
3.93047265e-01	1.98426196e-01	-5.60340274e-01	5.73731243e-02

```

-6.12553686e-02 -1.25237877e-01 -1.35002680e-01 -1.81085416e-01
-6.79180620e-02 -7.41833327e-03 -5.88078338e-04 -6.96659496e-03
3.10967177e-03 1.40130679e-03 1.55559155e-02 1.32090234e-02
1.08340577e-02 6.37025975e-02 -5.60303781e-03 1.37320400e-02
1.11663870e-01 -1.43847221e-01 2.39025884e-02 -7.54936788e-03
1.92871528e-02 -3.68769096e-02 2.44986775e-03 1.57048191e-02
1.82151369e-02 1.24293946e-02 -2.69842232e-02 8.80897638e-03
-8.90613026e-03 -1.63908935e-02 -1.30924846e-02 9.17042110e-03
1.26804208e-02 -1.34665299e-02 -3.95054841e-03 -2.92461980e-03
2.69753468e-03 1.98426941e-03 -5.81897178e-03 -3.67341491e-03
1.89254865e-02 -2.23819827e-03 -8.91494748e-03 1.41127865e-02
2.25640289e-02 -4.58485570e-03 3.67975616e-03 2.68504826e-02
7.14036466e-03 1.90903863e-02 3.47194220e-02 -2.34486114e-02
1.76830338e-02 -2.02272901e-02 -1.79895090e-02 2.79439271e-02
-6.47400300e-02 2.05572692e-02 -6.10857465e-02 -4.54884480e-02
6.62551887e-02 1.80633560e-02 5.68241565e-02 -4.90056247e-02
-1.18268707e-02 3.22863760e-02 -3.26681646e-02 5.43255245e-04
5.18815432e-03 2.40950841e-02 -2.68392524e-02 2.61512186e-02
1.27571189e-02 -2.95234585e-02 4.28518256e-02 -3.01346583e-02
-9.10624538e-02 2.12484441e-02 1.68958945e-02 -1.87139139e-02
3.44644541e-02 3.44099742e-02 -1.98388012e-02 -4.00936757e-02
5.84670638e-02 -1.65826029e-02 -7.99214995e-02 -3.65728754e-02
-4.58269384e-02 4.97358424e-02 7.97454709e-02 2.29967797e-02
4.67208169e-03 -4.99648321e-02 6.49859010e-02 -4.83260360e-02
-3.17916278e-02 1.16084312e-01 2.63321884e-01 -4.74980079e-02
-7.91532628e-03 -3.32884801e-02 -3.38713228e-02 -6.87322792e-02
-2.63309196e-02 1.42772875e-01 -3.60209204e-04 8.58600243e-02
2.30998553e-02 -5.33895028e-03 4.29970523e-02 -9.63010997e-03
6.77102822e-03 5.93833961e-02 -9.98151961e-03 3.82269585e-02
1.19408813e-04 -1.85908254e-02 5.93783218e-03 -3.75461606e-02
2.23599972e-02 3.24241950e-02 -1.18431319e-02 -2.42572770e-02
9.36448971e-03 4.22663744e-02 -3.30514336e-02 1.66408292e-02]

```

In [134]:

```

def get_word_embeddings(data,col):
    embeddings = []
    cnt = 0
    for essay in data[col]:
        embedding = np.zeros(600)
        wc=0
        for word in essay.split():
            if co_mat.get(word) is not None:
                #print('index ', cnt)
                #print('here for ',word)
                embedding+=word2trsvd[word]
                #print(embedding)
                wc+=1
        cnt+=1
        #print("index ",cnt)
        if wc!=0:
            embedding /= wc
            #print(embedding)

        embeddings.append(embedding)
    return np.array(embeddings)

```

In [135]:

```
essay_train_embeddings = get_word_embeddings(X_train, 'clean_essay')
title_train_embeddings = get_word_embeddings(X_train, 'clean_title')
print(title_train_embeddings.shape, essay_train_embeddings.shape)
```

(35000, 600) (35000, 600)

In [138]:

```
essay_test_embeddings = get_word_embeddings(X_test, 'clean_essay')
title_test_embeddings = get_word_embeddings(X_test, 'clean_title')
print(title_test_embeddings.shape, essay_test_embeddings.shape)
```

(15000, 600) (15000, 600)

In [137]:

```
#get_word_embeddings(X_train,'clean_title')
title_train_embeddings[12787]
```

Out[137]:

```
array([-2.07798012e-18, -4.64919493e-19,  2.21938387e-18, -2.12341227e
-19,
       -3.53100864e-17,  3.30632233e-18, -8.42400407e-18,  1.05393800e
-17,
        5.47938270e-17, -2.85249833e-16, -2.70965797e-17,  2.25257944e
-17,
        1.33997672e-17, -2.03837954e-17,  4.31002665e-18, -7.27091459e
-18,
        2.23403183e-16,  2.85592100e-16,  2.28469279e-16,  4.75589125e
-17,
       -4.22192820e-16,  2.38096588e-16, -4.51916397e-16, -1.54769787e
-16,
       -2.73918805e-16,  1.89647141e-17,  9.06158612e-16,  1.24850475e
-15,
       -1.68971897e-15,  1.74446921e-15, -3.99977022e-16, -7.80413569e
-16,
       -1.88219972e-15,  6.40562524e-16,  5.17761458e-16,  1.09802556e
-15,
        2.69983455e-16,  4.30977171e-17,  1.43665571e-15, -4.23601028e
-15,
       -3.31110992e-15, -2.10465504e-16,  6.36852233e-15,  1.58248718e
-15,
       -1.15651919e-15, -8.00422240e-15, -5.95457828e-15,  1.68311911e
-14,
        1.81235956e-14, -4.26293613e-16,  6.52533907e-15, -1.02342404e
-14,
        1.75712533e-14,  2.68281793e-14, -2.51409161e-14,  6.13123985e
-14,
        6.84962830e-14, -1.11986130e-14,  4.99062201e-14, -1.79434539e
-14,
        5.39457527e-14,  1.09447476e-13, -1.15622806e-14, -1.46827892e
-13,
       -3.62220256e-14,  7.20145161e-14, -4.19638417e-15, -7.91842487e
-14,
       -3.03404310e-14, -8.10868602e-13,  6.51850680e-13,  1.79526587e
-12,
        1.94860938e-12,  5.11433058e-12,  5.78040874e-12,  5.18794014e
-12,
        5.51603003e-13,  5.98286497e-12,  2.44263492e-12,  4.81944251e
-13,
        1.17629519e-12, -3.09795324e-12,  1.72925428e-12, -6.09755234e
-12,
        1.26368921e-11,  2.50606528e-11,  5.05590839e-12,  8.14351844e
-12,
       -7.15140677e-12, -8.28679301e-12,  1.45383502e-11, -1.10547065e
-12,
       -1.28292545e-11,  4.09577301e-12,  2.38270115e-11,  4.25990097e
-11,
       -1.67472618e-11,  2.66392375e-11, -7.08502696e-11, -1.94721732e
-10,
        1.67763355e-11, -4.76882385e-11, -1.91849027e-11,  7.19858382e
-11,
        3.19205309e-10, -1.40080551e-09,  8.99015009e-10, -5.66070881e
-10,
```


-10, 3.32081532e+00, 2.22983982e+00, 3.78561038e-09, -4.66969447e
-10, 6.11312965e-10, 9.65365109e-10, 8.69144644e-10, 6.52588213e
-11, 3.10300934e-10, -1.02083108e-09, -9.29843779e-10, 6.89455770e
-11, -3.54521278e-10, -1.62053734e-10, -3.51206132e-10, -7.74061764e
-10, -2.56962423e-10, -7.47206224e-10, -1.06732629e-10, 4.53124374e
-10, 4.39149409e-10, 6.78426644e-10, -7.27988527e-10, -8.30820029e
-10, 4.35228293e-11, -5.25024193e-10, -1.80543084e-10, -5.58691576e
-10, 4.17455675e-10, 4.64054461e-10, -7.80266812e-10, -2.45805576e
-10, -4.23361548e-10, -2.13340233e-10, 3.14328791e-10, 8.16660275e
-10, 1.33608816e-09, 2.31700272e-10, -5.95539416e-10, 4.65068224e
-10, 4.90416723e-10, 1.75432874e-10, -7.87215634e-11, -8.30863643e
-10, 4.70368807e-10, 8.73817713e-10, 9.09938075e-10, -3.90030066e
-10, 4.57688518e-10, 1.08735895e-10, -3.03213775e-10, -1.16310047e
-09, -5.59951927e-10, 1.09893022e-09, 5.04859702e-10, 9.00843878e
-10, 6.69224447e-10, -6.31422236e-10, 2.91907360e-10, 1.82616425e
-09, 6.72771344e-10, -9.66777376e-10, -6.77114680e-10, -6.72201130e
-10, 5.78962379e-11, 6.08321029e-10, -4.78309972e-11, -1.11340860e
-09, -6.46898638e-10, -8.16322116e-10, 4.34051547e-10, -8.53587655e
-10, 4.67509478e-12, 1.16786903e-09, 1.87331259e-09, 1.41733385e
-09, 4.73452329e-10, -1.63801695e-10, 9.55589129e-10, -2.89652465e
-10, 3.51572227e-10, 1.35797803e-09, 7.22134420e-10, -1.77155467e
-09, -7.58505981e-10, -1.84230660e-09, -7.91826883e-11, -3.10325294e
-09, 8.57121675e-10, -1.49478508e-09, 4.39066585e-10, -2.72976987e
-09, -2.31918665e-09, -1.47649041e-09, 1.59938921e-09, -1.84643135e
-10, -2.38953481e-09, -9.42089565e-10, 2.66939182e-09, -5.06744555e
-10, -8.82333893e-11, -1.99323985e-09, 3.65291156e-10, 5.82028671e
-10, 1.83825738e-09, -1.39833368e-10, 1.80763217e-09, 1.28524046e
-09, -3.54048388e-10, 3.59748510e-09, 2.35965152e-09, 4.60818113e
-09, 7.56881815e-10, -2.71600357e-09, -1.29850500e-09, 4.39309067e
-09, -9.96758323e-10, -4.63145254e-09, -2.00119048e-09, 2.77335969e
-09, 8.86372304e-10, 1.69175312e-10, 3.16387632e-09, 1.34947600e

-09,
1.51135635e-09, 1.83943615e-09, 3.79204690e-11, -1.67037685e
-09,
5.92301104e-09, -2.88843180e-09, -3.06909727e-09, 6.47696061e
-10,
-1.85133911e-09, -1.97737198e-09, 3.30396106e-09, 1.03662924e
-09,
-1.23376056e-09, 2.01983979e-09, 4.34716496e-09, -6.85038205e
-09,
1.42660146e-09, -6.93230350e-10, 8.45171207e-09, -8.83411816e
-11,
2.18314048e-09, -5.11455008e-09, 1.51793643e-09, 9.04218245e
-10,
-2.71723503e-10, -3.05277173e-09, -5.77539431e-10, -5.81856140e
-09,
3.13304741e-09, -6.61427731e-10, -7.96087444e-09, 6.66436261e
-10,
-9.04836258e-09, -1.70487119e-09, -9.25697687e-10, -2.14781297e
-09,
7.48480444e-09, -4.66719738e-10, -1.34108931e-10, 3.63187953e
-09,
2.55807581e-09, 1.07092510e-08, 1.23713449e-08, -3.91959157e
-09,
-4.95534603e-09, -1.91917766e-09, 7.70215658e-09, -9.47800964e
-09,
2.41443932e-09, -5.80552058e-09, 1.13117036e-08, -4.87118453e
-10,
6.74439596e-10, -2.19566389e-09, 9.41984574e-09, 8.91766327e
-09,
3.76861816e-09, -7.31062301e-09, -8.30737594e-09, 1.05411906e
-08,
2.16872840e-08, 1.04970504e-08, -1.07400352e-08, -9.10748161e
-09,
1.18031037e-09, -6.84422635e-09, -3.17924571e-09, -1.39539699e
-09,
5.78566521e-09, 8.15264618e-09, -1.28893840e-08, 1.92864874e
-10,
9.83983178e-09, 1.13877571e-08, 1.20456495e-08, 1.75081528e
-08,
-7.74007451e-09, -8.34831825e-09, -1.20516693e-09, -4.45854925e
-09,
-5.90741842e-09, -1.03151127e-08, -7.99935588e-09, 4.78805861e
-09,
-8.81608404e-10, 1.23227519e-08, -3.79038308e-10, -2.39115916e
-09,
-1.74507041e-08, -1.21414295e-08, 1.11012292e-09, -1.48448739e
-08,
-5.77573418e-09, 7.14679334e-09, 1.02858642e-08, 1.80482711e
-09,
-2.56018968e-09, 7.71160440e-09, 2.41273866e-08, -5.69702014e
-09,
1.31229699e-09, 9.12933632e-09, -1.77589363e-08, -6.84186049e
-09,
6.32985571e-09, -1.23188981e-08, 9.53984549e-09, -4.14455586e
-09,
-1.01582127e-08, -1.83202161e-09, 9.17308868e-09, -1.03458137e
-08,
5.53742384e-10, 3.80796701e-08, -4.93113306e-09, -1.13202684e
-08,
9.45099064e-09, -9.37243589e-09, -2.02550899e-08, -7.69431381e
-09,

-8.95325136e-09, 4.63349135e-08, -4.05274602e-08, 3.97774976e-08,
4.45347026e-08, -3.58333063e-08, -6.47967627e-09, -2.68433298e-08,
1.22802246e-08, -8.89241694e-09, 3.53992530e-09, -2.38659182e-08,
3.45379475e-08, 9.03170490e-09, 2.03921447e-08, -1.26032437e-08,
-2.33775575e-09, -8.56506549e-10, -2.48826496e-09, 3.82574454e-08,
1.00112554e-09, 7.93660529e-09, -2.60132371e-08, -4.72912361e-08,
-9.59573347e-10, 8.05979402e-09, -4.36372867e-08, -1.03976709e-08,
-2.16101692e-08, -4.44833510e-08, 6.08511636e-10, 5.73954374e-10,
7.65123396e-08, -3.20084508e-08, 4.38338714e-10, 3.20935341e-08,
-1.29983788e-08, -7.70130948e-09, 4.49678804e-09, -1.15812731e-08,
-2.88998708e-09, 2.93932246e-08, 2.59148716e-09, 1.40411177e-08,
-7.88480044e-09, -1.38424850e-08, 4.81319517e-08, 5.79311134e-08,
1.51140763e-09, -5.90064654e-08, -1.17040345e-07, -1.15605224e-08,
-6.06888640e-09, 5.49549054e-08, -7.03288475e-09, 1.69107500e-08,
1.64805682e-08, -3.30152822e-08, 5.42590318e-09, 2.13094198e-08,
4.73639284e-08, 2.07160319e-09, 6.96695877e-09, 1.30498821e-08,
1.40560108e-09, -2.26338627e-08, 3.76035497e-08, -1.21180416e-08,
-1.68916964e-08, 8.19031498e-08, -3.47453779e-08, 1.59058707e-08,
-1.36021173e-08, 2.05254862e-08, -1.58467554e-08, -3.21745800e-08,
-8.53612237e-08, -2.12818294e-08, -3.59883158e-08, -4.04263460e-08,
-3.43742479e-08, -6.93980898e-08, -1.11469522e-07, 2.05711398e-08,
3.42945719e-09, -8.05140537e-08, -1.01128213e-07, 9.79633662e-08,
-3.74819184e-08, -4.94966057e-08, 2.00120554e-08, 1.42075588e-07,
-1.16379850e-07, -7.56964716e-08, 1.50662636e-07, -4.23208109e-08,
-1.97032272e-08, 2.95657568e-08, -5.07810986e-08, -4.04318994e-09,
7.85340169e-08, -1.69258613e-08, 7.31744719e-09, 4.68410794e-08,
1.21317455e-08, 2.55404480e-08, 3.92207246e-08, 2.13534611e-08,
6.86920605e-08, 2.82332877e-08, 1.22503940e-07, 1.87152213e-08,
-4.11693058e-08, 1.33491368e-07, 9.91082805e-08, 1.40660622e-08,
1.57856211e-08, -5.23128958e-08, -1.63984101e-07, -3.27547022e-08,
6.27850948e-08, -2.80684252e-08, -8.23157446e-08, -1.06434672e-08

-07,
7.48068452e-08, -9.83227342e-09, -3.83151408e-08, -9.97940311e
-09,
-3.92852297e-08, 5.96711439e-08, 3.10157510e-08, -3.99407083e
-08,
1.40264649e-07, 9.86566874e-08, -6.98878102e-09, -6.19891213e
-08,
-1.24975991e-08, -6.11136141e-08, 8.17112248e-08, -1.02517250e
-08,
5.90433331e-08, -7.02149368e-08, -8.16955624e-08, -9.66612525e
-08,
-6.65574025e-09, -2.63650649e-08, -1.85085625e-07, 2.63902550e
-08,
4.46172420e-08, -1.55453694e-07, -6.97038469e-08, 1.74369633e
-07,
-1.00670373e-07, -1.25755902e-07, -4.79229361e-08, 2.79744713e
-08,
-2.56818281e-08, 2.11211546e-08, 8.44162234e-09, -6.15808167e
-08,
8.78324632e-09, 7.03390062e-08, -2.33212299e-07, -2.18628911e
-07,
1.00530616e-07, -1.03150575e-07, -8.89071129e-08, -1.13563211e
-07,
8.29940445e-08, -3.30472182e-08, 9.65360829e-08, -1.53875551e
-07,
2.04473964e-08, 1.11642433e-07, 2.56743045e-08, 1.37996828e
-07,
-1.12416098e-08, 5.63972362e-08, -1.83716656e-07, -2.42239306e
-08,
6.90056333e-08, -6.84055202e-08, 4.20864551e-08, 4.11884248e
-08,
-1.90931910e-08, 6.49506293e-08, -1.45584021e-07, 7.66999117e
-08,
5.13963955e-08, 2.45556487e-07, 1.00542161e-07, -2.11256807e
-07,
1.05828348e-07, 3.18580926e-07, -7.92744515e-08, 5.64587130e
-08,
1.02037831e-07, 1.29480397e-07, -1.19791494e-07, -3.97536187e
-08,
-6.69576313e-08, 2.34584374e-07, 8.87642089e-08, 2.83439843e
-07,
1.20049856e-07, 1.24984008e-07, 6.16196323e-08, 3.18761298e
-07,
6.65725736e-08, 3.62845758e-08, -1.29332250e-07, -1.41891132e
-07,
8.05743530e-08, 6.29081109e-08, -5.24522123e-09, -8.61905242e
-08,
6.82841503e-08, 1.12146638e-07, 2.02758809e-08, -1.00601851e
-08,
-4.14010174e-08, 3.61565068e-08, -3.91278297e-08, 1.88677050e
-08,
9.93386705e-08, 1.52985675e-08, 2.93864147e-08, -9.46077968e
-08,
-2.72608128e-08, 4.07647495e-08, -3.40867676e-08, -1.09308329e
-07,
-3.21572328e-08, -1.99530246e-09, 1.02272104e-07, -4.45939673e
-08,
1.31851501e-07, 3.97952939e-08, 5.16635778e-08, 4.07915881e
-08,
1.74749311e-07, -4.95701984e-08, -9.84166563e-08, 1.62401805e
-07,

```
2.91811618e-08, 8.36550184e-08, 1.43197625e-07, -1.80382448e-07])
```

1. Normalizing Quantity

In [139]:

```
quantity_vectorizer = preprocessing.Normalizer().fit(X_train['quantity'].values.reshape(1,-1))
X_train_qty = quantity_vectorizer.transform(X_train['quantity'].values.reshape(1,-1))
X_test_qty = quantity_vectorizer.transform(X_test['quantity'].values.reshape(1,-1))
```

2. Sentiment score of each essay

In [140]:

```
def get_sentiment_score(feature):

    analyser = SentimentIntensityAnalyzer()
    pos = []
    neg = []
    neu = []
    compound = []

    for text in tqdm(feature) :
        score = analyser.polarity_scores(text)
        nt = score['neg']
        pt = score['pos']
        ntr = score['neu']
        cmp = score['compound']
        neg.append(nt)
        pos.append(pt)
        neu.append(ntr)
        compound.append(cmp)

    return (pos,neg,neu,compound)
```

In [141]:

```
X_train_essay_pos, X_train_essay_neg, X_train_essay_neu, X_train_cmp = get_sentimen
100%|██████████| 35000/35000 [01:06<00:00, 528.90it/s]
```

In [142]:

```
X_test_essay_pos, X_test_essay_neg, X_test_essay_neu, X_test_cmp = get_sentiment_sc
100%|██████████| 15000/15000 [00:29<00:00, 515.88it/s]
```

3. Number of words in Title

In [143]:

```
#Reference: https://stackoverflow.com/questions/27488446/how-do-i-get-word-frequenc
def get_word_count(feature):
    vectorizer = CountVectorizer()
    vectorizer.fit(feature)
    v0 = vectorizer.transform(feature)
    return np.sum(v0,axis=1)
```

In [144]:

```
X_train_title_wc = get_word_count(X_train['clean_title'])
X_test_title_wc = get_word_count(X_test['clean_title'])
print(X_train_title_wc.shape)
print(X_test_title_wc.shape)
```

```
(35000, 1)
(15000, 1)
```

4. Number of words in combined essays

In [145]:

```
X_train_essay_wc = get_word_count(X_train['clean_essay'])
X_test_essay_wc = get_word_count(X_test['clean_essay'])
print(X_train_essay_wc.shape)
print(X_test_essay_wc.shape)
```

```
(35000, 1)
(15000, 1)
```

Applying GBDT : Creating final data matrix

In [146]:

```
f1 = X_train_school_state_ohe
f2 = X_train_category_ohe
f3 = X_train_subcategory_ohe
f4 = X_train_grade_category_ohe
f5 = X_train_teacher_prefix_ohe
f6 = np.array(X_train_price_normalized).reshape(-1,1)
f7 = np.array(X_train_normal_previous_project).reshape(-1,1)
f8 = np.array(X_train_qty).reshape(-1,1)
f9 = np.array(X_train_essay_pos).reshape(-1,1)
f10 = np.array(X_train_essay_neg).reshape(-1,1)
f11 = np.array(X_train_essay_neu).reshape(-1,1)
f12 = np.array(X_train_cmp).reshape(-1,1)
f13 = X_train_title_wc
f14 = X_train_essay_wc
f15 = essay_train_embeddings
f16 = title_train_embeddings
X_tr_new = hstack((f1,f2,f3,f4,f5,f6,f7,f8,f9, f10, f11, f12, f13, f14,f15,f16))
X_tr_new.shape
```

Out[146]:

```
(35000, 1309)
```

In [147]:

```

f1 = X_test_school_state_ohe
f2 = X_test_category_ohe
f3 = X_test_subcategory_ohe
f4 = X_test_grade_category_ohe
f5 = X_test_teacher_prefix_ohe
f6 = np.array(X_test_price_normalized).reshape(-1,1)
f7 = np.array(X_test_normal_previous_project).reshape(-1,1)
f8 = np.array(X_test_qty).reshape(-1,1)
f9 = np.array(X_test_essay_pos).reshape(-1,1)
f10 = np.array(X_test_essay_neg).reshape(-1,1)
f11 = np.array(X_test_essay_neu).reshape(-1,1)
f12 = np.array(X_test_cmp).reshape(-1,1)
f13 = X_test_title_wc
f14 = X_test_essay_wc
f15 = essay_test_embeddings
f16 = title_test_embeddings
X_test_new = hstack((f1,f2,f3,f4,f5,f6,f7,f8,f9, f10, f11, f12, f13, f14,f15,f16))
X_test_new.shape

```

Out[147]:

(15000, 1309)

In [165]:

```

from xgboost.sklearn import XGBClassifier
xgb = XGBClassifier(class_weight='balanced')
tune_parameters = {'n_estimators': [10, 100, 500, 1000], 'max_depth': [2, 3, 4, 5, 6]
clf = GridSearchCV(xgb, tune_parameters, cv= 5, scoring='roc_auc', n_jobs=-1, verbose=1)
clf.fit(X_tr_new, Y_train)

```

Fitting 5 folds for each of 36 candidates, totalling 180 fits

[Parallel(n_jobs=-1)]: Using backend LokyBackend with 16 concurrent workers.

[Parallel(n_jobs=-1)]: Done 18 tasks | elapsed: 3.4min

[Parallel(n_jobs=-1)]: Done 180 out of 180 | elapsed: 173.7min finished

Out[165]:

```

GridSearchCV(cv=5, error_score='raise-deprecating',
             estimator=XGBClassifier(base_score=0.5, booster='gbtree', class_weight='balanced',
                                     colsample_bylevel=1, colsample_bynode=1, colsample_bytree=1,
                                     gamma=0, learning_rate=0.1, max_delta_step=0, max_depth=3,
                                     min_child_weight=1, missing=None, n_estimators=100, n_jobs=1,
                                     nthread=None, objective='binary:logistic', random_state=0,
                                     reg_alpha=0, reg_lambda=1, scale_pos_weight=1, seed=None,
                                     silent=None, subsample=1, verbosity=1),
             fit_params=None, iid='warn', n_jobs=-1,
             param_grid={'n_estimators': [10, 100, 500, 1000], 'max_depth': [2, 3, 4, 5, 6, 7, 8, 9, 10]},
             pre_dispatch='2*n_jobs', refit=True, return_train_score='warn',
             scoring='roc_auc', verbose=True)

```

In [166]:

```
clf.best_estimator_
```

Out[166]:

```
XGBClassifier(base_score=0.5, booster='gbtree', class_weight='balance  
d',  
               colsample_bylevel=1, colsample_bynode=1, colsample_bytree=1,  
               gamma=0, learning_rate=0.1, max_delta_step=0, max_depth=3,  
               min_child_weight=1, missing=None, n_estimators=100, n_jobs=1,  
               nthread=None, objective='binary:logistic', random_state=0,  
               reg_alpha=0, reg_lambda=1, scale_pos_weight=1, seed=None,  
               silent=None, subsample=1, verbosity=1)
```

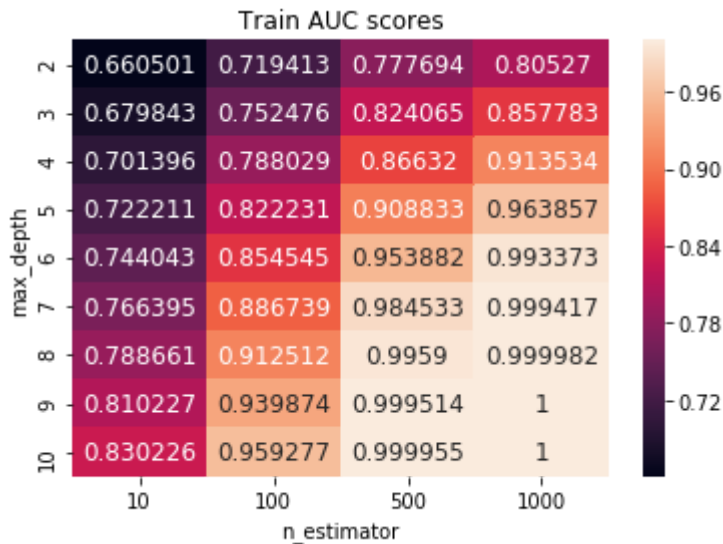
In [167]:

```
def train_cv_scores_for_params(model):  
    results = pd.DataFrame.from_dict(model.cv_results_)  
    max_depths = []  
    n_estimators = []  
    mean_cv_scores = []  
    mean_train_scores = []  
    for p in zip(results['params'], results['mean_test_score'], results['mean_train_score']):  
        param_dict, score_test, score_train = p  
        max_depth, n_estimator = param_dict.values()  
        max_depths.append(max_depth)  
        n_estimators.append(n_estimator)  
        mean_cv_scores.append(score_test)  
        mean_train_scores.append(score_train)  
    return max_depths, n_estimators, mean_cv_scores, mean_train_scores
```

Train AUC Scores

In [168]:

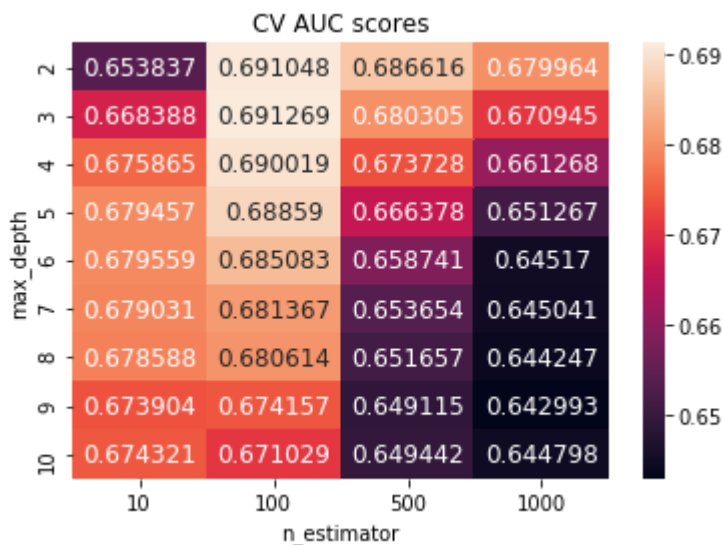
```
max_depths, n_estimators, mean_cv_scores, mean_train_scores = train_cv_scores_for_pa
df = pd.DataFrame({'max_depth': max_depths, 'n_estimator': n_estimators, 'mean_test_score': mean_test_scores})
pivot = df.pivot(index = "max_depth", columns = "n_estimator", values = "mean_test_score")
sns.heatmap(pivot, annot=True, annot_kws={"size": 12}, fmt='g')
plt.title("Train AUC scores")
plt.show()
```



CV AUC Scores

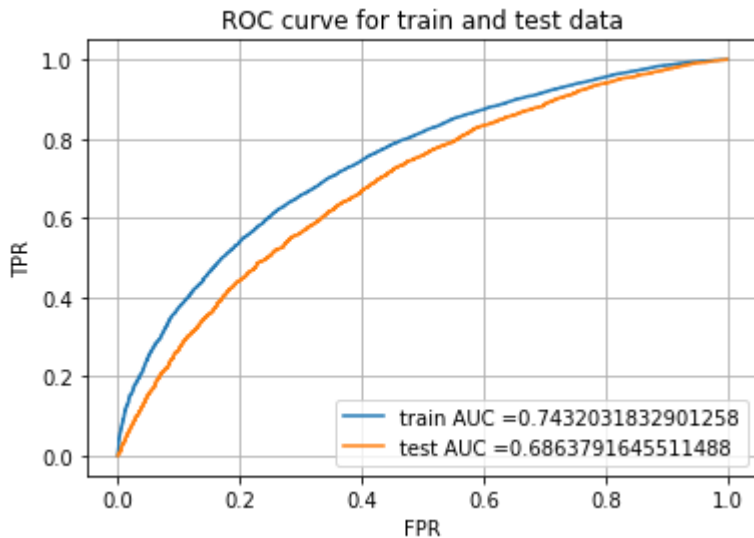
In [169]:

```
df = pd.DataFrame({'max_depth': max_depths, 'n_estimator': n_estimators, 'mean_test_score': mean_test_scores})
pivot = df.pivot(index = "max_depth", columns = "n_estimator", values = "mean_test_score")
sns.heatmap(pivot, annot=True, annot_kws={"size": 12}, fmt='g')
plt.title("CV AUC scores")
plt.show()
```



In [170]:

```
xgb = clf.best_estimator_  
xgb.fit(X_tr_new,Y_train)  
  
y_train_pred = xgb.predict_proba(X_tr_new)  
y_test_pred = xgb.predict_proba(X_test_new)  
  
# roc_auc_score(y_true, y_score) the 2nd parameter should be probability estimates  
# not the predicted outputs  
train_fpr, train_tpr, tr_thresholds = roc_curve(Y_train, y_train_pred[:,1])  
test_fpr, test_tpr, te_thresholds = roc_curve(Y_test, y_test_pred[:,1])  
  
plt.plot(train_fpr, train_tpr, label="train AUC =" +str(auc(train_fpr, train_tpr)))  
plt.plot(test_fpr, test_tpr, label="test AUC =" +str(auc(test_fpr, test_tpr)))  
plt.legend()  
  
plt.xlabel("FPR")  
plt.ylabel("TPR")  
plt.title("ROC curve for train and test data")  
plt.grid()  
plt.show()
```



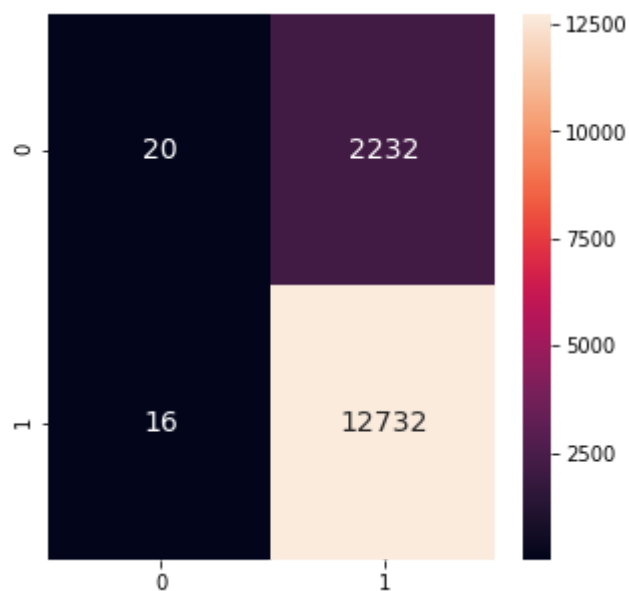
In [171]:

```
y_test_predict = xgb.predict(X_test_new)

results = confusion_matrix(Y_test, y_test_predict)
plt.figure(figsize = (5,5))
sns.heatmap(results, annot=True,annot_kws={"size": 14}, fmt='g')
```

Out[171]:

<matplotlib.axes._subplots.AxesSubplot at 0x7f9daca896a0>





3. Conclusion


In [174]:


```
x = PrettyTable()
x.field_names = ["Vectorizer", "Model", "n_estimator:hyperparam", "max_depth:hyperpa
x.add_row(["Word-Vec Matrix-Fact (Co-occur)", "GBDT", 100, 3, 0.68])
print(x)
```

Vectorizer		Model	n_estimator:hyperparam	max_depth:hyperparam	AUC
Word-Vec Matrix-Fact (Co-occur)		GBDT	100	3	0.68

 Present

 Slides

 Themes

 Help