mayankgupta9968_gmail_com_14

November 23, 2019

0.1 Assignment : 14

0.1.1 Model-1

Build and Train deep neural network as shown below
ref: https://i.imgur.com/w395Yk9.png

- **Input_seq_total_text_data** — You have to give Total text data columns. After this use the Embedding layer to get word vectors. Use given predefined glove word vectors, don't train any word vectors. After this use LSTM and get the LSTM output and Flatten that output.
- **Input_school_state** — Give 'school_state' column as input to embedding layer and Train the Keras Embedding layer.
- **Project_grade_category** — Give 'project_grade_category' column as input to embedding layer and Train the Keras Embedding layer.
- **Input_clean_categories** — Give 'input_clean_categories' column as input to embedding layer and Train the Keras Embedding layer.
- **Input_clean_subcategories** — Give 'input_clean_subcategories' column as input to embedding layer and Train the Keras Embedding layer.
- **Input_teacher_prefix** — Give 'input_teacher_prefix' column as input to embedding layer and Train the Keras Embedding layer.
- **Input_remaining_teacher_number_of_previously_posted_projects._resource_summary_contains_nume** —concatenate remaining columns and add a Dense layer after that.

- For LSTM, you can choose your sequence padding methods on your own or you can train your LSTM without padding, there is no restriction on that.

Below is an example of embedding layer for a categorical columns. In below code all are dummy values, we gave only for referance.

[0]:
```
# https://stats.stackexchange.com/questions/270546/
 ↪how-does-keras-embedding-layer-work
# input_layer = Input(shape=(n,))
# embedding = Embedding(no_1, no_2, input_length=n)(input_layer)
# flatten = Flatten()(embedding)
```

0.1.2 1. Go through this blog, if you have any doubt on using predefined Embedding values in Embedding layer - https://machinelearningmastery.com/use-word-embedding-layers-deep-learning-keras/

0.1.3 2. Please go through this link https://keras.io/getting-started/functional-api-guide/ and check the 'Multi-input and multi-output models' then you will get to know how to give multiple inputs.

0.1.4 Model-2

Use the same model as above but for 'input_seq_total_text_data' give only some words in the sentance not all the words. Filter the words as below.

0.1.5 Model-3

ref: https://i.imgur.com/fkQ8nGo.png

- **input_seq_total_text_data**:
- **Other_than_text_data**:
  . Convert all your Categorical values to onehot coded and then concatenate all these onehot vectors . Neumerical values and use CNN1D as shown in above figure. . You are free to choose all CNN parameters like kernel sizes, stride.

[2]:

```
#importing all the required lib
import pandas as pd
import numpy as np
import os
import math
from collections import defaultdict
import matplotlib.pyplot as plt
from sklearn.feature_extraction.text import TfidfVectorizer,CountVectorizer
from sklearn.model_selection import train_test_split
from keras.preprocessing.text import Tokenizer
from keras.preprocessing.sequence import pad_sequences
from keras.layers import SpatialDropout1D, LSTM,␣
 ↪BatchNormalization,concatenate,Flatten,Embedding,Dense,Dropout,MaxPooling2D,Reshape,CuDNNLS
from keras.models import Sequential
from keras import Model,Input
from keras.layers.convolutional import Conv2D,Conv1D
import keras.backend as k
from sklearn.metrics import roc_auc_score
import tensorflow as tf
import keras
from sklearn.utils import compute_class_weight
from keras.initializers import he_normal,glorot_normal
from keras.regularizers import l1,l2
from keras.callbacks import Callback, EarlyStopping,␣
 ↪ModelCheckpoint,LearningRateScheduler
from time import time
from tensorflow.python.keras.callbacks import TensorBoard
from IPython.display import SVG, display
import pickle
import warnings
warnings.filterwarnings("ignore")
tf.compat.v1.disable_eager_execution()
```

```
Using TensorFlow backend.

<IPython.core.display.HTML object>
```

[0]:

```
# create logger instance
# Create a TensorBoard instance with the path to the logs directory
# tensorboard = TensorBoard(log_dir='logs/{}'.format(time()))
```

[4]:

```
# mounted my Google Drive in colab
from google.colab import drive
drive.mount('/content/drive')
```

```
Drive already mounted at /content/drive; to attempt to forcibly remount, call
drive.mount("/content/drive", force_remount=True).
```

[5]:

```
# Check ls command
!ls "/content/drive/My Drive/LSTM Assignment"
```

```
 1_Reference_EDA.ipynb             'LSTM - Assignment.ipynb'   train_data.csv
 2_Reference_Preprocessing.ipynb   preprocessed_data.csv
 glove_vectors                     resources.csv
```

[6]:

```
!ls "/content/drive/My Drive/LSTM Output"
```

```
epochs:001-val_acc:0.500.hdf5  model_2.png
epochs:002-val_acc:0.500.hdf5  model_3.png
epochs:003-val_acc:0.500.hdf5  weights_copy_assig_2.best.hdf5
epochs:004-val_acc:0.500.hdf5  weights_copy_assig_3.best.hdf5
epochs:005-val_acc:0.500.hdf5  weights_copy_assig.best.hdf5
glove.6B.300d.txt              weights_copy.best.hdf5
glove.6B.300d.txt.zip          weights_copy_new_23_1.best.hdf5
logs                           weights_copy_new_23_2.best.hdf5
model_1.png                    weights_copy_new_23.best.hdf5
```

[0]:

```
glovevectorfile = open('/content/drive/My Drive/LSTM Assignment/glove_vectors',␣
 ↪'rb')
glovevector = pickle.load(glovevectorfile)
```

[8]:

```
glovevector['respirometer'].shape
```

[8]:

```
(300,)
```

[9]:

```
glovevector['respirometer']
```

[9]:

```
array([-0.26498  , -0.35035  ,  0.094222 , -0.11979  , -0.61102  ,
       -0.19592  ,  0.45899  ,  0.13644  , -0.021064 ,  0.078949 ,
        0.30338  , -0.53602  , -0.1295   , -0.059114 , -0.26958  ,
       -0.0594   , -0.30946  ,  0.24687  , -0.21573  , -0.065925 ,
        0.20449  , -0.30173  , -0.044619 , -0.57615  , -0.5174   ,
        0.34105  ,  0.051588 ,  0.14976  , -0.062579 , -0.23767  ,
        0.047176 ,  0.042792 , -0.42746  , -0.16878  ,  0.0021385,
       -0.010016 ,  0.34403  , -0.1048   ,  0.069748 , -0.01309  ,
        0.0051577, -0.49232  , -0.098993 ,  0.30827  ,  0.15935  ,
       -0.24144  , -0.53102  ,  0.039552 ,  0.22399  ,  0.36858  ,
        0.12315  , -0.22694  ,  0.64236  , -0.17091  ,  0.34865  ,
        0.13089  , -0.053819 ,  0.04895  , -0.24464  ,  0.43476  ,
        0.2408   ,  0.181    , -0.2394   ,  0.072814 ,  0.55916  ,
        0.10483  , -0.19562  , -0.061841 , -0.015999 , -0.038759 ,
        0.31525  , -0.36021  , -0.49485  , -0.33829  , -0.29361  ,
        0.20097  , -0.23206  , -0.17589  ,  0.24775  , -0.46655  ,
       -0.33826  ,  0.027942 , -0.025474 , -0.1404   , -0.74613  ,
        0.34042  , -0.25104  , -0.3126   ,  0.12158  ,  0.10922  ,
        0.013001 , -0.29323  , -0.2061   , -0.076021 , -0.66986  ,
        0.39975  ,  0.72107  , -0.25044  , -0.40863  ,  0.092596 ,
        0.1902   ,  0.12928  , -0.07334  , -0.093319 , -0.094435 ,
        0.027871 , -0.35144  , -0.053247 ,  0.026268 ,  0.66624  ,
       -0.0054439,  0.064125 , -0.009884 , -0.29844  ,  0.26926  ,
       -0.26021  , -0.3124   ,  0.40833  , -0.30415  , -0.53151  ,
       -0.02642  ,  0.21056  , -0.0099894,  0.07908  ,  0.026796 ,
       -0.089431 , -0.25709  , -0.086917 , -0.04149  ,  0.21066  ,
       -0.43214  , -0.43353  , -0.33388  , -0.45878  ,  0.13682  ,
        0.18526  ,  0.20796  , -0.24534  ,  0.52191  , -0.38073  ,
        0.15307  , -0.019677 ,  0.24239  ,  0.35903  ,  0.16858  ,
       -0.34968  , -0.15988  , -0.055486 , -0.32685  ,  0.17707  ,
        0.11939  , -0.23142  , -0.47032  , -0.20014  , -0.27967  ,
       -0.22426  ,  0.16883  , -0.36905  ,  0.21411  , -0.25568  ,
        0.15459  ,  0.39802  ,  0.055307 ,  0.30675  , -0.35765  ,
       -0.33988  , -0.57406  ,  0.11335  ,  0.096264 ,  0.16433  ,
        0.24512  , -0.16316  , -0.13592  , -0.058084 ,  0.23689  ,
       -0.19782  ,  0.018044 , -0.13635  ,  0.27039  , -0.15852  ,
        0.068199 , -0.4105   , -0.36368  ,  0.23394  ,  0.30523  ,
       -0.17253  , -0.031084 , -0.15681  , -0.28011  , -0.20914  ,
       -0.34704  ,  0.20816  ,  0.43446  , -0.10953  , -0.059969 ,
       -0.38553  ,  0.09257  , -0.30378  , -0.16808  ,  0.59528  ,
        0.48256  , -0.053949 , -0.019616 , -0.1764   , -0.40992  ,
        0.24213  , -0.22405  , -0.2381   ,  0.48239  ,  0.21925  ,
       -0.26817  ,  0.25413  , -0.062772 ,  0.37848  ,  0.20836  ,
       -0.0097771,  0.1063   ,  0.17081  ,  0.11206  , -0.19159  ,
       -0.17996  ,  0.025413 , -0.044346 ,  0.23861  ,  0.60771  ,
       -0.63148  ,  0.0066335,  0.27842  ,  0.31507  ,  0.14262  ,
       -0.1118   , -0.33488  ,  0.57944  , -0.079418 ,  0.116    ,
        0.10209  , -0.13081  , -0.055932 ,  0.21742  , -0.027025 ,
       -0.056908 , -0.067584 ,  0.14821  , -0.11014  ,  0.22236  ,
       -0.044174 , -0.11088  ,  0.15732  ,  0.11629  , -0.021035 ,
       -0.13935  ,  0.14022  , -0.39925  ,  0.11226  ,  0.017962 ,
       -0.26014  ,  0.029143 ,  0.088292 ,  0.77077  , -0.0041007,
       -0.51501  , -0.011301 ,  0.053099 , -0.079612 ,  0.014349 ,
        0.19904  , -0.32254  , -0.18502  ,  0.39045  , -0.092955 ,
       -0.47454  , -0.33162  ,  0.38562  ,  0.48883  , -0.064384 ,
       -0.28041  ,  0.08933  ,  0.13039  ,  0.092847 ,  0.6397   ,
       -0.34479  , -0.56525  ,  0.15083  ,  0.38102  ,  0.30173  ,
       -0.067981 ,  0.44332  ,  0.079129 , -0.14437  , -0.27496  ,
        0.096804 , -0.4821   , -0.10849  ,  0.3439   , -0.41897  ,
       -0.45853  ,  0.21763  ,  0.33992  ,  0.045876 , -0.012425 ])
```

[10]:

```
# load pre-processed file
project_data = pd.read_csv('/content/drive/My Drive/LSTM Assignment/
 ↪preprocessed_data.csv')
project_data.shape
```

[10]:

```
(109248, 9)
```

[0]:

```
project_data['remaining_input'] =␣
 ↪project_data['teacher_number_of_previously_posted_projects'] +␣
 ↪project_data['price']
```

[12]:

```
project_data.head()
```

[12]:

```
  school_state teacher_prefix  ...   price  remaining_input
0           ca            mrs  ...  725.05           778.05
1           ut             ms  ...  213.03           217.03
2           ca            mrs  ...  329.00           339.00
3           ga            mrs  ...  481.04           483.04
4           wa            mrs  ...   17.74            19.74

[5 rows x 10 columns]
```

[13]:

```
project_data.tail()
```

[13]:

```
       school_state teacher_prefix  ...   price  remaining_input
109243           hi            mrs  ...  143.36           148.36
109244           nm             ms  ...  268.57           271.57
109245           il            mrs  ...  399.00           399.00
109246           hi            mrs  ...  287.73           288.73
109247           ca            mrs  ...    5.50             7.50

[5 rows x 10 columns]
```

- Process following categorical features

- **Input_seq_total_text_data** — You have to give Total text data columns. After this use the Embedding layer to get word vectors. Use given predefined glove word vectors, don't train any word vectors. After this use LSTM and get the LSTM output and Flatten that output.
- **Input_school_state** — Give 'school_state' column as input to embedding layer and Train the Keras Embedding layer.
- **Project_grade_category** — Give 'project_grade_category' column as input to embedding layer and Train the Keras Embedding layer.
- **Input_clean_categories** — Give 'input_clean_categories' column as input to embedding layer and Train the Keras Embedding layer.
- **Input_clean_subcategories** — Give 'input_clean_subcategories' column as input to embedding layer and Train the Keras Embedding layer.
- **Input_teacher_prefix** — Give 'input_teacher_prefix' column as input to embedding layer and Train the Keras Embedding layer.
- **Input_remaining_teacher_number_of_previously_posted_projects._resource_summary_contains_nume** —concatenate remaining columns and add a Dense layer after that.

[14]:

```
# identify distinct values in school_state
print(project_data['school_state'].describe())
print(project_data['school_state'].unique())
```

```
count     109248
unique        51
top           ca
freq       15388
Name: school_state, dtype: object
['ca' 'ut' 'ga' 'wa' 'hi' 'il' 'oh' 'ky' 'sc' 'fl' 'mo' 'mi' 'ny' 'va'
 'md' 'tx' 'ms' 'nj' 'az' 'ok' 'pa' 'wv' 'nc' 'co' 'dc' 'ma' 'id' 'al'
 'me' 'tn' 'in' 'la' 'ct' 'ar' 'ks' 'or' 'wi' 'ia' 'sd' 'ak' 'mn' 'nm'
 'nv' 'mt' 'ri' 'nh' 'wy' 'ne' 'de' 'nd' 'vt']
```

[15]:

```
# identify distinct values in project_grade_category
print(project_data['project_grade_category'].describe())
print(project_data['project_grade_category'].unique())
```

```
count            109248
unique                4
top       grades_prek_2
freq              44225
Name: project_grade_category, dtype: object
['grades_prek_2' 'grades_3_5' 'grades_9_12' 'grades_6_8']
```

[16]:

```
# identify distinct values in clean_categories
print(project_data['clean_categories'].describe())
print(project_data['clean_categories'].unique())
```

```
count                 109248
unique                    51
top        literacy_language
freq                   23655
Name: clean_categories, dtype: object
['math_science' 'specialneeds' 'literacy_language' 'appliedlearning'
 'math_science history_civics' 'literacy_language math_science'
 'appliedlearning music_arts' 'math_science appliedlearning'
 'math_science literacy_language' 'history_civics literacy_language'
 'appliedlearning health_sports' 'math_science music_arts'
 'appliedlearning literacy_language' 'music_arts' 'health_sports'
 'literacy_language specialneeds' 'math_science specialneeds'
 'appliedlearning history_civics' 'appliedlearning specialneeds'
 'health_sports literacy_language' 'literacy_language music_arts'
 'history_civics math_science' 'specialneeds health_sports'
 'literacy_language history_civics' 'health_sports specialneeds'
 'history_civics music_arts' 'math_science health_sports'
 'music_arts specialneeds' 'specialneeds music_arts'
 'health_sports history_civics' 'history_civics'
 'health_sports appliedlearning' 'history_civics specialneeds'
 'appliedlearning math_science' 'health_sports music_arts'
 'literacy_language health_sports' 'literacy_language appliedlearning'
 'music_arts health_sports' 'music_arts appliedlearning'
 'music_arts history_civics' 'health_sports math_science'
 'history_civics appliedlearning' 'history_civics health_sports'
 'health_sports warmth care_hunger' 'history_civics warmth care_hunger'
 'math_science warmth care_hunger' 'specialneeds warmth care_hunger'
 'warmth care_hunger' 'literacy_language warmth care_hunger'
 'music_arts warmth care_hunger' 'appliedlearning warmth care_hunger']
```

[17]:

```
# identify distinct values in clean_subcategories
print(project_data['clean_subcategories'].describe())
print(project_data['clean_subcategories'].unique())
```

```
count        109248
unique          401
top        literacy
freq           9486
Name: clean_subcategories, dtype: object
['appliedsciences health_lifescience' 'specialneeds' 'literacy'
 'earlydevelopment' 'mathematics socialsciences' 'literacy mathematics'
 'appliedsciences history_geography' 'esl literacy'
 'appliedsciences mathematics' 'extracurricular visualarts'
 'appliedsciences earlydevelopment' 'environmentalscience literacy'
 'appliedsciences environmentalscience'
```

```
'history_geography literature_writing' 'literacy literature_writing'
'earlydevelopment gym_fitness' 'environmentalscience visualarts'
'environmentalscience mathematics' 'appliedsciences visualarts'
'earlydevelopment literacy' 'music' 'teamsports'
'health_lifescience mathematics' 'music performingarts'
'esl environmentalscience' 'college_careerprep esl'
'appliedsciences other' 'college_careerprep visualarts'
'literature_writing specialneeds' 'health_lifescience specialneeds'
'environmentalscience literature_writing' 'college_careerprep other'
'charactereducation socialsciences' 'literature_writing'
'earlydevelopment other' 'environmentalscience health_lifescience'
'other specialneeds' 'foreignlanguages' 'college_careerprep'
'literature_writing mathematics' 'health_wellness literature_writing'
'literacy specialneeds' 'literacy visualarts'
'health_lifescience visualarts' 'gym_fitness teamsports' 'mathematics'
'health_wellness teamsports' 'appliedsciences civics_government'
'economics mathematics' 'esl literature_writing'
'environmentalscience socialsciences' 'health_wellness'
'health_lifescience literature_writing' 'mathematics specialneeds'
'specialneeds teamsports' 'earlydevelopment visualarts'
'literacy socialsciences' 'esl' 'health_wellness specialneeds'
'history_geography music' 'earlydevelopment specialneeds' 'gym_fitness'
'appliedsciences literacy' 'communityservice earlydevelopment' 'other'
'charactereducation' 'esl mathematics' 'literacy performingarts'
'literature_writing visualarts' 'health_lifescience health_wellness'
'earlydevelopment literature_writing' 'literacy music'
'gym_fitness health_wellness' 'visualarts' 'charactereducation literacy'
'mathematics visualarts' 'music specialneeds' 'health_lifescience'
'history_geography literacy' 'literature_writing socialsciences'
'specialneeds visualarts' 'appliedsciences' 'environmentalscience'
'environmentalscience history_geography' 'health_wellness socialsciences'
'environmentalscience health_wellness' 'performingarts'
'appliedsciences literature_writing' 'extracurricular teamsports'
'charactereducation earlydevelopment' 'appliedsciences socialsciences'
'civics_government economics' 'extracurricular' 'health_wellness other'
'history_geography specialneeds' 'health_wellness literacy'
'communityservice extracurricular' 'charactereducation specialneeds'
'extracurricular literacy' 'environmentalscience specialneeds'
'college_careerprep literacy' 'esl specialneeds'
'appliedsciences specialneeds' 'music visualarts'
'college_careerprep communityservice' 'health_lifescience literacy'
'college_careerprep environmentalscience' 'charactereducation teamsports'
'financialliteracy mathematics' 'nutritioneducation visualarts'
'history_geography' 'foreignlanguages mathematics'
'literacy nutritioneducation' 'earlydevelopment health_wellness'
'charactereducation college_careerprep'
'history_geography socialsciences' 'appliedsciences college_careerprep'
'literacy other' 'literature_writing performingarts' 'other visualarts'
```

```
'college_careerprep specialneeds' 'college_careerprep literature_writing'
'esl foreignlanguages' 'nutritioneducation'
'charactereducation health_wellness' 'communityservice literacy'
'esl earlydevelopment' 'foreignlanguages literacy'
'history_geography visualarts' 'socialsciences visualarts'
'performingarts visualarts' 'appliedsciences foreignlanguages'
'civics_government literacy' 'esl health_lifescience'
'appliedsciences extracurricular' 'literature_writing parentinvolvement'
'esl history_geography' 'health_lifescience history_geography'
'extracurricular other' 'charactereducation other'
'charactereducation literature_writing' 'mathematics music'
'communityservice environmentalscience' 'communityservice visualarts'
'socialsciences' 'mathematics other' 'parentinvolvement visualarts'
'foreignlanguages literature_writing'
'charactereducation communityservice' 'charactereducation mathematics'
'health_wellness visualarts' 'extracurricular music'
'civics_government environmentalscience'
'health_lifescience nutritioneducation' 'appliedsciences music'
'esl socialsciences' 'appliedsciences parentinvolvement'
'charactereducation visualarts' 'foreignlanguages performingarts'
'literature_writing music' 'communityservice other'
'civics_government history_geography'
'appliedsciences charactereducation' 'performingarts teamsports'
'college_careerprep mathematics' 'health_wellness nutritioneducation'
'health_lifescience socialsciences' 'gym_fitness performingarts'
'college_careerprep history_geography'
'environmentalscience extracurricular' 'college_careerprep teamsports'
'esl visualarts' 'extracurricular gym_fitness'
'college_careerprep extracurricular' 'esl music'
'literature_writing other' 'extracurricular socialsciences'
'earlydevelopment environmentalscience' 'nutritioneducation other'
'extracurricular literature_writing' 'civics_government socialsciences'
'earlydevelopment music' 'music other' 'extracurricular specialneeds'
'performingarts socialsciences' 'communityservice specialneeds'
'charactereducation extracurricular'
'earlydevelopment health_lifescience' 'economics socialsciences'
'college_careerprep economics' 'gym_fitness literature_writing'
'communityservice' 'environmentalscience nutritioneducation'
'earlydevelopment mathematics' 'gym_fitness literacy'
'health_wellness mathematics' 'gym_fitness specialneeds'
'charactereducation environmentalscience' 'mathematics performingarts'
'college_careerprep health_wellness' 'college_careerprep performingarts'
'literacy parentinvolvement' 'economics other'
'history_geography mathematics' 'college_careerprep earlydevelopment'
'appliedsciences gym_fitness' 'appliedsciences teamsports'
'health_wellness history_geography'
'college_careerprep health_lifescience'
'charactereducation history_geography' 'socialsciences specialneeds'
```

```
'mathematics parentinvolvement' 'financialliteracy specialneeds'
'extracurricular mathematics' 'civics_government health_lifescience'
'parentinvolvement' 'health_wellness performingarts' 'esl other'
'environmentalscience other' 'earlydevelopment performingarts'
'communityservice performingarts' 'appliedsciences esl'
'communityservice history_geography' 'communityservice mathematics'
'health_lifescience music' 'economics literacy'
'college_careerprep financialliteracy' 'charactereducation music'
'college_careerprep music' 'college_careerprep parentinvolvement'
'economics financialliteracy' 'literacy teamsports'
'foreignlanguages specialneeds' 'extracurricular health_lifescience'
'extracurricular health_wellness' 'other socialsciences'
'nutritioneducation teamsports' 'civics_government'
'financialliteracy literacy' 'civics_government literature_writing'
'foreignlanguages other' 'civics_government visualarts'
'charactereducation health_lifescience' 'gym_fitness other'
'communityservice parentinvolvement' 'teamsports visualarts'
'foreignlanguages visualarts' 'other parentinvolvement'
'music teamsports' 'appliedsciences health_wellness'
'economics history_geography' 'earlydevelopment parentinvolvement'
'communityservice health_lifescience'
'foreignlanguages history_geography' 'history_geography other'
'charactereducation parentinvolvement' 'esl performingarts'
'communityservice literature_writing' 'charactereducation esl'
'civics_government communityservice' 'appliedsciences communityservice'
'parentinvolvement specialneeds' 'civics_government college_careerprep'
'communityservice health_wellness' 'charactereducation civics_government'
'esl health_wellness' 'health_lifescience other' 'health_wellness music'
'gym_fitness mathematics' 'earlydevelopment extracurricular'
'music socialsciences' 'economics' 'college_careerprep socialsciences'
'earlydevelopment socialsciences' 'parentinvolvement socialsciences'
'financialliteracy visualarts' 'performingarts specialneeds'
'health_lifescience parentinvolvement' 'foreignlanguages socialsciences'
'civics_government specialneeds' 'earlydevelopment nutritioneducation'
'civics_government financialliteracy' 'gym_fitness nutritioneducation'
'history_geography performingarts' 'esl financialliteracy'
'charactereducation performingarts' 'communityservice socialsciences'
'gym_fitness visualarts' 'foreignlanguages music'
'appliedsciences economics' 'charactereducation financialliteracy'
'literature_writing nutritioneducation' 'extracurricular performingarts'
'civics_government mathematics' 'environmentalscience parentinvolvement'
'mathematics nutritioneducation' 'environmentalscience foreignlanguages'
'college_careerprep nutritioneducation' 'gym_fitness health_lifescience'
'health_lifescience teamsports' 'gym_fitness music'
'nutritioneducation specialneeds' 'appliedsciences performingarts'
'esl nutritioneducation' 'foreignlanguages health_wellness'
'mathematics teamsports' 'civics_government esl'
'environmentalscience gym_fitness' 'gym_fitness history_geography'
```

```
'health_wellness parentinvolvement' 'civics_government extracurricular'
'financialliteracy' 'financialliteracy health_wellness'
'earlydevelopment history_geography' 'earlydevelopment teamsports'
'appliedsciences nutritioneducation' 'charactereducation gym_fitness'
'environmentalscience financialliteracy'
'earlydevelopment foreignlanguages' 'college_careerprep gym_fitness'
'communityservice financialliteracy' 'extracurricular nutritioneducation'
'nutritioneducation socialsciences' 'economics literature_writing'
'literature_writing teamsports' 'communityservice nutritioneducation'
'civics_government health_wellness' 'college_careerprep foreignlanguages'
'extracurricular history_geography' 'communityservice esl'
'economics health_lifescience' 'gym_fitness parentinvolvement'
'environmentalscience performingarts' 'environmentalscience music'
'economics environmentalscience' 'esl parentinvolvement'
'charactereducation foreignlanguages' 'esl extracurricular'
'health_wellness warmth care_hunger' 'economics specialneeds'
'esl gym_fitness' 'charactereducation nutritioneducation'
'civics_government performingarts' 'extracurricular parentinvolvement'
'health_lifescience performingarts' 'history_geography teamsports'
'economics music' 'civics_government foreignlanguages'
'economics foreignlanguages' 'financialliteracy history_geography'
'earlydevelopment economics' 'foreignlanguages gym_fitness'
'economics nutritioneducation' 'communityservice music'
'foreignlanguages health_lifescience' 'other teamsports'
'history_geography warmth care_hunger' 'extracurricular foreignlanguages'
'communityservice gym_fitness' 'music parentinvolvement'
'earlydevelopment financialliteracy' 'gym_fitness socialsciences'
'socialsciences teamsports' 'health_lifescience warmth care_hunger'
'other performingarts' 'communityservice economics'
'specialneeds warmth care_hunger' 'mathematics warmth care_hunger'
'warmth care_hunger' 'literacy warmth care_hunger'
'appliedsciences financialliteracy'
'nutritioneducation warmth care_hunger'
'environmentalscience warmth care_hunger' 'visualarts warmth care_hunger'
'financialliteracy other' 'charactereducation warmth care_hunger'
'civics_government teamsports' 'literature_writing warmth care_hunger'
'earlydevelopment warmth care_hunger' 'other warmth care_hunger'
'economics visualarts' 'charactereducation economics'
'appliedsciences warmth care_hunger'
'parentinvolvement warmth care_hunger' 'gym_fitness warmth care_hunger'
'esl teamsports' 'environmentalscience teamsports'
'financialliteracy literature_writing'
'civics_government nutritioneducation' 'financialliteracy socialsciences'
'parentinvolvement performingarts' 'civics_government parentinvolvement'
'history_geography parentinvolvement' 'extracurricular financialliteracy'
'financialliteracy health_lifescience' 'financialliteracy performingarts'
'financialliteracy parentinvolvement'
'financialliteracy foreignlanguages' 'esl economics'
```

```
'parentinvolvement teamsports' 'college_careerprep warmth care_hunger']
```

[18]:
```
# identify distinct values in teacher_prefix
project_data['teacher_prefix'].describe()
project_data['teacher_prefix'].values
```

[18]:
```
array(['mrs', 'ms', 'mrs', ..., 'mrs', 'mrs', 'mrs'], dtype=object)
```

[19]:
```
# droping class label data
y = project_data['project_is_approved'].values
project_data.drop(['project_is_approved'], axis=1, inplace=True)
X = project_data
project_data.shape
```

[19]:
```
(109248, 9)
```

[20]:
```
# Split Train, CV and Test data (64, 16, 20)
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2,␣
 ↪stratify=y)
X_train, X_cv, y_train, y_cv = train_test_split(X_train, y_train, test_size=0.
 ↪2, stratify=y_train)

print('Train Data Set', X_train.shape, y_train.shape)
print('Cross Validate Data Set', X_cv.shape, y_cv.shape)
print('Test Data Set', X_test.shape, y_test.shape)
```

```
Train Data Set (69918, 9) (69918,)
Cross Validate Data Set (17480, 9) (17480,)
Test Data Set (21850, 9) (21850,)
```

[21]:
```
print('Train Data Set', X_train.shape, y_train.shape)
print('Cross Validate Data Set', X_cv.shape, y_cv.shape)
print('Test Data Set', X_test.shape, y_test.shape)
print('*'*100)
```

```
Train Data Set (69918, 9) (69918,)
Cross Validate Data Set (17480, 9) (17480,)
Test Data Set (21850, 9) (21850,)
********************************************************************************
********************
```

[22]:
```
# build logic to tokenize input
# Steps below
# 1. Count frequencies of each word
# 2. Sort frequencies by desc order
# 3. Assign rank to frequencies
# 4. Tokenize words in the main categorical variable
sent_list = [
    'The phone is very fast',
    'The phone is not bad',
    'I have good phone',
]

# count no. of words and assign it to dictionary
words_dict = {}
for sent in sent_list:
    words = sent.split()
    for i in words:
        if(i in words_dict):
            words_dict[i] += 1
        else:
            words_dict[i] = 1

print(words_dict)
print('*'*100)

# sort dictionary by their frequencies
sorted_dict = sorted(words_dict.items(), key=lambda x: x[1], reverse=True)
print(sorted_dict)
print('*'*100)


# assign rank to each word
rank = 1
final_dict = {}
for item in sorted_dict:
    item = list(item)
    final_dict[item[0]] = rank
    rank += 1

print(final_dict)
print('*'*100)

# finally convert main sentences into tokens
tokenize_list = []
for sent in sent_list:
    words = sent.split()
    tokenize_sublist = []
    for item in words:
        if(item in final_dict):
            tokenize_sublist.append(final_dict[item])

    tokenize_list.append(tokenize_sublist)

print(tokenize_list)
print('*'*100)
```

```
{'The': 2, 'phone': 3, 'is': 2, 'very': 1, 'fast': 1, 'not': 1, 'bad': 1, 'I':
1, 'have': 1, 'good': 1}
********************************************************************************
********************
[('phone', 3), ('The', 2), ('is', 2), ('very', 1), ('fast', 1), ('not', 1),
('bad', 1), ('I', 1), ('have', 1), ('good', 1)]
********************************************************************************
********************
{'phone': 1, 'The': 2, 'is': 3, 'very': 4, 'fast': 5, 'not': 6, 'bad': 7, 'I':
8, 'have': 9, 'good': 10}
********************************************************************************
********************
[[2, 1, 3, 4, 5], [2, 1, 3, 6, 7], [8, 9, 10, 1]]
********************************************************************************
********************
```

[0]:

```
def fit_transform_train_data(train_data):
    bag_of_words = CountVectorizer(lowercase= False)
    bow_words = bag_of_words.fit_transform(train_data)

    # Store calculated frequencies in the dictionaries
    freqs = bow_words.sum(axis=0).A1
    index = freqs.argsort()
    words = bag_of_words.get_feature_names()
    #print(freqs, index, words)

    rank_dict = {}
    rank = 1
    for item in index[::-1]:
        feature_name = words[item]
        rank_dict[feature_name] = rank
        rank += 1

#     print(rank_dict)

    return [words, rank_dict]


def transform_data(data, rank_dict):
    # finally convert main sentences into tokens
    tokenize_list = []
    for sent in data:
        words = sent.split()
        tokenize_sublist = []
        for item in words:
            if(item in rank_dict):
                tokenize_sublist.append(rank_dict[item])

        tokenize_list.append(tokenize_sublist)


    return tokenize_list
```

[24]:

```
# Test above implementation
features, rank_dict = fit_transform_train_data(project_data['school_state'])
print(features, rank_dict)
tokenize_data = transform_data(project_data['school_state'], rank_dict)
print(project_data['school_state'][0])
print(tokenize_data[0])
print(project_data['school_state'][10])
print(tokenize_data[10])
print(len(features))
```

```
['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'] {'ca':
1, 'tx': 2, 'ny': 3, 'fl': 4, 'nc': 5, 'il': 6, 'ga': 7, 'sc': 8, 'mi': 9, 'pa':
10, 'in': 11, 'mo': 12, 'oh': 13, 'la': 14, 'ma': 15, 'wa': 16, 'ok': 17, 'nj':
18, 'az': 19, 'va': 20, 'wi': 21, 'al': 22, 'ut': 23, 'tn': 24, 'ct': 25, 'md':
26, 'nv': 27, 'ms': 28, 'ky': 29, 'or': 30, 'mn': 31, 'co': 32, 'ar': 33, 'id':
34, 'ia': 35, 'ks': 36, 'nm': 37, 'dc': 38, 'hi': 39, 'me': 40, 'wv': 41, 'nh':
42, 'ak': 43, 'de': 44, 'ne': 45, 'sd': 46, 'ri': 47, 'mt': 48, 'nd': 49, 'wy':
50, 'vt': 51}
ca
[1]
il
[6]
51
```

[25]:

```
# One hot encoding of Categorical Feature
# - school_state : categorical data
(school_state_features, rank_dict) =␣
 ↪fit_transform_train_data(X_train['school_state'].values)# Fit has to happen␣
 ↪only on train data

X_train_school_state_ohe = transform_data(X_train['school_state'].values,␣
 ↪rank_dict)
X_cv_school_state_ohe = transform_data(X_cv['school_state'].values, rank_dict)
X_test_school_state_ohe = transform_data(X_test['school_state'].values,␣
 ↪rank_dict)
```

```
print(len(X_train_school_state_ohe), y_train.shape)
print(len(X_cv_school_state_ohe), y_cv.shape)
print(len(X_test_school_state_ohe), y_test.shape)
print(school_state_features)
print(len(school_state_features))
print('*'*100)
```

```
69918 (69918,)
17480 (17480,)
21850 (21850,)
['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']
51
********************************************************************************
********************
```

[26]:

```
# print(X_train['project_grade_category'])
# One hot encoding of Categorical Feature
# - project_grade_category : categorical data
# Convert one hot encoding for project grade category
(project_grade_category_features, rank_dict) = ␣
 ↪fit_transform_train_data(X_train['project_grade_category'].values)# Fit has␣
 ↪to happen only on train data

X_train_project_grade_category_ohe =␣
 ↪transform_data(X_train['project_grade_category'].values, rank_dict)
X_cv_project_grade_category_ohe = transform_data(X_cv['project_grade_category'].
 ↪values, rank_dict)
X_test_project_grade_category_ohe =␣
 ↪transform_data(X_test['project_grade_category'].values, rank_dict)

print(len(X_train_project_grade_category_ohe), y_train.shape)
print(len(X_cv_project_grade_category_ohe), y_cv.shape)
print(len(X_test_project_grade_category_ohe), y_test.shape)
# print(project_grade_category_features)
print(len(project_grade_category_features))
print('*'*100)
```

```
69918 (69918,)
17480 (17480,)
21850 (21850,)
4
********************************************************************************
********************
```

[27]:

```
# One hot encoding of Categorical Feature
# - clean_categories : categorical data
# print(X_train['clean_categories'].describe())
(clean_categories_features, rank_dict) = ␣
 ↪fit_transform_train_data(X_train['clean_categories'].values)# Fit has to␣
 ↪happen only on train data

# print(rank_dict)

X_train_clean_categories_ohe = transform_data(X_train['clean_categories'].
 ↪values, rank_dict)
X_cv_clean_categories_ohe = transform_data(X_cv['clean_categories'].values,␣
 ↪rank_dict)
X_test_clean_categories_ohe = transform_data(X_test['clean_categories'].values,␣
 ↪rank_dict)

print(len(X_train_clean_categories_ohe), y_train.shape)
print(len(X_cv_clean_categories_ohe), y_cv.shape)
print(len(X_test_clean_categories_ohe), y_test.shape)
print(clean_categories_features)
print(len(clean_categories_features))
print('*'*100)
```

```
69918 (69918,)
17480 (17480,)
21850 (21850,)
['appliedlearning', 'care_hunger', 'health_sports', 'history_civics',
'literacy_language', 'math_science', 'music_arts', 'specialneeds', 'warmth']
9
********************************************************************************
********************
```

[28]:

```
# One hot encoding of Categorical Feature
# - clean_subcategories : categorical data
(clean_subcategories_features, rank_dict) = ␣
 ↪fit_transform_train_data(X_train['clean_subcategories'].values)# Fit has to␣
 ↪happen only on train data

X_train_clean_subcategories_ohe = transform_data(X_train['clean_subcategories'].
 ↪values, rank_dict)
X_cv_clean_subcategories_ohe = transform_data(X_cv['clean_subcategories'].
 ↪values, rank_dict)
X_test_clean_subcategories_ohe = transform_data(X_test['clean_subcategories'].
 ↪values, rank_dict)

print(len(X_train_clean_subcategories_ohe), y_train.shape)
print(len(X_cv_clean_subcategories_ohe), y_cv.shape)
print(len(X_test_clean_subcategories_ohe), y_test.shape)
print(clean_subcategories_features)
print(len(clean_subcategories_features))
print('*'*100)
```

```
69918 (69918,)
17480 (17480,)
21850 (21850,)
['appliedsciences', 'care_hunger', 'charactereducation', 'civics_government',
'college_careerprep', 'communityservice', 'earlydevelopment', 'economics',
'environmentalscience', 'esl', 'extracurricular', 'financialliteracy',
'foreignlanguages', 'gym_fitness', 'health_lifescience', 'health_wellness',
'history_geography', 'literacy', 'literature_writing', 'mathematics', 'music',
'nutritioneducation', 'other', 'parentinvolvement', 'performingarts',
'socialsciences', 'specialneeds', 'teamsports', 'visualarts', 'warmth']
30
********************************************************************************
********************
```

[29]:

```
# One hot encoding of Categorical Feature
# - teacher_prefix : categorical data
# print(X_train['teacher_prefix'])
(teacher_prefix_features, rank_dict) = ␣
 ↪fit_transform_train_data(X_train['teacher_prefix'].values)# Fit has to␣
 ↪happen only on train data

X_train_teacher_prefix_ohe = transform_data(X_train['teacher_prefix'].values,␣
 ↪rank_dict)
X_cv_teacher_prefix_ohe = transform_data(X_cv['teacher_prefix'].values,␣
 ↪rank_dict)
X_test_teacher_prefix_ohe = transform_data(X_test['teacher_prefix'].values,␣
 ↪rank_dict)

print(len(X_train_teacher_prefix_ohe), y_train.shape)
print(len(X_cv_teacher_prefix_ohe), y_cv.shape)
print(len(X_test_teacher_prefix_ohe), y_test.shape)
print(teacher_prefix_features)
print(len(teacher_prefix_features))
print('*'*100)
```

```
69918 (69918,)
17480 (17480,)
21850 (21850,)
['dr', 'mr', 'mrs', 'ms', 'teacher']
5
********************************************************************************
********************
```

[0]:

```
# # convert review word as well i.e. tokenize review text
# # - text : text data
# print(X_train.shape, y_train.shape)
# print(X_cv.shape, y_cv.shape)
# print(X_test.shape, y_test.shape)

# print("*"*100)

# (easy_features, rank_dict)  = fit_transform_train_data(X_train['essay'].
 ↪values) # fit has to happen only on train data

# # we use the fitted CountVectorizer to convert the text to vector
# X_train_essay = transform_data(X_train['essay'].values, rank_dict)
# X_cv_essay = transform_data(X_cv['essay'].values, rank_dict)
# X_test_essay = transform_data(X_test['essay'].values, rank_dict)

# print("After vectorizations")
# print(len(X_train_essay), y_train.shape)
# print(len(X_cv_essay), y_cv.shape)
# print(len(X_test_essay), y_test.shape)
# # print(easy_features)
# print(len(easy_features))
# print("*"*100)
```

[0]:

```
#https://machinelearningmastery.com/
 ↪use-word-embedding-layers-deep-learning-keras/
def padded(encoded_docs):
  max_length = 250
  padded_docs = pad_sequences(encoded_docs, maxlen=max_length, padding='post')
  return padded_docs
```

[0]:

```
#https://stackoverflow.com/posts/51956230/revisions
t = Tokenizer()
t.fit_on_texts(X_train['essay'])
vocab_size = len(t.word_index) + 1
# integer encode the documents
encoded_docs = t.texts_to_sequences(X_train['essay'])
X_train_essay = padded(encoded_docs)
```

[0]:

```
#t = Tokenizer()
#t.fit_on_texts(x_cross.cleaned_essay)
#vocab_size = len(t.word_index) + 1
# integer encode the documents
encoded_docs = t.texts_to_sequences(X_cv['essay'])
X_cv_essay = padded(encoded_docs)
```

[0]:

```
#t = Tokenizer()
#t.fit_on_texts(x_test.cleaned_essay)
#vocab_size = len(t.word_index) + 1
```

```
# integer encode the documents
encoded_docs = t.texts_to_sequences(X_test['essay'])
X_test_essay = padded(encoded_docs)
```

[0]:
```
embeddings_index = dict()
f = open('/content/drive/My Drive/LSTM Output/glove.6B.300d.txt')
for line in f:
        values = line.split()
        word = values[0]
        coefs = np.asarray(values[1:], dtype='float32')
        embeddings_index[word] = coefs
f.close()
```

[0]:
```
embedding_matrix = np.zeros((vocab_size, 300))
for word, i in t.word_index.items():
        embedding_vector = embeddings_index.get(word)
        if embedding_vector is not None:
                embedding_matrix[i] = embedding_vector
```

[37]:
```
print("embedding matrix shape",embedding_matrix.shape)
```

```
embedding matrix shape (47268, 300)
```

[0]:
```
#converting class labels to categorical variables
from keras.utils import to_categorical

y_train = to_categorical(y_train)
y_cv = to_categorical(y_cv)
y_test = to_categorical(y_test)
```

[39]:
```
y_train
```

[39]:
```
array([[0., 1.],
       [0., 1.],
       [0., 1.],
       ...,
       [0., 1.],
       [0., 1.],
       [0., 1.]], dtype=float32)
```

[0]:
```
class_weight = compute_class_weight("balanced", classes= np.unique(y),y=y)
```

[41]:
```
class_weight
```

[41]:
```
array([3.30214001, 0.58921753])
```

Sequence Padding Text Data

[0]:
```
# #padding zeros at the begining of each easy to make max len as 250
# max_easy_length = 250
# X_train_essay = pad_sequences(X_train_essay, maxlen=max_easy_length)
# X_cv_essay = pad_sequences(X_cv_essay, maxlen=max_easy_length)
# X_test_essay = pad_sequences(X_test_essay, maxlen=max_easy_length)
# print(X_train_essay.shape)
# print(X_train_essay[0])
# print(X_train_essay[1])
# print(X_cv_essay.shape)
# print(X_cv_essay[0])
# # print(X_cv_essay[1])
# print(X_test_essay.shape)
# print(X_test_essay[0])
# # print(X_test_essay[1])
```

[0]:

```
max_length_categorical_variable = 1
```

[44]:

```
X_train_school_state_ohe = pad_sequences(X_train_school_state_ohe,␣
 ↪maxlen=max_length_categorical_variable)
X_cv_school_state_ohe = pad_sequences(X_cv_school_state_ohe,␣
 ↪maxlen=max_length_categorical_variable)
X_test_school_state_ohe = pad_sequences(X_test_school_state_ohe,␣
 ↪maxlen=max_length_categorical_variable)
print(X_train_school_state_ohe.shape)
print(X_train_school_state_ohe[0])
print(X_train_school_state_ohe[1])
print(X_cv_school_state_ohe.shape)
print(X_cv_school_state_ohe[0])
print(X_test_school_state_ohe.shape)
print(X_test_school_state_ohe[0])
```

```
(69918, 1)
[7]
[17]
(17480, 1)
[15]
(21850, 1)
[7]
```

[45]:

```
X_train_project_grade_category_ohe =␣
 ↪pad_sequences(X_train_project_grade_category_ohe,␣
 ↪maxlen=max_length_categorical_variable)
X_cv_project_grade_category_ohe =␣
 ↪pad_sequences(X_cv_project_grade_category_ohe,␣
 ↪maxlen=max_length_categorical_variable)
X_test_project_grade_category_ohe =␣
 ↪pad_sequences(X_test_project_grade_category_ohe,␣
 ↪maxlen=max_length_categorical_variable)
print(X_train_project_grade_category_ohe.shape)
print(X_train_project_grade_category_ohe[0])
print(X_train_project_grade_category_ohe[1])
print(X_cv_project_grade_category_ohe.shape)
print(X_cv_project_grade_category_ohe[0])
print(X_test_school_state_ohe.shape)
print(X_test_school_state_ohe[0])
```

```
(69918, 1)
[1]
[1]
(17480, 1)
[1]
(21850, 1)
[7]
```

[46]:
```
X_train_clean_categories_ohe = pad_sequences(X_train_clean_categories_ohe,␣
 ↪maxlen=max_length_categorical_variable)
X_cv_clean_categories_ohe = pad_sequences(X_cv_clean_categories_ohe,␣
 ↪maxlen=max_length_categorical_variable)
X_test_clean_categories_ohe = pad_sequences(X_test_clean_categories_ohe,␣
 ↪maxlen=max_length_categorical_variable)
print(X_train_clean_categories_ohe.shape)
print(X_train_clean_categories_ohe[0])
print(X_train_clean_categories_ohe[1])
print(X_cv_clean_categories_ohe.shape)
print(X_cv_clean_categories_ohe[0])
print(X_test_clean_categories_ohe.shape)
print(X_test_clean_categories_ohe[0])
```

```
(69918, 1)
[2]
[5]
(17480, 1)
[1]
(21850, 1)
[3]
```

[47]:
```
X_train_clean_subcategories_ohe =␣
 ↪pad_sequences(X_train_clean_subcategories_ohe,␣
 ↪maxlen=max_length_categorical_variable)
X_cv_clean_subcategories_ohe = pad_sequences(X_cv_clean_subcategories_ohe,␣
 ↪maxlen=max_length_categorical_variable)
X_test_clean_subcategories_ohe = pad_sequences(X_test_clean_subcategories_ohe,␣
 ↪maxlen=max_length_categorical_variable)
print(X_train_clean_subcategories_ohe.shape)
print(X_train_clean_subcategories_ohe[0])
print(X_train_clean_subcategories_ohe[1])
print(X_cv_clean_subcategories_ohe.shape)
print(X_cv_clean_subcategories_ohe[0])
print(X_test_clean_subcategories_ohe.shape)
print(X_test_clean_subcategories_ohe[0])
```

```
(69918, 1)
[2]
[12]
(17480, 1)
[1]
(21850, 1)
[6]
```

[48]:

```
X_train_teacher_prefix_ohe = pad_sequences(X_train_teacher_prefix_ohe,␣
 ↪maxlen=max_length_categorical_variable)
X_cv_teacher_prefix_ohe = pad_sequences(X_cv_teacher_prefix_ohe,␣
 ↪maxlen=max_length_categorical_variable)
X_test_teacher_prefix_ohe = pad_sequences(X_test_teacher_prefix_ohe,␣
 ↪maxlen=max_length_categorical_variable)
print(X_train_teacher_prefix_ohe.shape)
print(X_train_teacher_prefix_ohe[0])
print(X_train_teacher_prefix_ohe[1])
print(X_cv_teacher_prefix_ohe.shape)
print(X_cv_teacher_prefix_ohe[0])
print(X_test_teacher_prefix_ohe.shape)
print(X_test_teacher_prefix_ohe[0])
```

```
(69918, 1)
[1]
[2]
(17480, 1)
[2]
(21850, 1)
[1]
```

[0]:

```
#AUC score
def auc( y_true, y_pred ) :
    score = tf.py_func( lambda y_true, y_pred : roc_auc_score( y_true, y_pred,␣
 ↪average='macro', sample_weight=None).astype('float32'),
                        [y_true, y_pred],
                        'float32',
                        stateful=True,
                        name='sklearnAUC' )
    return score

# def auc(y_true, y_pred):
#   return tf.py_func(roc_auc_score, (y_true, y_pred), tf.double)
# model.
 ↪compile(loss='binary_crossentropy',optimizer='adam',metrics=['accuracy',auc])
# model.fit(...,validation_A=(x_val,y_val),..,batch_size=300)
```

[0]:
```
# #Creating a matrix with rows as words and columns with 300 dim vectors for␣
 ↪each word
# def embedding_mat(word_index, embedding_dim = 300):
#     embedding_matrix = np.zeros((len(word_index) + 1, embedding_dim))
#     # initialize counter
#     cnt = 0
#     for i, sent in word_index.items():
#         vector = np.zeros(embedding_dim) # as word vectors are of zero length
#         words = sent.split()
#         for word in words:
#           # check word in glove vector
#           embedding_vector = glovevector.get(word)
#           if embedding_vector is not None:
#             #sum words weights
#             vector += embedding_vector

#         # words not found in embedding index will be all-zeros.
#         embedding_matrix[cnt] = vector
#         cnt += 1

#     return embedding_matrix
```

[0]:
```
# X_train_essay_mat = embedding_mat(X_train['essay'])
# X_train_essay_mat.shape
```

[0]:
```
X_train_essay_mat = embedding_matrix
```

Assignment 1

- input_dim: This is the size of the vocabulary in the text data. For example, if your data is integer encoded to values between 0-10, then the size of the vocabulary would be 11 words.
- output_dim: This is the size of the vector space in which words will be embedded. It defines the size of the output vectors from this layer for each word. For example, it could be 32 or 100 or even larger. Test different values for your problem.
- input_length: This is the length of input sequences, as you would define for any input layer of a Keras model. For example, if all of your input documents are comprised of 1000 words, this would be 1000.

[54]:
```
from keras.layers import LeakyReLU

# Create model 1

#input 1 essay text
input1 = Input(shape=(250,))
# x1 = Embedding(input_dim=49042,output_dim= 300)(input1)
```

```
x1 = Embedding(input_dim=47268,output_dim=␣
 ↪300,weights=[X_train_essay_mat],trainable=False)(input1)
x1 = SpatialDropout1D(0.3)(x1)
x1 = CuDNNLSTM(128,return_sequences=True)(x1)
# x1 = LSTM(128,return_sequences=True)(x1)
x1 = Flatten()(x1)

#https://medium.com/@davidheffernan_99410/
 ↪an-introduction-to-using-categorical-embeddings-ee686ed7e7f9
cat_vars =␣
 ↪["teacher_prefix","school_state","project_grade_category","clean_categories","clean_subcate
cat_sizes = {}
cat_embsizes = {}
for cat in cat_vars:
    cat_sizes[cat] = X_train[cat].nunique()
    cat_embsizes[cat] = min(50, cat_sizes[cat]//2+1)

# input 2 school_state 51
input2 = Input(shape=(1,))
x2 = Embedding(input_dim=cat_sizes['school_state']+1,␣
 ↪output_dim=cat_embsizes['school_state'])(input2)#input_dim=52
x2 = Flatten()(x2)

# input 3 project_grade_cat 4
input3 = Input(shape=(1,))
x3 = Embedding(input_dim=cat_sizes['project_grade_category']+1,␣
 ↪output_dim=cat_embsizes['project_grade_category'])(input3)
x3 = Flatten()(x3)

# input 4 clean_categories 9
input4 = Input(shape=(1,))
x4 = Embedding(input_dim=cat_sizes['clean_categories']+1,␣
 ↪output_dim=cat_embsizes['clean_categories'])(input4)
x4 = Flatten()(x4)

# input 5 clean_subcategories 30
input5 = Input(shape=(1,))
x5 = Embedding(input_dim=cat_sizes['clean_subcategories']+1,␣
 ↪output_dim=cat_embsizes['clean_subcategories'])(input5)
x5 = Flatten()(x5)

# input 6 teacher_prefix 5
input6 = Input(shape=(1,))
x6 = Embedding(input_dim=cat_sizes['teacher_prefix']+1,␣
 ↪output_dim=cat_embsizes['teacher_prefix'])(input6)
x6 = Flatten()(x6)
```

```
# print(cat_concat)

#input 7 remaining inout
input7 = Input(shape=(1,))
x7 = Dense(16,kernel_initializer=he_normal(),kernel_regularizer=l2(0.
 ↪0001))(input7)
x7 = LeakyReLU()(x7)

concat = concatenate([x1, x2, x3, x4, x5, x6, x7])

x = Dense(128,kernel_initializer=he_normal(),kernel_regularizer=l2(0.
 ↪0001))(concat)
x = Dropout(0.5)(x)
x = LeakyReLU()(x)
x = Dense(64,kernel_initializer=he_normal(),kernel_regularizer=l2(0.0001))(x)
x = Dropout(0.5)(x)
x = BatchNormalization()(x)
x = LeakyReLU()(x)
x = Dense(32,kernel_initializer=he_normal(),kernel_regularizer=l2(0.0001))(x)
x = LeakyReLU()(x)
# x = Dropout(0.5)(x)
output = Dense(2, activation = 'softmax')(x)

# create model with seven inputs
model = Model([input1,input2,input3,input4,input5,input6,input7], output)#
model.run_eagerly = True
tensorboard = TensorBoard(log_dir='/content/drive/My Drive/LSTM Output/logs/{}'.
 ↪format(time()))
# tensorboard = TensorBoard(log_dir='logs')
model.compile(loss='categorical_crossentropy', optimizer=keras.optimizers.
 ↪Adam(lr=0.0006,decay = 1e-4),metrics=['accuracy', auc])
print(model.summary())
```

```
WARNING:tensorflow:From /usr/local/lib/python3.6/dist-
packages/keras/backend/tensorflow_backend.py:148: The name
tf.placeholder_with_default is deprecated. Please use
tf.compat.v1.placeholder_with_default instead.

WARNING:tensorflow:From /usr/local/lib/python3.6/dist-
packages/keras/backend/tensorflow_backend.py:3733: calling dropout (from
tensorflow.python.ops.nn_ops) with keep_prob is deprecated and will be removed
in a future version.
Instructions for updating:
Please use `rate` instead of `keep_prob`. Rate should be set to `rate = 1 -
keep_prob`.
WARNING:tensorflow:From /usr/local/lib/python3.6/dist-
```

```
packages/keras/backend/tensorflow_backend.py:4479: The name tf.truncated_normal
is deprecated. Please use tf.random.truncated_normal instead.

WARNING:tensorflow:From /usr/local/lib/python3.6/dist-
packages/keras/optimizers.py:793: The name tf.train.Optimizer is deprecated.
Please use tf.compat.v1.train.Optimizer instead.

WARNING:tensorflow:From /usr/local/lib/python3.6/dist-
packages/keras/backend/tensorflow_backend.py:3576: The name tf.log is
deprecated. Please use tf.math.log instead.

WARNING:tensorflow:From <ipython-input-49-f61a545c8113>:6: py_func (from
tensorflow.python.ops.script_ops) is deprecated and will be removed in a future
version.
Instructions for updating:
tf.py_func is deprecated in TF V2. Instead, there are two
    options available in V2.
    - tf.py_function takes a python function which manipulates tf eager
    tensors instead of numpy arrays. It's easy to convert a tf eager tensor to
    an ndarray (just call tensor.numpy()) but having access to eager tensors
    means `tf.py_function`s can use accelerators such as GPUs as well as
    being differentiable using a gradient tape.
    - tf.numpy_function maintains the semantics of the deprecated tf.py_func
    (it is not differentiable, and manipulates numpy arrays). It drops the
    stateful argument making all functions stateful.

Model: "model_1"
--------------------------------------------------------------------------------
------------------
Layer (type)                    Output Shape         Param #     Connected to
================================================================================
==================
input_2 (InputLayer)            (None, 250)          0

--------------------------------------------------------------------------------
------------------
embedding_2 (Embedding)         (None, 250, 300)     14180400    input_2[0][0]

--------------------------------------------------------------------------------
------------------
spatial_dropout1d_1 (SpatialDro (None, 250, 300)     0
embedding_2[0][0]

--------------------------------------------------------------------------------
------------------
input_3 (InputLayer)            (None, 1)            0

--------------------------------------------------------------------------------
------------------
input_4 (InputLayer)            (None, 1)            0

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

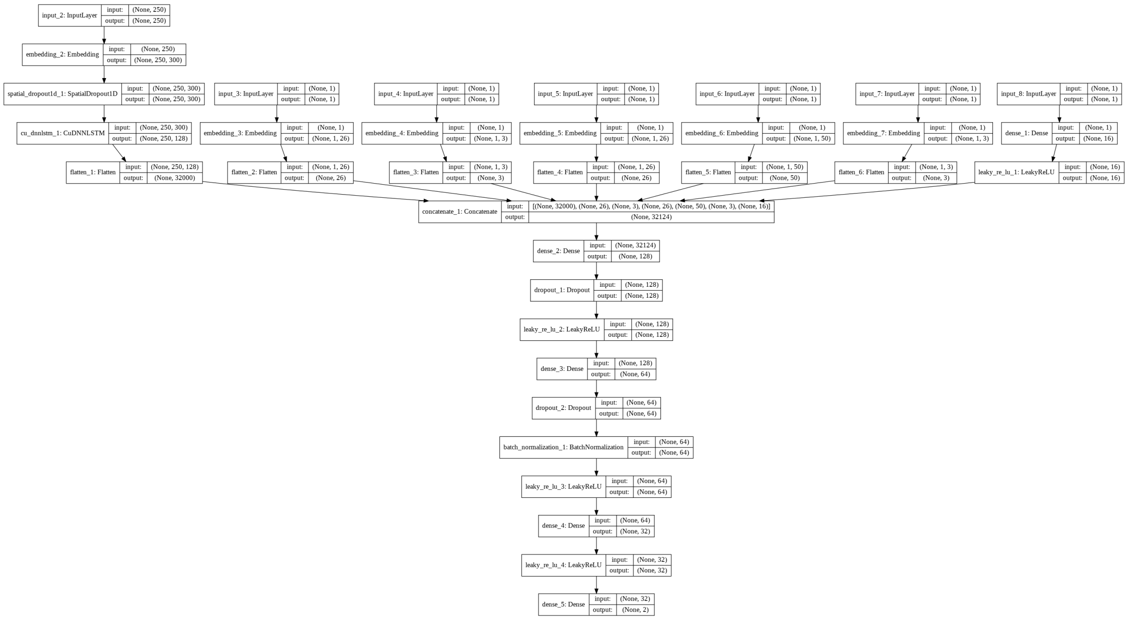
```
input_5 (InputLayer)            (None, 1)            0

--------------------------------------------------------------------------------
------------------
input_6 (InputLayer)            (None, 1)            0

--------------------------------------------------------------------------------
------------------
input_7 (InputLayer)            (None, 1)            0

--------------------------------------------------------------------------------
------------------
input_8 (InputLayer)            (None, 1)            0

--------------------------------------------------------------------------------
------------------
cu_dnnlstm_1 (CuDNNLSTM)        (None, 250, 128)     220160
spatial_dropout1d_1[0][0]

--------------------------------------------------------------------------------
------------------
embedding_3 (Embedding)         (None, 1, 26)        1352        input_3[0][0]

--------------------------------------------------------------------------------
------------------
embedding_4 (Embedding)         (None, 1, 3)         15          input_4[0][0]

--------------------------------------------------------------------------------
------------------
embedding_5 (Embedding)         (None, 1, 26)        1326        input_5[0][0]

--------------------------------------------------------------------------------
------------------
embedding_6 (Embedding)         (None, 1, 50)        19700       input_6[0][0]

--------------------------------------------------------------------------------
------------------
embedding_7 (Embedding)         (None, 1, 3)         18          input_7[0][0]

--------------------------------------------------------------------------------
------------------
dense_1 (Dense)                 (None, 16)           32          input_8[0][0]

--------------------------------------------------------------------------------
------------------
flatten_1 (Flatten)             (None, 32000)        0
cu_dnnlstm_1[0][0]

--------------------------------------------------------------------------------
------------------
flatten_2 (Flatten)             (None, 26)           0
embedding_3[0][0]

--------------------------------------------------------------------------------
------------------
flatten_3 (Flatten)             (None, 3)            0
embedding_4[0][0]

--------------------------------------------------------------------------------
------------------
flatten_4 (Flatten)             (None, 26)           0
embedding_5[0][0]
```

```
--------------------------------------------------------------------------------
------------------
flatten_5 (Flatten)             (None, 50)           0
embedding_6[0][0]
--------------------------------------------------------------------------------
------------------
flatten_6 (Flatten)             (None, 3)            0
embedding_7[0][0]
--------------------------------------------------------------------------------
------------------
leaky_re_lu_1 (LeakyReLU)       (None, 16)           0           dense_1[0][0]
--------------------------------------------------------------------------------
------------------
concatenate_1 (Concatenate)     (None, 32124)        0           flatten_1[0][0]
                                                                 flatten_2[0][0]
                                                                 flatten_3[0][0]
                                                                 flatten_4[0][0]
                                                                 flatten_5[0][0]
                                                                 flatten_6[0][0]
leaky_re_lu_1[0][0]
--------------------------------------------------------------------------------
------------------
dense_2 (Dense)                 (None, 128)          4112000
concatenate_1[0][0]
--------------------------------------------------------------------------------
------------------
dropout_1 (Dropout)             (None, 128)          0           dense_2[0][0]
--------------------------------------------------------------------------------
------------------
leaky_re_lu_2 (LeakyReLU)       (None, 128)          0           dropout_1[0][0]
--------------------------------------------------------------------------------
------------------
dense_3 (Dense)                 (None, 64)           8256
leaky_re_lu_2[0][0]
--------------------------------------------------------------------------------
------------------
dropout_2 (Dropout)             (None, 64)           0           dense_3[0][0]
--------------------------------------------------------------------------------
------------------
batch_normalization_1 (BatchNor (None, 64)           256         dropout_2[0][0]
--------------------------------------------------------------------------------
------------------
leaky_re_lu_3 (LeakyReLU)       (None, 64)           0
batch_normalization_1[0][0]
--------------------------------------------------------------------------------
------------------
dense_4 (Dense)                 (None, 32)           2080
leaky_re_lu_3[0][0]
```

```
--------------------------------------------------------------------------------
------------------
leaky_re_lu_4 (LeakyReLU)       (None, 32)           0           dense_4[0][0]

--------------------------------------------------------------------------------
------------------
dense_5 (Dense)                 (None, 2)            66
leaky_re_lu_4[0][0]
================================================================================
==================
Total params: 18,545,661
Trainable params: 4,365,133
Non-trainable params: 14,180,528

--------------------------------------------------------------------------------
------------------
None
```

[55]:
```
#https://machinelearningmastery.com/
↪visualize-deep-learning-neural-network-model-keras/
from keras.utils.vis_utils import plot_model
plot_model(model, to_file='/content/drive/My Drive/LSTM Output/model_1.png',␣
↪show_shapes=True, show_layer_names=True)
```

[55]:



input_2: InputLayer | input: (None, 250) | output: (None, 250)
embedding_2: Embedding | input: (None, 250) | output: (None, 250, 300)
spatial_dropout1d_1: SpatialDropout1D | input: (None, 250, 300) | output: (None, 250, 300)
cu_dnnlstm_1: CuDNNLSTM | input: (None, 250, 300) | output: (None, 250, 128)
flatten_1: Flatten | input: (None, 250, 128) | output: (None, 32000)
input_3: InputLayer | input: (None, 1) | output: (None, 1)
embedding_3: Embedding | input: (None, 1) | output: (None, 1, 26)
flatten_2: Flatten | input: (None, 1, 26) | output: (None, 26)
input_4: InputLayer | input: (None, 1) | output: (None, 1)
embedding_4: Embedding | input: (None, 1) | output: (None, 1, 3)
flatten_3: Flatten | input: (None, 1, 3) | output: (None, 3)
input_5: InputLayer | input: (None, 1) | output: (None, 1)
embedding_5: Embedding | input: (None, 1) | output: (None, 1, 26)
flatten_4: Flatten | input: (None, 1, 26) | output: (None, 26)
input_6: InputLayer | input: (None, 1) | output: (None, 1)
embedding_6: Embedding | input: (None, 1) | output: (None, 1, 50)
flatten_5: Flatten | input: (None, 1, 50) | output: (None, 50)
input_7: InputLayer | input: (None, 1) | output: (None, 1)
embedding_7: Embedding | input: (None, 1) | output: (None, 1, 3)
flatten_6: Flatten | input: (None, 1, 3) | output: (None, 3)
input_8: InputLayer | input: (None, 1) | output: (None, 1)
dense_1: Dense | input: (None, 1) | output: (None, 16)
leaky_re_lu_1: LeakyReLU | input: (None, 16) | output: (None, 16)
concatenate_1: Concatenate | input: [(None, 32000), (None, 26), (None, 3), (None, 26), (None, 50), (None, 3), (None, 16)] | output: (None, 32124)
dense_2: Dense | input: (None, 32124) | output: (None, 128)
dropout_1: Dropout | input: (None, 128) | output: (None, 128)
leaky_re_lu_2: LeakyReLU | input: (None, 128) | output: (None, 128)
dense_3: Dense | input: (None, 128) | output: (None, 64)
dropout_2: Dropout | input: (None, 64) | output: (None, 64)
batch_normalization_1: BatchNormalization | input: (None, 64) | output: (None, 64)
leaky_re_lu_3: LeakyReLU | input: (None, 64) | output: (None, 64)
dense_4: Dense | input: (None, 64) | output: (None, 32)
leaky_re_lu_4: LeakyReLU | input: (None, 32) | output: (None, 32)
dense_5: Dense | input: (None, 32) | output: (None, 2)

[56]:
```
# print shape of train data
print(X_train_essay.shape)
print(X_train_school_state_ohe.shape)
print(X_train_project_grade_category_ohe.shape)
print(X_train_clean_categories_ohe.shape)
```

```python
print(X_train_clean_subcategories_ohe.shape)
print(X_train_teacher_prefix_ohe.shape)
print(X_train['remaining_input'].shape)
print('*'*100)
# print shape of cv data
print(X_cv_essay.shape)
print(X_cv_school_state_ohe.shape)
print(X_cv_project_grade_category_ohe.shape)
print(X_cv_clean_categories_ohe.shape)
print(X_cv_clean_subcategories_ohe.shape)
print(X_cv_teacher_prefix_ohe.shape)
print(X_cv['remaining_input'].shape)
print('*'*100)
# print shape of test data
print(X_test_essay.shape)
print(X_test_school_state_ohe.shape)
print(X_test_project_grade_category_ohe.shape)
print(X_test_clean_categories_ohe.shape)
print(X_test_clean_subcategories_ohe.shape)
print(X_test_teacher_prefix_ohe.shape)
print(X_test['remaining_input'].shape)
print('*'*100)
```

```
(69918, 250)
(69918, 1)
(69918, 1)
(69918, 1)
(69918, 1)
(69918, 1)
(69918,)
********************************************************************************
********************
(17480, 250)
(17480, 1)
(17480, 1)
(17480, 1)
(17480, 1)
(17480, 1)
(17480,)
********************************************************************************
********************
(21850, 250)
(21850, 1)
(21850, 1)
(21850, 1)
(21850, 1)
(21850, 1)
```

```
(21850,)
********************************************************************************
********************
```

[57]:

```
# with open('/content/drive/My Drive/LSTM Output/abc.gdoc', 'w') as f:
#   f.write('content')

#model fitting
#https://machinelearningmastery.com/check-point-deep-learning-models-keras/
filepath="/content/drive/My Drive/LSTM Output/weights_copy_new_23_1.best.hdf5"

earlystopping_1 = EarlyStopping(monitor='val_loss', patience=2, verbose=1)

checkpoint = ModelCheckpoint(filepath, monitor='val_auc', verbose=1,␣
 ↪save_best_only=True, mode='max')

callbacks_list = [checkpoint,tensorboard,earlystopping_1]

model.fit([X_train_essay, X_train_school_state_ohe,␣
 ↪X_train_project_grade_category_ohe, X_train_clean_categories_ohe,␣
 ↪X_train_clean_subcategories_ohe, X_train_teacher_prefix_ohe,␣
 ↪X_train['remaining_input']], y_train, nb_epoch=50, verbose=1,␣
 ↪batch_size=256, validation_data=([X_cv_essay, X_cv_school_state_ohe,␣
 ↪X_cv_project_grade_category_ohe, X_cv_clean_categories_ohe,␣
 ↪X_cv_clean_subcategories_ohe, X_cv_teacher_prefix_ohe,␣
 ↪X_cv['remaining_input']]  , y_cv), callbacks = callbacks_list)
```

```
WARNING:tensorflow:From /usr/local/lib/python3.6/dist-
packages/tensorflow_core/python/ops/math_grad.py:1424: where (from
tensorflow.python.ops.array_ops) is deprecated and will be removed in a future
version.
Instructions for updating:
Use tf.where in 2.0, which has the same broadcast rule as np.where
WARNING:tensorflow:From /usr/local/lib/python3.6/dist-
packages/keras/backend/tensorflow_backend.py:1033: The name tf.assign_add is
deprecated. Please use tf.compat.v1.assign_add instead.

WARNING:tensorflow:From /usr/local/lib/python3.6/dist-
packages/keras/backend/tensorflow_backend.py:1020: The name tf.assign is
deprecated. Please use tf.compat.v1.assign instead.

Train on 69918 samples, validate on 17480 samples
Epoch 1/50
69918/69918 [==============================] - 36s 517us/step - loss: 0.4732 -
acc: 0.8373 - auc: 0.6026 - val_loss: 0.4409 - val_acc: 0.8490 - val_auc: 0.6797

Epoch 00001: val_auc improved from -inf to 0.67965, saving model to
```

```
/content/drive/My Drive/LSTM Output/weights_copy_new_23_1.best.hdf5
Epoch 2/50
69918/69918 [==============================] - 34s 484us/step - loss: 0.4279 -
acc: 0.8478 - auc: 0.6839 - val_loss: 0.4441 - val_acc: 0.8467 - val_auc: 0.7208

Epoch 00002: val_auc improved from 0.67965 to 0.72079, saving model to
/content/drive/My Drive/LSTM Output/weights_copy_new_23_1.best.hdf5
Epoch 3/50
69918/69918 [==============================] - 34s 482us/step - loss: 0.4144 -
acc: 0.8503 - auc: 0.7073 - val_loss: 0.4369 - val_acc: 0.8519 - val_auc: 0.7056

Epoch 00003: val_auc did not improve from 0.72079
Epoch 4/50
69918/69918 [==============================] - 34s 483us/step - loss: 0.4073 -
acc: 0.8496 - auc: 0.7185 - val_loss: 0.4156 - val_acc: 0.8497 - val_auc: 0.7247

Epoch 00004: val_auc improved from 0.72079 to 0.72473, saving model to
/content/drive/My Drive/LSTM Output/weights_copy_new_23_1.best.hdf5
Epoch 5/50
69918/69918 [==============================] - 34s 483us/step - loss: 0.4028 -
acc: 0.8500 - auc: 0.7278 - val_loss: 0.4166 - val_acc: 0.8533 - val_auc: 0.7350

Epoch 00005: val_auc improved from 0.72473 to 0.73497, saving model to
/content/drive/My Drive/LSTM Output/weights_copy_new_23_1.best.hdf5
Epoch 6/50
69918/69918 [==============================] - 34s 484us/step - loss: 0.3957 -
acc: 0.8519 - auc: 0.7385 - val_loss: 0.4016 - val_acc: 0.8542 - val_auc: 0.7334

Epoch 00006: val_auc did not improve from 0.73497
Epoch 7/50
69918/69918 [==============================] - 34s 483us/step - loss: 0.3932 -
acc: 0.8516 - auc: 0.7414 - val_loss: 0.3928 - val_acc: 0.8535 - val_auc: 0.7403

Epoch 00007: val_auc improved from 0.73497 to 0.74029, saving model to
/content/drive/My Drive/LSTM Output/weights_copy_new_23_1.best.hdf5
Epoch 8/50
69918/69918 [==============================] - 34s 484us/step - loss: 0.3897 -
acc: 0.8524 - auc: 0.7462 - val_loss: 0.3883 - val_acc: 0.8527 - val_auc: 0.7514

Epoch 00008: val_auc improved from 0.74029 to 0.75138, saving model to
/content/drive/My Drive/LSTM Output/weights_copy_new_23_1.best.hdf5
Epoch 9/50
69918/69918 [==============================] - 34s 483us/step - loss: 0.3848 -
acc: 0.8536 - auc: 0.7519 - val_loss: 0.3880 - val_acc: 0.8547 - val_auc: 0.7454

Epoch 00009: val_auc did not improve from 0.75138
Epoch 10/50
69918/69918 [==============================] - 34s 482us/step - loss: 0.3818 -
```

```
acc: 0.8545 - auc: 0.7567 - val_loss: 0.3913 - val_acc: 0.8535 - val_auc: 0.7494

Epoch 00010: val_auc did not improve from 0.75138
Epoch 11/50
69918/69918 [==============================] - 34s 484us/step - loss: 0.3802 -
acc: 0.8544 - auc: 0.7608 - val_loss: 0.3905 - val_acc: 0.8547 - val_auc: 0.7552

Epoch 00011: val_auc improved from 0.75138 to 0.75523, saving model to
/content/drive/My Drive/LSTM Output/weights_copy_new_23_1.best.hdf5
Epoch 00011: early stopping
```

[57]:

```
<keras.callbacks.History at 0x7f930d9a03c8>
```

[0]:

```
# Load model with best weights

#input 1 essay text
input1 = Input(shape=(250,))
# x1 = Embedding(input_dim=49042,output_dim= 300)(input1)
x1 = Embedding(input_dim=47268,output_dim=␣
 ↪300,weights=[X_train_essay_mat],trainable=False)(input1)
x1 = SpatialDropout1D(0.3)(x1)
x1 = CuDNNLSTM(128,return_sequences=True)(x1)
# x1 = LSTM(128,return_sequences=True)(x1)
x1 = Flatten()(x1)

#https://medium.com/@davidheffernan_99410/
 ↪an-introduction-to-using-categorical-embeddings-ee686ed7e7f9
cat_vars =␣
 ↪["teacher_prefix","school_state","project_grade_category","clean_categories","clean_subcate
cat_sizes = {}
cat_embsizes = {}
for cat in cat_vars:
    cat_sizes[cat] = X_train[cat].nunique()
    cat_embsizes[cat] = min(50, cat_sizes[cat]//2+1)

# input 2 school_state 51
input2 = Input(shape=(1,))
x2 = Embedding(input_dim=cat_sizes['school_state']+1,␣
 ↪output_dim=cat_embsizes['school_state'])(input2)#input_dim=52
x2 = Flatten()(x2)

# input 3 project_grade_cat 4
input3 = Input(shape=(1,))
x3 = Embedding(input_dim=cat_sizes['project_grade_category']+1,␣
 ↪output_dim=cat_embsizes['project_grade_category'])(input3)
x3 = Flatten()(x3)

# input 4 clean_categories 9
```

```
input4 = Input(shape=(1,))
x4 = Embedding(input_dim=cat_sizes['clean_categories']+1,␣
 ↪output_dim=cat_embsizes['clean_categories'])(input4)
x4 = Flatten()(x4)

# input 5 clean_subcategories 30
input5 = Input(shape=(1,))
x5 = Embedding(input_dim=cat_sizes['clean_subcategories']+1,␣
 ↪output_dim=cat_embsizes['clean_subcategories'])(input5)
x5 = Flatten()(x5)

# input 6 teacher_prefix 5
input6 = Input(shape=(1,))
x6 = Embedding(input_dim=cat_sizes['teacher_prefix']+1,␣
 ↪output_dim=cat_embsizes['teacher_prefix'])(input6)
x6 = Flatten()(x6)

# print(cat_concat)

#input 7 remaining inout
input7 = Input(shape=(1,))
x7 = Dense(16,kernel_initializer=he_normal(),kernel_regularizer=l2(0.
 ↪0001))(input7)
x7 = LeakyReLU()(x7)

concat = concatenate([x1, x2, x3, x4, x5, x6, x7])

x = Dense(128,kernel_initializer=he_normal(),kernel_regularizer=l2(0.
 ↪0001))(concat)
x = Dropout(0.5)(x)
x = LeakyReLU()(x)
x = Dense(64,kernel_initializer=he_normal(),kernel_regularizer=l2(0.0001))(x)
x = Dropout(0.5)(x)
x = BatchNormalization()(x)
x = LeakyReLU()(x)
x = Dense(32,kernel_initializer=he_normal(),kernel_regularizer=l2(0.0001))(x)
x = LeakyReLU()(x)
# x = Dropout(0.5)(x)
output = Dense(2, activation = 'softmax')(x)

# create model with seven inputs
model = Model([input1,input2,input3,input4,input5,input6,input7], output)#
model.run_eagerly = True
tensorboard = TensorBoard(log_dir='/content/drive/My Drive/LSTM Output/logs/{}'.
 ↪format(time()))
# tensorboard = TensorBoard(log_dir='logs')
```

```
model.load_weights(filepath)#"weights_copy.best.hdf5"
model.compile(loss='categorical_crossentropy', optimizer=keras.optimizers.
 ↪Adam(lr=0.0006,decay = 1e-4),metrics=['accuracy', auc])
# print(model.summary())
# print(model.summary())
```

[59]:

```
print("AUC for test data: %0.3f"%roc_auc_score(y_test,model.
 ↪predict([X_test_essay, X_test_school_state_ohe,␣
 ↪X_test_project_grade_category_ohe, X_test_clean_categories_ohe,␣
 ↪X_test_clean_subcategories_ohe, X_test_teacher_prefix_ohe,␣
 ↪X_test['remaining_input']])))
print("AUC for CV data: %0.3f"%roc_auc_score(y_cv,model.predict([X_cv_essay,␣
 ↪X_cv_school_state_ohe, X_cv_project_grade_category_ohe,␣
 ↪X_cv_clean_categories_ohe, X_cv_clean_subcategories_ohe,␣
 ↪X_cv_teacher_prefix_ohe, X_cv['remaining_input']])))
print("AUC for train data: %0.3f"%roc_auc_score(y_train,model.
 ↪predict([X_train_essay, X_train_school_state_ohe,␣
 ↪X_train_project_grade_category_ohe, X_train_clean_categories_ohe,␣
 ↪X_train_clean_subcategories_ohe, X_train_teacher_prefix_ohe,␣
 ↪X_train['remaining_input']])))
```

```
AUC for test data: 0.761
AUC for CV data: 0.755
AUC for train data: 0.780
```

Attach Tensorboard Image from logs directory
Ref: https://i.imgur.com/W8BwiNL.png
Assignment 2
Use the same model as above but for 'input_seq_total_text_data' give only some words in the sentance not all the words. Filter the words as below.

[0]:

```
vectorizer = TfidfVectorizer(min_df=10,max_features=10000) #Defining TFIDF with␣
 ↪min_df=10
imp_tf = vectorizer.fit(X_train['essay'])
```

[0]:

```
idf_values = vectorizer.idf_
```

[62]:

```
df = pd.DataFrame(idf_values, columns= ["idf"])
df.head()
```

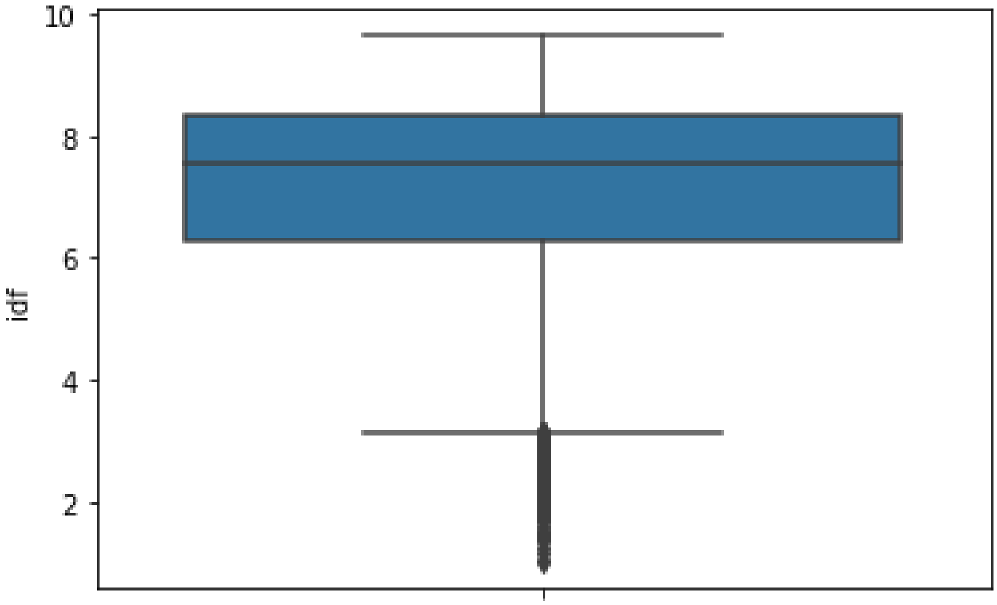
[62]:

```
        idf
0  7.164660
1  5.916768
2  4.481870
3  3.815592
4  7.073688
```

[63]:

```
import seaborn as sns

sns.boxplot(y = "idf", data = df )
```

[63]: <matplotlib.axes._subplots.AxesSubplot at 0x7f92a9227ba8>



[64]:
```
#calculating speed values at each percntile 0,10,20,30,40,50,60,70,80,90,100
for i in range(0,100,10):
    var =idf_values
    var = np.sort(var,axis = None)
    print("{} percentile value is {}".format(i,var[int(len(var)*(float(i)/
 ↪100))]))
print("100 percentile value is ",var[-1])
```

```
0 percentile value is 1.0076667233625343
10 percentile value is 4.9576573540710545
20 percentile value is 5.878449218826001
30 percentile value is 6.5905723008449515
40 percentile value is 7.118140105754016
50 percentile value is 7.549922522179553
60 percentile value is 7.906597466118286
70 percentile value is 8.203848989586218
80 percentile value is 8.491531062038
90 percentile value is 8.78779687818117
100 percentile value is  9.670186058379645
```

[65]:
```
for i in range(0,100,5):
    var =idf_values
    var = np.sort(var,axis = None)
    print("{} percentile value is {}".format(i,var[int(len(var)*(float(i)/
 ↪100))]))
```

```
print("100 percentile value is ",var[-1])
```

```
0 percentile value is 1.0076667233625343
5 percentile value is 4.110466428400308
10 percentile value is 4.9576573540710545
15 percentile value is 5.4692317610992855
20 percentile value is 5.878449218826001
25 percentile value is 6.268988676717489
30 percentile value is 6.5905723008449515
35 percentile value is 6.861787883443153
40 percentile value is 7.118140105754016
45 percentile value is 7.342908352795227
50 percentile value is 7.549922522179553
55 percentile value is 7.736252100371047
60 percentile value is 7.906597466118286
65 percentile value is 8.044218843994333
70 percentile value is 8.203848989586218
75 percentile value is 8.348430218397326
80 percentile value is 8.491531062038
85 percentile value is 8.628732183551485
90 percentile value is 8.78779687818117
95 percentile value is 8.896996170146164
100 percentile value is  9.670186058379645
```

[66]:

```
print("The 25 percentile of idf score is :", np.percentile(idf_values,[25]))
print("The 75 percentile of idf score is :",np.percentile(idf_values,[75]))
```

```
The 25 percentile of idf score is : [6.26898868]
The 75 percentile of idf score is : [8.34843022]
```

- We will pick essay text features that has idf_ values b/w 6.27 to 8.33
- As per reviewer I considered idf values from 2 to 10

[0]:

```
percentile_25th = 2#np.percentile(idf_values,[25])
percentile_75th = 10#np.percentile(idf_values,[75])
```

[68]:

```
feature_idf = zip(imp_tf.get_feature_names(),idf_values)

feature_name = []
for x,y in feature_idf:

    if y >= percentile_25th and y <= percentile_75th:
        feature_name.append(x)

print(len(feature_name))
```

```
9975
```

[69]:
```
print(feature_name)
```

```
['00', '000', '10', '100', '1000', '100th', '101', '10th', '11', '110', '1100',
'115', '11th', '12', '120', '1200', '125', '12th', '13', '130', '14', '140',
'1400', '15', '150', '1500', '16', '160', '1600', '17', '170', '18', '180',
'19', '1st', '20', '200', '2000', '2007', '2008', '2009', '2010', '2011',
'2012', '2013', '2014', '2015', '2016', '2017', '2018', '20th', '21', '21st',
'22', '23', '24', '25', '250', '26', '27', '28', '280', '29', '2d', '2nd', '30',
'300', '3000', '31', '32', '320', '33', '34', '35', '350', '36', '360', '365',
'37', '38', '39', '3d', '3doodler', '3doodlers', '3rd', '40', '400', '41', '42',
'43', '44', '45', '450', '46', '47', '48', '49', '4k', '4th', '50', '500',
'504', '51', '52', '53', '54', '55', '550', '56', '560', '57', '58', '5k',
'5th', '60', '600', '61', '62', '63', '64', '65', '650', '66', '67', '68', '69',
'693', '6th', '70', '700', '71', '72', '73', '74', '75', '750', '76', '77',
'78', '79', '7th', '80', '800', '81', '82', '83', '84', '85', '850', '86', '87',
'88', '89', '8th', '90', '900', '91', '92', '93', '94', '95', '96', '97', '98',
'99', '9th', 'aac', 'abandoned', 'abc', 'abcs', 'abcya', 'abdominal',
'abilities', 'ability', 'abound', 'about', 'above', 'abraham', 'abroad',
'absence', 'absences', 'absent', 'absolute', 'absolutely', 'absorb', 'absorbed',
'absorbing', 'abstract', 'abundance', 'abundant', 'abuse', 'abused', 'academia',
'academic', 'academically', 'academics', 'academies', 'academy', 'accelerate',
'accelerated', 'acceleration', 'accept', 'acceptable', 'acceptance', 'accepted',
'accepting', 'accepts', 'access', 'accessed', 'accessibility', 'accessible',
'accessing', 'accessories', 'accessory', 'accident', 'accidentally',
'accidents', 'acclimate', 'accommodate', 'accommodated', 'accommodates',
'accommodating', 'accommodation', 'accommodations', 'accompanied',
'accompaniment', 'accompany', 'accompanying', 'accomplish', 'accomplished',
'accomplishing', 'accomplishment', 'accomplishments', 'according',
'accordingly', 'account', 'accountability', 'accountable', 'accounting',
'accounts', 'accredited', 'accumulate', 'accumulated', 'accuracy', 'accurate',
'accurately', 'accustomed', 'ace', 'acer', 'achievable', 'achieve', 'achieved',
'achievement', 'achievements', 'achiever', 'achievers', 'achieving', 'acid',
'acknowledge', 'acknowledged', 'acoustic', 'acquainted', 'acquire', 'acquired',
'acquiring', 'acquisition', 'across', 'acrylic', 'act', 'acting', 'action',
'actions', 'activate', 'activating', 'active', 'actively', 'activism',
'activists', 'activites', 'activities', 'activity', 'actors', 'acts', 'actual',
'actually', 'ad', 'adapt', 'adaptable', 'adaptations', 'adapted', 'adapter',
'adapting', 'adaptive', 'add', 'added', 'addicted', 'addiction', 'adding',
'addition', 'additional', 'additionally', 'additions', 'address', 'addressed',
'addresses', 'addressing', 'adds', 'adept', 'adequate', 'adequately', 'adhd',
'adhere', 'adhesive', 'adjacent', 'adjectives', 'adjust', 'adjustable',
'adjusted', 'adjusting', 'adjustment', 'adjustments', 'administered',
'administration', 'administrative', 'administrator', 'administrators',
'admirable', 'admire', 'admission', 'admit', 'admitted', 'adobe', 'adolescence',
'adolescent', 'adolescents', 'adopt', 'adopted', 'adoption', 'adorable',
'adore', 'adult', 'adulthood', 'adults', 'advance', 'advanced', 'advancement',
'advancements', 'advances', 'advancing', 'advantage', 'advantaged',
'advantages', 'adventure', 'adventurers', 'adventures', 'adventurous',
'adverse', 'adversities', 'adversity', 'advertise', 'advertisements',
'advertising', 'advice', 'advisory', 'advocacy', 'advocate', 'advocates',
'advocating', 'aed', 'aerial', 'aerobic', 'aerospace', 'aesthetic',
'aesthetically', 'affect', 'affected', 'affecting', 'affection', 'affective',
'affects', 'affiliated', 'affirmation', 'affluent', 'afford', 'affordable',
'afforded', 'affording', 'affords', 'afghanistan', 'aforementioned', 'afraid',
'africa', 'african', 'after', 'afternoon', 'afternoons', 'afterschool',
'afterwards', 'again', 'age', 'aged', 'agency', 'agenda', 'agendas', 'agents',
'ages', 'aggression', 'aggressive', 'agility', 'aging', 'ago', 'agree',
'agreed', 'agricultural', 'agriculture', 'ah', 'aha', 'ahead', 'aid', 'aide',
'aided', 'aides', 'aiding', 'aids', 'aig', 'aim', 'aimed', 'aiming', 'aims',
'air', 'airplanes', 'aka', 'al', 'alabama', 'alarm', 'alaska', 'alaskan',
'albert', 'alcohol', 'alert', 'alertness', 'alexa', 'alexander', 'algebra',
'algebraic', 'algorithms', 'alice', 'align', 'aligned', 'aligns', 'alike',
'alive', 'all', 'allergies', 'alleviate', 'allocated', 'allotted', 'allow',
'allowed', 'allowing', 'allows', 'almost', 'alone', 'along', 'alongs',
'alongside', 'alot', 'aloud', 'alouds', 'alphabet', 'already', 'alter',
'altering', 'alternate', 'alternative', 'alternatives', 'although', 'alto',
'altogether', 'alumni', 'always', 'am', 'amaze', 'amazed', 'amazement',
'amazes', 'amazing', 'amazingly', 'amazon', 'ambassadors', 'ambition',
'ambitions', 'ambitious', 'amenities', 'america', 'american', 'americans',
'amharic', 'among', 'amongst', 'amount', 'amounts', 'ample', 'amplification',
'amplified', 'amplifier', 'amplify', 'amusement', 'an', 'analog', 'analysis',
'analytical', 'analyze', 'analyzed', 'analyzing', 'anatomy', 'ancestors',
'anchor', 'anchorage', 'anchors', 'ancient', 'and', 'anderson', 'android',
'angeles', 'angelou', 'angels', 'anger', 'angle', 'angles', 'angry', 'animal',
'animals', 'animate', 'animated', 'animation', 'animations', 'anne', 'annotate',
'annotating', 'annotation', 'annotations', 'announce', 'announcement',
'announcements', 'annual', 'annually', 'anonymous', 'another', 'answer',
'answered', 'answering', 'answers', 'ant', 'anti', 'anticipate', 'anticipated',
'anticipating', 'anticipation', 'antiquated', 'antonio', 'antonyms', 'ants',
'antsy', 'anxieties', 'anxiety', 'anxious', 'anxiously', 'any', 'anybody',
'anymore', 'anyone', 'anything', 'anytime', 'anyway', 'anywhere', 'ap', 'apart',
'apartment', 'apartments', 'app', 'appalachia', 'appalachian', 'apparatus',
'apparent', 'appeal', 'appealing', 'appeals', 'appear', 'appearance', 'appears',
'appetite', 'apple', 'apples', 'applesauce', 'appliances', 'applicable',
'application', 'applications', 'applied', 'applies', 'apply', 'applying',
'appreciate', 'appreciated', 'appreciating', 'appreciation', 'appreciative',
'apprehensive', 'approach', 'approachable', 'approached', 'approaches',
'approaching', 'appropriate', 'appropriately', 'approval', 'approved', 'approx',
'approximately', 'apps', 'april', 'aprons', 'apt', 'aptitude', 'aquaponics',
'aquarium', 'aquariums', 'aquatic', 'ar', 'arabic', 'archery', 'architect',
'architects', 'architectural', 'architecture', 'arctic', 'arduino', 'are',
'area', 'areas', 'arena', 'argue', 'arguing', 'argument', 'argumentative',
'arguments', 'arise', 'arises', 'arithmetic', 'arizona', 'arkansas', 'arm',
'armed', 'arms', 'army', 'around', 'arrange', 'arranged', 'arrangement',
'arrangements', 'arranging', 'array', 'arrays', 'arrival', 'arrivals', 'arrive',
'arrived', 'arrives', 'arriving', 'arrows', 'art', 'article', 'articles',
'articulate', 'articulation', 'artifacts', 'artificial', 'artist', 'artistic',
'artistically', 'artists', 'arts', 'artwork', 'artworks', 'as', 'asd',
'ashamed', 'asia', 'asian', 'aside', 'ask', 'asked', 'asking', 'asks', 'asl',
'asleep', 'aspect', 'aspects', 'asperger', 'aspirations', 'aspire', 'aspiring',
'assemble', 'assembled', 'assemblies', 'assembly', 'assess', 'assessed',
'assessing', 'assessment', 'assessments', 'asset', 'assets', 'assign',
'assigned', 'assigning', 'assignment', 'assignments', 'assimilate', 'assist',
'assistance', 'assistant', 'assistants', 'assisted', 'assisting', 'assistive',
'assists', 'associate', 'associated', 'association', 'assorted', 'assortment',
'assume', 'assure', 'assured', 'asthma', 'astound', 'astounding', 'astronauts',
'astronomy', 'asus', 'at', 'ate', 'athlete', 'athletes', 'athletic',
'athletics', 'atlanta', 'atlases', 'atmosphere', 'atomic', 'atoms', 'atpe',
'attach', 'attached', 'attack', 'attain', 'attainable', 'attained', 'attaining',
'attempt', 'attempted', 'attempting', 'attempts', 'attend', 'attendance',
'attended', 'attending', 'attends', 'attention', 'attentive', 'attentiveness',
'attire', 'attitude', 'attitudes', 'attract', 'attracted', 'attractive',
'attracts', 'attribute', 'attributes', 'auction', 'audible', 'audience',
'audiences', 'audio', 'audiobook', 'audiobooks', 'audition', 'auditorium',
'auditory', 'auggie', 'augment', 'augmentative', 'augmented', 'august', 'aunt',
'aunts', 'austin', 'authentic', 'authentically', 'author', 'authority',
'authors', 'autism', 'autistic', 'automatic', 'automatically', 'automaticity',
'autonomous', 'autonomy', 'availability', 'available', 'avenue', 'avenues',
'average', 'averages', 'avid', 'avoid', 'avoiding', 'await', 'awaiting',
'awaits', 'awake', 'awaken', 'award', 'awarded', 'awards', 'aware', 'awareness',
'away', 'awe', 'awesome', 'awful', 'awhile', 'awkward', 'az', 'babies', 'baby',
'babysitting', 'baccalaureate', 'back', 'backbone', 'backdrop', 'backdrops',
'backed', 'background', 'backgrounds', 'backpack', 'backpacks', 'backpatter',
'backs', 'backwards', 'backyard', 'bacteria', 'bad', 'badges', 'badly',
'badminton', 'bag', 'baggage', 'baggies', 'bags', 'bake', 'baking', 'balance',
'balanced', 'balances', 'balancing', 'ball', 'ballet', 'balloon', 'balloons',
'balls', 'baltimore', 'band', 'bands', 'bang', 'bangladesh', 'bank', 'banks',
'bar', 'bare', 'barely', 'baritone', 'barn', 'barred', 'barrier', 'barriers',
'bars', 'basal', 'base', 'baseball', 'baseballs', 'based', 'baseline', 'bases',
'basic', 'basically', 'basics', 'basis', 'basket', 'basketball', 'basketballs',
'baskets', 'bass', 'bat', 'batch', 'bath', 'bathroom', 'baton', 'bats',
'batteries', 'battery', 'batting', 'battle', 'battles', 'battling', 'bay', 'be',
'beach', 'beaches', 'bead', 'beads', 'beakers', 'beam', 'beaming', 'beams',
'bean', 'beanbag', 'beanbags', 'beans', 'bear', 'bears', 'beat', 'beating',
'beats', 'beautiful', 'beautifully', 'beautify', 'beauty', 'became', 'because',
'become', 'becomes', 'becoming', 'bed', 'bedroom', 'beds', 'bee', 'bees',
'before', 'beg', 'began', 'begged', 'begging', 'begin', 'beginner', 'beginners',
'beginning', 'beginnings', 'begins', 'begun', 'behalf', 'behave', 'behaved',
'behavior', 'behavioral', 'behaviorally', 'behaviors', 'behind', 'behold',
'being', 'beings', 'belief', 'beliefs', 'believe', 'believed', 'believer',
'believers', 'believes', 'believing', 'bell', 'bellies', 'bells', 'belly',
'belong', 'belonging', 'belongings', 'belongs', 'beloved', 'belt', 'belts',
'ben', 'bench', 'benches', 'benchmark', 'benchmarks', 'bend', 'bending',
'beneath', 'beneficial', 'benefit', 'benefited', 'benefiting', 'benefits',
'bengali', 'benjamin', 'bent', 'bernardino', 'beside', 'besides', 'best', 'bet',
'beta', 'better', 'bettering', 'betterment', 'between', 'beyond', 'bi', 'bias',
'biased', 'bicultural', 'bicycle', 'bicycles', 'big', 'bigger', 'biggest',
'bike', 'bikes', 'biking', 'bilingual', 'bilingualism', 'bilinguals',
'biliteracy', 'biliterate', 'bill', 'bills', 'billy', 'bin', 'bind', 'binder',
'binders', 'binding', 'bingo', 'binoculars', 'bins', 'biographies', 'biography',
'biological', 'biologists', 'biology', 'biomedical', 'biomes', 'biotechnology',
'bird', 'birds', 'birth', 'birthday', 'birthdays', 'bit', 'bits', 'black',
'blacktop', 'blame', 'blank', 'blanket', 'blankets', 'blast', 'bleak', 'blend',
'blended', 'blender', 'blending', 'blends', 'bless', 'blessed', 'blessing',
'blessings', 'blind', 'block', 'blocking', 'blocks', 'blog', 'blogging',
'blogs', 'blood', 'bloom', 'blooming', 'blossom', 'blossoming', 'blow',
'blowing', 'blown', 'bloxels', 'blue', 'blueprints', 'blues', 'bluetooth',
'board', 'boards', 'boasts', 'boat', 'boats', 'bodies', 'bodily', 'body',
'bold', 'bolster', 'bolts', 'bombarded', 'bond', 'bonding', 'bonds', 'bone',
'bones', 'bonus', 'boogie', 'book', 'bookcase', 'bookcases', 'booklets',
'bookmarks', 'books', 'bookshelf', 'bookshelves', 'bookstore', 'bookworms',
'boom', 'booming', 'boomwhackers', 'boost', 'booster', 'boosting', 'boosts',
'boot', 'booth', 'boots', 'border', 'borders', 'bored', 'boredom', 'boring',
'born', 'borough', 'boroughs', 'borrow', 'borrowed', 'borrowing', 'boston',
'bot', 'both', 'bothering', 'bots', 'bottle', 'bottled', 'bottles', 'bottom',
'bottoms', 'bought', 'bounce', 'bouncing', 'bouncy', 'bound', 'boundaries',
'boundary', 'boundless', 'bounds', 'bouts', 'bow', 'bowl', 'bowling', 'bowls',
'bows', 'box', 'boxes', 'boy', 'boys', 'bracelet', 'bracelets', 'bracket',
'brag', 'braille', 'brain', 'brainpop', 'brains', 'brainstorm', 'brainstormed',
'brainstorming', 'branch', 'branches', 'brand', 'brass', 'brave', 'bravery',
'brazil', 'break', 'breakdown', 'breakfast', 'breakfasts', 'breaking',
'breakout', 'breakoutedu', 'breakouts', 'breaks', 'breath', 'breathe',
'breathing', 'breeze', 'brick', 'bricks', 'bridge', 'bridgeport', 'bridges',
'bridging', 'brief', 'bright', 'brighten', 'brighter', 'brightest', 'brightly',
'brilliance', 'brilliant', 'brim', 'brimming', 'bring', 'bringing', 'brings',
'broad', 'broadcast', 'broadcasting', 'broaden', 'broadening', 'broader',
'broadway', 'brochures', 'broke', 'broken', 'bronx', 'brooklyn', 'brother',
'brothers', 'brought', 'brown', 'browse', 'browsing', 'brush', 'brushes',
'brushing', 'bubble', 'bubbles', 'bubbling', 'bubbly', 'bucket', 'buckets',
'bucks', 'buddies', 'budding', 'buddy', 'budget', 'budgetary', 'budgeting',
'budgets', 'buds', 'buffalo', 'bug', 'bugs', 'build', 'builder', 'builders',
'building', 'buildings', 'builds', 'built', 'bulb', 'bulbs', 'bulk', 'bulky',
'bulletin', 'bullied', 'bullying', 'bump', 'bumping', 'bumps', 'bumpy', 'bunch',
'bundle', 'bundles', 'bungee', 'buoyancy', 'burden', 'burdened', 'burdens',
'burlington', 'burma', 'burmese', 'burn', 'burned', 'burner', 'burning',
'burns', 'burst', 'bursting', 'bus', 'bused', 'buses', 'business', 'businesses',
'bussed', 'bustling', 'busy', 'but', 'butler', 'butterflies', 'butterfly',
'button', 'buttons', 'buy', 'buying', 'buzz', 'buzzers', 'buzzing', 'by',
'byod', 'ca', 'cabinet', 'cabinets', 'cable', 'cables', 'cad', 'caddies',
'caddy', 'cadets', 'cafe', 'cafeteria', 'cage', 'cajon', 'cake', 'cakes',
'calculate', 'calculating', 'calculation', 'calculations', 'calculator',
'calculators', 'calculus', 'calendar', 'caliber', 'california', 'calkins',
'call', 'called', 'calligraphy', 'calling', 'calls', 'calm', 'calmer',
'calming', 'calmness', 'calorie', 'calories', 'cam', 'camaraderie', 'cambodia',
'camcorder', 'camden', 'came', 'camera', 'cameras', 'camp', 'campaign',
'camping', 'camps', 'campus', 'campuses', 'can', 'canceling', 'cancelling',
'cancer', 'candidates', 'candy', 'cannot', 'canon', 'cans', 'cant', 'cantonese',
'canvas', 'canvases', 'canyon', 'cap', 'capabilities', 'capability', 'capable',
'capacities', 'capacity', 'cape', 'capital', 'capitalize', 'caps', 'capstone',
'captain', 'captions', 'captivate', 'captivated', 'captivating', 'capture',
'captured', 'captures', 'capturing', 'car', 'carbon', 'card', 'cardboard',
'cardio', 'cardiovascular', 'cards', 'cardstock', 'care', 'cared', 'career',
'careers', 'careful', 'carefully', 'caregiver', 'caregivers', 'cares',
'caretakers', 'caribbean', 'caring', 'carl', 'carle', 'carnegie', 'carnival',
'carol', 'carolina', 'carpet', 'carpeted', 'carpets', 'carried', 'carries',
'carrots', 'carry', 'carrying', 'cars', 'cart', 'cartoon', 'cartoons',
'cartridge', 'cartridges', 'carts', 'carve', 'carving', 'case', 'caseload',
'cases', 'cash', 'cassette', 'cast', 'castle', 'castles', 'cat', 'catalog',
'catalyst', 'catapult', 'catapults', 'catch', 'catching', 'categorical',
'categories', 'categorize', 'categorized', 'category', 'cater', 'catered',
'caterpillar', 'caterpillars', 'caters', 'cats', 'caucasian', 'caught', 'cause',
'caused', 'causes', 'causing', 'ccss', 'cd', 'cds', 'cease', 'ceases',
'ceiling', 'celebrate', 'celebrated', 'celebrates', 'celebrating',
'celebration', 'celebrations', 'cell', 'cello', 'cellos', 'cells', 'cellular',
'cement', 'center', 'centered', 'centers', 'central', 'centrally', 'century',
'ceo', 'ceramic', 'ceramics', 'cereal', 'cerebral', 'ceremony', 'certain',
'certainly', 'certificate', 'certificates', 'certification', 'certified',
'chain', 'chains', 'chair', 'chairs', 'chalk', 'chalkboard', 'challenge',
'challenged', 'challenges', 'challenging', 'chamber', 'champion', 'champions',
'championship', 'chance', 'chances', 'change', 'changed', 'changer', 'changers',
'changes', 'changing', 'channel', 'channeling', 'channels', 'chant', 'chants',
'chaos', 'chaotic', 'chapter', 'chapters', 'character', 'characteristic',
'characteristics', 'characterization', 'characterized', 'characters', 'charge',
'charged', 'charger', 'chargers', 'charging', 'charismatic', 'charity',
'charles', 'charleston', 'charlie', 'charlotte', 'charm', 'charming', 'chart',
'charter', 'charts', 'chase', 'chasing', 'chat', 'chatter', 'chatting',
'chatty', 'cheap', 'cheaper', 'check', 'checked', 'checking', 'checkout',
'checks', 'cheer', 'cheerful', 'cheering', 'cheerleader', 'cheerleaders',
'cheerleading', 'cheers', 'cheese', 'chef', 'chefs', 'chemical', 'chemicals',
'chemistry', 'cherish', 'cherished', 'chess', 'chest', 'chevron', 'chew',
'chicago', 'chick', 'chicken', 'chickens', 'chicks', 'child', 'childhood',
'children', 'childrens', 'chill', 'chimes', 'china', 'chinese', 'chip', 'chips',
'chocolate', 'choice', 'choices', 'choir', 'choirs', 'choose', 'chooses',
'choosing', 'choral', 'chords', 'chore', 'choreography', 'chores', 'chorus',
'chose', 'chosen', 'christmas', 'chrome', 'chromebook', 'chromebooks',
'chronic', 'chronically', 'chunk', 'chunks', 'church', 'churches', 'circle',
'circles', 'circuit', 'circuitry', 'circuits', 'circular', 'circulate',
'circulation', 'circumstance', 'circumstances', 'cite', 'cities', 'citing',
'citizen', 'citizens', 'citizenship', 'city', 'citywide', 'civic', 'civics',
'civil', 'civilization', 'civilizations', 'claim', 'claims', 'clap', 'clarify',
'clarinet', 'clarinets', 'clarity', 'clark', 'classdojo', 'classes', 'classic',
'classical', 'classics', 'classification', 'classified', 'classify',
'classmate', 'classmates', 'classrooms', 'classwork', 'clay', 'clean',
'cleaned', 'cleaner', 'cleaners', 'cleaning', 'cleanliness', 'clear', 'clearer',
'clearly', 'cleats', 'cleveland', 'clever', 'click', 'clickers', 'clicking',
'clicks', 'climate', 'climb', 'climbing', 'clinic', 'clinical', 'clip',
'clipboard', 'clipboards', 'clips', 'clock', 'clocks', 'clorox', 'close',
'closed', 'closely', 'closer', 'closest', 'closet', 'closets', 'closing',
'cloth', 'clothes', 'clothing', 'cloud', 'clouds', 'club', 'clubs', 'clue',
'clues', 'cluster', 'clusters', 'clutter', 'cluttered', 'co', 'coach',
'coaches', 'coaching', 'coal', 'coast', 'coastal', 'coaster', 'coasters',
'coat', 'coats', 'code', 'coded', 'coders', 'codes', 'coding', 'coffee',
'cognition', 'cognitive', 'cognitively', 'cohesive', 'cohort', 'coin',
'coincide', 'coins', 'cold', 'collaborate', 'collaborating', 'collaboration',
'collaborations', 'collaborative', 'collaboratively', 'collaborators',
'collage', 'collages', 'collar', 'colleague', 'colleagues', 'collect',
'collected', 'collecting', 'collection', 'collections', 'collective',
'collectively', 'college', 'colleges', 'collegiate', 'colombia', 'colonial',
'colony', 'color', 'colorado', 'colored', 'colorful', 'coloring', 'colors',
'columbia', 'columbus', 'com', 'combat', 'combination', 'combinations',
'combine', 'combined', 'combines', 'combining', 'combo', 'comers', 'comes',
'comfort', 'comfortable', 'comfortably', 'comforting', 'comforts', 'comfy',
'comic', 'comics', 'coming', 'command', 'commands', 'comment', 'commented',
'comments', 'commercial', 'commercials', 'commit', 'commitment', 'committed',
'committee', 'commodity', 'common', 'commonality', 'commonly', 'commonplace',
'commons', 'communal', 'communicate', 'communicating', 'communication',
'communications', 'communicative', 'communicators', 'communities', 'community',
'commute', 'compact', 'companies', 'company', 'comparable', 'compare',
'compared', 'comparing', 'comparison', 'comparisons', 'compass', 'compasses',
'compassion', 'compassionate', 'compatible', 'compelled', 'compelling',
'compensate', 'compete', 'competed', 'competence', 'competencies', 'competency',
'competent', 'competing', 'competition', 'competitions', 'competitive',
'competitors', 'compile', 'compiled', 'complain', 'complained', 'complaining',
'complaint', 'complaints', 'complement', 'complete', 'completed', 'completely',
'completes', 'completing', 'completion', 'complex', 'complexes', 'complexities',
'complexity', 'complicated', 'compliment', 'compliments', 'comply', 'component',
'components', 'compose', 'composed', 'composer', 'composers', 'composing',
'composition', 'compositions', 'compost', 'composting', 'compound', 'compounds',
'comprehend', 'comprehending', 'comprehensible', 'comprehension',
'comprehensive', 'comprise', 'comprised', 'compromise', 'computation',
'computational', 'computations', 'compute', 'computer', 'computerized',
'computers', 'computing', 'concentrate', 'concentrated', 'concentrating',
'concentration', 'concept', 'concepts', 'conceptual', 'conceptualize',
'concern', 'concerned', 'concerning', 'concerns', 'concert', 'concerts',
'concise', 'conclude', 'conclusion', 'conclusions', 'concrete', 'concretely',
'condition', 'conditioner', 'conditioning', 'conditions', 'conducive',
'conduct', 'conducted', 'conducting', 'cone', 'cones', 'conference',
'conferences', 'conferencing', 'confidence', 'confident', 'confidential',
'confidently', 'configurations', 'confined', 'confines', 'confirms', 'conflict',
'conflicts', 'confront', 'confronted', 'confused', 'confusing', 'confusion',
'congo', 'congressional', 'conjunction', 'connect', 'connected', 'connecticut',
'connecting', 'connection', 'connections', 'connectivity', 'connects',
'conquer', 'conscientious', 'conscious', 'consciousness', 'consecutive',
'consequences', 'consequently', 'conservation', 'conserve', 'consider',
'considerable', 'considerably', 'considerate', 'consideration', 'considered',
'considering', 'consist', 'consistency', 'consistent', 'consistently',
'consisting', 'consists', 'console', 'consonant', 'constant', 'constantly',
'constitution', 'constraints', 'construct', 'constructed', 'constructing',
'construction', 'constructions', 'constructive', 'constructively', 'consumable',
'consumables', 'consume', 'consumed', 'consumer', 'consumers', 'consuming',
'consumption', 'contact', 'contagious', 'contain', 'contained', 'container',
'containers', 'containing', 'contains', 'contemporary', 'contend', 'content',
'contents', 'contest', 'contests', 'context', 'contexts', 'continent',
'continents', 'continual', 'continually', 'continuation', 'continue',
'continued', 'continues', 'continuing', 'continuity', 'continuous',
'continuously', 'continuum', 'contract', 'contrast', 'contrasting',
'contribute', 'contributed', 'contributes', 'contributing', 'contribution',
'contributions', 'contributors', 'control', 'controlled', 'controllers',
'controlling', 'controls', 'convenience', 'convenient', 'conventional',
'conventions', 'conversation', 'conversational', 'conversations', 'converse',
'convert', 'converted', 'converting', 'convex', 'convey', 'convince',
'convinced', 'cook', 'cookbook', 'cookbooks', 'cookie', 'cookies', 'cooking',
'cool', 'cooler', 'coolest', 'coop', 'cooperate', 'cooperating', 'cooperation',
'cooperative', 'cooperatively', 'coordinate', 'coordinated', 'coordination',
'coordinator', 'cope', 'copied', 'copier', 'copies', 'coping', 'copper', 'copy',
'copying', 'coral', 'cord', 'cords', 'core', 'cores', 'corn', 'corner',
'corners', 'cornerstone', 'corporation', 'correct', 'correcting', 'corrections',
'correctly', 'correlate', 'correlated', 'correlates', 'correlation',
'correspond', 'correspondence', 'corresponding', 'cost', 'costly', 'costs',
'costume', 'costumes', 'cotton', 'couch', 'couches', 'could', 'council',
'counseling', 'counselor', 'counselors', 'count', 'counted', 'counter',
'counterparts', 'counters', 'counties', 'counting', 'countless', 'countries',
'country', 'counts', 'county', 'couple', 'coupled', 'courage', 'courageous',
'course', 'courses', 'coursework', 'court', 'courts', 'courtyard', 'cousins',
'cover', 'covered', 'covering', 'covers', 'coveted', 'covey', 'cow',
'coworkers', 'cows', 'cozy', 'cpr', 'cps', 'crack', 'cracked', 'crackers',
'cracks', 'craft', 'crafted', 'crafting', 'crafts', 'craftsmanship', 'crammed',
'cramped', 'crash', 'crate', 'crates', 'crave', 'craving', 'crawl', 'crayola',
'crayon', 'crayons', 'crazy', 'cream', 'create', 'created', 'creates',
'creating', 'creation', 'creations', 'creative', 'creatively', 'creativeness',
'creativity', 'creator', 'creators', 'creature', 'creatures', 'credible',
'credit', 'credits', 'creed', 'creek', 'creole', 'crew', 'cricut', 'crime',
'crimes', 'criminal', 'crisis', 'crisp', 'criss', 'criteria', 'critical',
'critically', 'criticism', 'critique', 'critiquing', 'crop', 'crops', 'cross',
'crossed', 'crossfit', 'crossing', 'crowd', 'crowded', 'crucial', 'cry',
'crying', 'cs', 'ct', 'cuba', 'cubbies', 'cubby', 'cube', 'cubelets', 'cubes',
'cuddle', 'cue', 'cues', 'culinary', 'culminate', 'culminating', 'culmination',
'cultivate', 'cultivated', 'cultivates', 'cultivating', 'cultural',
'culturally', 'culture', 'cultures', 'cumbersome', 'cup', 'cups', 'curb',
'cure', 'curiosities', 'curiosity', 'curious', 'curl', 'curling', 'current',
'currently', 'curricula', 'curricular', 'curriculum', 'curriculums', 'cursive',
'curve', 'cushion', 'cushioned', 'cushions', 'custodians', 'custom', 'customer',
'customers', 'customize', 'customized', 'customs', 'cut', 'cute', 'cuts',
'cutter', 'cutting', 'cvc', 'cyber', 'cycle', 'cycles', 'cycling', 'cylinders',
'cymbal', 'cymbals', 'dad', 'dade', 'dads', 'dahl', 'daily', 'dairy', 'dakota',
'dallas', 'damage', 'damaged', 'damaging', 'dance', 'dancers', 'dances',
'dancing', 'danger', 'dangerous', 'dangers', 'dare', 'dark', 'dash', 'data',
'database', 'databases', 'date', 'dated', 'dates', 'daughter', 'daughters',
'daunting', 'david', 'dawn', 'daycare', 'days', 'dc', 'de', 'dead', 'deadlines',
'deaf', 'deal', 'dealing', 'deals', 'dealt', 'dear', 'dearly', 'death',
'debate', 'debaters', 'debates', 'debating', 'debt', 'decade', 'decades',
'december', 'decent', 'decide', 'decided', 'deciding', 'decimals', 'decipher',
'decision', 'decisions', 'deck', 'decline', 'decodable', 'decode', 'decoding',
'decompose', 'decomposing', 'decomposition', 'decompress', 'decor', 'decorate',
'decorated', 'decorating', 'decorations', 'decorative', 'decrease', 'decreased',
'decreases', 'decreasing', 'dedicate', 'dedicated', 'dedication', 'deemed',
'deep', 'deepen', 'deepening', 'deeper', 'deeply', 'defeat', 'defeated',
'defend', 'defense', 'deficiencies', 'deficiency', 'deficient', 'deficit',
'deficits', 'define', 'defined', 'defines', 'defining', 'definite',
'definitely', 'definition', 'definitions', 'defy', 'defying', 'degree',
'degrees', 'delaware', 'delay', 'delayed', 'delays', 'delicate', 'delicious',
'delight', 'delighted', 'delightful', 'deliver', 'delivered', 'delivering',
'delivery', 'dell', 'delta', 'delve', 'delving', 'demand', 'demanding',
'demands', 'democracy', 'democratic', 'demographic', 'demographics',
'demonstrate', 'demonstrated', 'demonstrates', 'demonstrating', 'demonstration',
'demonstrations', 'denham', 'denied', 'dense', 'density', 'dental', 'denver',
'deny', 'deodorant', 'department', 'departmentalized', 'departments', 'depend',
'dependable', 'dependent', 'dependents', 'depending', 'depends', 'depleted',
'deployed', 'deployment', 'depressed', 'depression', 'deprived', 'depth',
'depths', 'descent', 'describe', 'described', 'describes', 'describing',
'description', 'descriptions', 'descriptive', 'desert', 'deserts', 'deserve',
'deserved', 'deserves', 'deserving', 'design', 'designate', 'designated',
'designed', 'designer', 'designers', 'designing', 'designs', 'desirable',
'desire', 'desired', 'desires', 'desk', 'desks', 'desktop', 'desktops',
'desperate', 'desperately', 'despite', 'destination', 'destined', 'destiny',
'destroyed', 'detail', 'detailed', 'details', 'detectives', 'deter',
'detergent', 'determination', 'determine', 'determined', 'determining',
'detrimental', 'detroit', 'devastated', 'devastating', 'devastation', 'develop',
'developed', 'developing', 'development', 'developmental', 'developmentally',
'developments', 'develops', 'device', 'devices', 'devise', 'devote', 'devoted',
'devour', 'dewey', 'dexterity', 'diabetes', 'diagnosed', 'diagnoses',
'diagnosis', 'diagnostic', 'diagram', 'diagrams', 'dialects', 'dialogue',
'diamond', 'diamonds', 'diaries', 'diary', 'dicamillo', 'dice', 'dictate',
'dictation', 'dictionaries', 'dictionary', 'did', 'die', 'died', 'diego',
'diet', 'diets', 'differ', 'difference', 'differences', 'different',
'differentiate', 'differentiated', 'differentiating', 'differentiation',
'differently', 'differing', 'difficult', 'difficulties', 'difficulty',
'diffuser', 'dig', 'digest', 'digging', 'digit', 'digital', 'digitally',
'dignity', 'digraphs', 'dilapidated', 'dilemma', 'diligence', 'diligent',
'diligently', 'dimension', 'dimensional', 'dimensions', 'diminish',
'diminished', 'dinner', 'dinosaur', 'dinosaurs', 'dioramas', 'diploma',
'diplomas', 'dire', 'direct', 'directed', 'directing', 'direction',
'directional', 'directions', 'directly', 'director', 'directors', 'dirt',
'dirty', 'disabilities', 'disability', 'disabled', 'disadvantage',
'disadvantaged', 'disadvantages', 'disagree', 'disappear', 'disappointed',
'disappointment', 'disaster', 'disasters', 'disc', 'disciplinary', 'discipline',
'disciplined', 'disciplines', 'discomfort', 'disconnect', 'disconnected',
'discourage', 'discouraged', 'discouraging', 'discourse', 'discover',
'discovered', 'discoveries', 'discovering', 'discovery', 'discreetly',
'discrimination', 'discs', 'discuss', 'discussed', 'discussing', 'discussion',
'discussions', 'disease', 'diseases', 'disenfranchised', 'disengaged', 'dish',
'disheartening', 'dishes', 'disinfectant', 'disinfecting', 'disk', 'disks',
'dislike', 'dislikes', 'dismissal', 'disney', 'disorder', 'disorders',
'disorganized', 'disparity', 'dispenser', 'displaced', 'display', 'displayed',
'displaying', 'displays', 'disposable', 'disposal', 'disrepair', 'disrupt',
'disrupting', 'disruption', 'disruptions', 'disruptive', 'dissect',
'dissecting', 'dissection', 'dissections', 'disservice', 'distance',
'distances', 'distant', 'distinct', 'distinction', 'distinguish',
'distinguished', 'distract', 'distracted', 'distracting', 'distraction',
'distractions', 'distracts', 'distress', 'distribute', 'distributed',
'distribution', 'district', 'districts', 'disturb', 'disturbance',
'disturbances', 'disturbed', 'disturbing', 'ditch', 'dive', 'divergent',
'diverse', 'diversified', 'diversify', 'diversity', 'divide', 'divided',
'divider', 'dividers', 'dividing', 'diving', 'division', 'divorce', 'divorced',
'dixie', 'dna', 'do', 'doc', 'docs', 'doctor', 'doctors', 'document',
'documentaries', 'documentary', 'documentation', 'documented', 'documenting',
'documents', 'dodge', 'doers', 'does', 'dog', 'dogs', 'doh', 'doing', 'dojo',
'doll', 'dollar', 'dollars', 'dollhouse', 'dolls', 'domain', 'domains',
'domestic', 'dominant', 'dominated', 'dominican', 'dominoes', 'donate',
'donated', 'donating', 'donation', 'donations', 'done', 'donor', 'donors',
'donorschoose', 'dont', 'doodle', 'doodler', 'door', 'doors', 'dose', 'dot',
'dots', 'double', 'doubled', 'doubt', 'dough', 'down', 'download', 'downloaded',
'downloading', 'downs', 'downtown', 'dozen', 'dozens', 'dr', 'draft',
'drafting', 'drafts', 'drag', 'dragon', 'drama', 'dramas', 'dramatic',
'dramatically', 'drastic', 'drastically', 'draw', 'drawer', 'drawers',
'drawing', 'drawings', 'drawn', 'draws', 'dread', 'dreaded', 'dream',
'dreambox', 'dreamed', 'dreamers', 'dreaming', 'dreams', 'dress', 'dressed',
'dressing', 'drew', 'dribble', 'dribbling', 'dried', 'drill', 'drills', 'drink',
'drinking', 'drinks', 'drive', 'driven', 'driver', 'drivers', 'drives',
'driving', 'drone', 'drones', 'drop', 'dropped', 'dropping', 'drops', 'drought',
'drug', 'drugs', 'drum', 'drumline', 'drumming', 'drums', 'dry', 'dryer',
'drying', 'dslr', 'dual', 'duck', 'duct', 'due', 'dull', 'dumbbells',
'durability', 'durable', 'duration', 'durham', 'during', 'dust', 'dusty',
'duties', 'duty', 'dvd', 'dvds', 'dwindling', 'dye', 'dying', 'dynamath',
'dynamic', 'dynamics', 'dysgraphia', 'dyslexia', 'dyslexic', 'dystopian',
'each', 'eachother', 'eager', 'eagerly', 'eagerness', 'eagle', 'eagles', 'ear',
'earbuds', 'earlier', 'earliest', 'early', 'earn', 'earned', 'earning',
'earphones', 'ears', 'earth', 'earthquake', 'earthquakes', 'ease', 'easel',
'easels', 'easier', 'easiest', 'easily', 'east', 'eastern', 'easy', 'eat',
'eaten', 'eating', 'eats', 'ebd', 'ebooks', 'ec', 'echo', 'eclectic', 'eclipse',
'eco', 'ecological', 'ecology', 'economic', 'economical', 'economically',
'economics', 'economy', 'ecosystem', 'ecosystems', 'ecstatic', 'ect', 'ecuador',
'ed', 'edge', 'edges', 'edible', 'edison', 'edit', 'edited', 'editing',
'edition', 'editions', 'edmodo', 'edu', 'educate', 'educated', 'educates',
'educating', 'education', 'educational', 'educationally', 'educations',
'educator', 'educators', 'edutopia', 'edward', 'effect', 'effected',
'effective', 'effectively', 'effectiveness', 'effects', 'efficacy',
'efficiency', 'efficient', 'efficiently', 'effort', 'effortlessly', 'efforts',
'egg', 'eggs', 'egypt', 'eight', 'eighteen', 'eighth', 'eighty', 'einstein',
'either', 'el', 'ela', 'elaborate', 'elastic', 'elated', 'elbow', 'eld',
'elected', 'election', 'elections', 'elective', 'electives', 'electric',
'electrical', 'electricity', 'electronic', 'electronically', 'electronics',
'electrophoresis', 'element', 'elementary', 'elements', 'elephant', 'elevate',
'elevated', 'eleven', 'eleventh', 'eligibility', 'eligible', 'eliminate',
'eliminated', 'eliminates', 'eliminating', 'elite', 'ell', 'elliptical', 'ells',
'elmo', 'else', 'elsewhere', 'email', 'emails', 'embark', 'embarked',
'embarking', 'embarrassed', 'embarrassing', 'embarrassment', 'embed',
'embedded', 'embodies', 'embody', 'embrace', 'embraced', 'embraces',
'embracing', 'emerge', 'emergency', 'emergent', 'emerging', 'emotion',
'emotional', 'emotionally', 'emotions', 'empathetic', 'empathize', 'empathy',
'emphasis', 'emphasize', 'emphasized', 'emphasizes', 'emphasizing', 'employ',
'employed', 'employee', 'employees', 'employers', 'employment', 'empower',
'empowered', 'empowering', 'empowerment', 'empowers', 'empty', 'emulate',
'enable', 'enabled', 'enables', 'enabling', 'encompass', 'encompasses',
'encompassing', 'encounter', 'encountered', 'encounters', 'encourage',
'encouraged', 'encouragement', 'encourages', 'encouraging', 'end', 'endangered',
'endearing', 'endeavor', 'endeavors', 'ended', 'ending', 'endings', 'endless',
'endlessly', 'ends', 'endurance', 'endure', 'endured', 'enduring', 'energetic',
'energies', 'energize', 'energized', 'energizing', 'energy', 'enforce',
'engage', 'engaged', 'engagement', 'engages', 'engaging', 'engine', 'engineer',
'engineering', 'engineers', 'engines', 'england', 'english', 'engrossed',
'engulfed', 'enhance', 'enhanced', 'enhancement', 'enhances', 'enhancing',
'enjoy', 'enjoyable', 'enjoyed', 'enjoying', 'enjoyment', 'enjoys', 'enl',
'enlarge', 'enlarged', 'enlighten', 'enlightening', 'enormous', 'enough',
'enrich', 'enriched', 'enriches', 'enriching', 'enrichment', 'enroll',
'enrolled', 'enrolling', 'enrollment', 'enrolls', 'ensemble', 'ensembles',
'ensure', 'ensures', 'ensuring', 'entails', 'enter', 'entered', 'entering',
'enterprise', 'enters', 'entertain', 'entertained', 'entertaining',
'entertainment', 'enthralled', 'enthused', 'enthusiasm', 'enthusiastic',
'enthusiastically', 'entice', 'enticing', 'entire', 'entirely', 'entirety',
'entitled', 'entrance', 'entrepreneurs', 'entrepreneurship', 'entries',
'entrusted', 'entry', 'envelopes', 'enviornment', 'enviroment', 'environment',
'environmental', 'environmentally', 'environments', 'envision', 'epic',
'epidemic', 'epson', 'equal', 'equality', 'equalizer', 'equally', 'equals',
'equation', 'equations', 'equip', 'equipment', 'equipped', 'equipping',
'equitable', 'equity', 'equivalent', 'era', 'erasable', 'erase', 'erased',
'eraser', 'erasers', 'erasing', 'ergonomic', 'ergonomically', 'eric', 'erosion',
'error', 'errors', 'escape', 'escaping', 'ese', 'esl', 'esol', 'especially',
'essay', 'essays', 'essence', 'essential', 'essentially', 'essentials',
'establish', 'established', 'establishing', 'esteem', 'estimate', 'estrada',
'et', 'etc', 'ethic', 'ethical', 'ethics', 'ethiopia', 'ethnic', 'ethnically',
'ethnicities', 'ethnicity', 'etiquette', 'europe', 'european', 'ev3',
'evaluate', 'evaluating', 'evaluation', 'evans', 'even', 'evening', 'evenings',
'event', 'events', 'eventually', 'ever', 'everlasting', 'every', 'everybody',
'everyday', 'everyone', 'everything', 'everywhere', 'evidence', 'evidenced',
'evident', 'evolution', 'evolve', 'evolved', 'evolving', 'ex', 'exact',
'exactly', 'exam', 'examination', 'examine', 'examining', 'example', 'examples',
'exams', 'exceed', 'exceeded', 'exceeding', 'exceeds', 'excel', 'excelled',
'excellence', 'excellent', 'excelling', 'except', 'exception', 'exceptional',
'exceptionalities', 'exceptionally', 'exceptions', 'excercise', 'excerpts',
'excess', 'excessive', 'exchange', 'exchanging', 'excitable', 'excite',
'excited', 'excitedly', 'excitement', 'excites', 'exciting', 'exclaimed',
'exclusively', 'excuse', 'excuses', 'execute', 'executing', 'executive',
'exemplary', 'exemplify', 'exercise', 'exercisers', 'exercises', 'exercising',
'exert', 'exhausted', 'exhausting', 'exhibit', 'exhibiting', 'exhibition',
'exhibitions', 'exhibits', 'exhilarating', 'exist', 'existed', 'existence',
'existent', 'existing', 'exists', 'exit', 'exited', 'exiting', 'expand',
'expanded', 'expanding', 'expands', 'expansion', 'expect', 'expectation',
'expectations', 'expected', 'expecting', 'expects', 'expedition',
'expeditionary', 'expeditions', 'expel', 'expend', 'expense', 'expenses',
'expensive', 'experience', 'experienced', 'experiences', 'experiencing',
'experiential', 'experiment', 'experimental', 'experimentation', 'experimented',
'experimenting', 'experiments', 'expert', 'expertise', 'experts', 'explain',
'explained', 'explaining', 'explains', 'explanation', 'explanations',
'explicit', 'explicitly', 'explode', 'exploration', 'explorations',
'exploratory', 'explore', 'explored', 'explorer', 'explorers', 'explores',
'exploring', 'explosion', 'expo', 'exponential', 'exponentially', 'expose',
'exposed', 'exposes', 'exposing', 'expository', 'exposure', 'exposures',
'express', 'expressed', 'expresses', 'expressing', 'expression', 'expressions',
'expressive', 'extend', 'extended', 'extending', 'extends', 'extension',
'extensions', 'extensive', 'extensively', 'extent', 'external', 'extra',
'extracurricular', 'extraordinary', 'extras', 'extreme', 'extremely',
'extrinsic', 'exuberant', 'exude', 'eye', 'eyed', 'eyes', 'fables', 'fabric',
'fabulous', 'face', 'facebook', 'faced', 'faces', 'facet', 'faceted', 'facets',
'facial', 'facilitate', 'facilitated', 'facilitates', 'facilitating',
'facilitator', 'facilities', 'facility', 'facing', 'fact', 'factor',
'factories', 'factors', 'factory', 'facts', 'factual', 'faculty', 'fail',
'failed', 'failing', 'fails', 'failure', 'failures', 'fair', 'fairly',
'fairness', 'fairs', 'fairy', 'faith', 'fake', 'fall', 'fallen', 'falling',
'falls', 'familial', 'familiar', 'familiarity', 'familiarize', 'families',
'family', 'famous', 'fan', 'fancy', 'fans', 'fantastic', 'fantasy', 'far',
'farm', 'farmer', 'farmers', 'farming', 'farmland', 'farms', 'farther',
'fascinated', 'fascinating', 'fascination', 'fashion', 'fashioned', 'fast',
'faster', 'fastest', 'fat', 'father', 'fathers', 'fathom', 'fatigue', 'fault',
'favor', 'favorite', 'favorites', 'fear', 'fearless', 'fears', 'feasible',
'feat', 'feathers', 'feature', 'featured', 'features', 'featuring', 'february',
'fed', 'federal', 'federally', 'fee', 'feed', 'feedback', 'feeder', 'feeding',
'feeds', 'feel', 'feeling', 'feelings', 'feels', 'fees', 'feet', 'fell',
'fellow', 'felt', 'female', 'females', 'feminine', 'fence', 'ferguson',
'festival', 'festivals', 'few', 'fewer', 'ffa', 'fi', 'fiber', 'fiction',
'fictional', 'fidget', 'fidgeting', 'fidgets', 'fidgety', 'field', 'fields',
'fierce', 'fifteen', 'fifth', 'fifty', 'fight', 'fighters', 'fighting',
'fights', 'figurative', 'figure', 'figured', 'figures', 'figuring', 'filament',
'file', 'files', 'filing', 'filipino', 'fill', 'filled', 'filling', 'fills',
'film', 'filming', 'films', 'filter', 'filters', 'filtration', 'final',
'finally', 'finance', 'finances', 'financial', 'financially', 'find', 'finding',
'findings', 'finds', 'fine', 'finest', 'finger', 'fingers', 'fingertips',
'finish', 'finished', 'finishers', 'finishes', 'finishing', 'fire', 'fired',
'firefighters', 'fires', 'firing', 'firm', 'firmly', 'first', 'firsthand',
'firsties', 'fish', 'fishing', 'fist', 'fit', 'fitbit', 'fitbits', 'fitness',
'fits', 'fitting', 'five', 'fives', 'fix', 'fixed', 'fixing', 'fl', 'flag',
'flags', 'flair', 'flame', 'flannel', 'flash', 'flashcards', 'flashlight',
'flashlights', 'flat', 'flavor', 'flex', 'flexibility', 'flexible', 'flexibly',
'fliers', 'flies', 'flight', 'flint', 'flip', 'flipped', 'flipping', 'fll',
'float', 'floating', 'flocabulary', 'flood', 'flooded', 'flooding', 'floods',
'floor', 'flooring', 'floors', 'florescent', 'florida', 'flourish',
'flourishing', 'flow', 'flower', 'flowers', 'flowing', 'flows', 'flu',
'fluency', 'fluent', 'fluently', 'fluid', 'fluorescent', 'flute', 'fly',
'flyers', 'flying', 'foam', 'focal', 'focus', 'focused', 'focuses', 'focusing',
'foil', 'fold', 'foldable', 'foldables', 'folded', 'folder', 'folders',
'folding', 'folk', 'folktales', 'follow', 'followed', 'following', 'follows',
'fond', 'font', 'food', 'foods', 'foot', 'footage', 'football', 'footballs',
'foothills', 'footing', 'footprint', 'footrest', 'for', 'force', 'forced',
'forces', 'forcing', 'forefront', 'foreign', 'foremost', 'forensic',
'forensics', 'foresee', 'forest', 'forests', 'forever', 'forge', 'forget',
'forgetting', 'forgot', 'forgotten', 'fork', 'form', 'formal', 'formally',
'format', 'formation', 'formative', 'formats', 'formed', 'former', 'formerly',
'forming', 'forms', 'formula', 'formulas', 'formulate', 'fort', 'forth',
'fortunate', 'fortunately', 'fortune', 'forty', 'forum', 'forward', 'fossil',
'fossils', 'foster', 'fostered', 'fostering', 'fosters', 'fought', 'found',
'foundation', 'foundational', 'foundations', 'founded', 'founding', 'fountain',
'fountains', 'four', 'fourteen', 'fourth', 'fourths', 'fraction', 'fractions',
'fragile', 'frame', 'framed', 'frames', 'framework', 'frameworks', 'france',
'francisco', 'frank', 'franklin', 'frankly', 'fred', 'frederick', 'free',
'freedom', 'freeing', 'freely', 'frees', 'freeze', 'freezing', 'french',
'frequency', 'frequent', 'frequently', 'fresh', 'freshman', 'freshmen',
'friction', 'friday', 'fridays', 'fridge', 'friend', 'friendly', 'friends',
'friendship', 'friendships', 'frisbee', 'frisbees', 'frog', 'frogs', 'from',
'front', 'frozen', 'fruit', 'fruition', 'fruits', 'frustrate', 'frustrated',
'frustrating', 'frustration', 'frustrations', 'fsa', 'fuel', 'fueled', 'fuels',
'fulfill', 'fulfilled', 'fulfilling', 'fulfillment', 'fulfills', 'full',
'fuller', 'fullest', 'fully', 'fun', 'function', 'functional', 'functionality',
'functioning', 'functions', 'fund', 'fundamental', 'fundamentals', 'funded',
'funding', 'fundraise', 'fundraiser', 'fundraisers', 'fundraising', 'funds',
'funny', 'furnish', 'furniture', 'further', 'furthering', 'furthermore',
'futon', 'future', 'futures', 'ga', 'gadgets', 'gaiam', 'gain', 'gained',
'gaining', 'gains', 'galaxy', 'galleries', 'gallery', 'gallon', 'game',
'gamers', 'games', 'gaming', 'gamut', 'gang', 'gangs', 'gap', 'gaps', 'garage',
'garbage', 'garden', 'gardening', 'gardens', 'gas', 'gate', 'gates', 'gateway',
'gather', 'gathered', 'gathering', 'gauge', 'gave', 'gear', 'geared', 'gearing',
'gears', 'gel', 'gem', 'gender', 'general', 'generalize', 'generally',
'generate', 'generated', 'generating', 'generation', 'generational',
'generations', 'generosity', 'generous', 'generously', 'genetic', 'genetics',
'genius', 'geniuses', 'genre', 'genres', 'gentle', 'gentlemen', 'gently',
'genuine', 'genuinely', 'geographic', 'geographical', 'geographically',
'geography', 'geology', 'geometric', 'geometry', 'george', 'georgia', 'germ',
'german', 'germs', 'gesture', 'gestures', 'get', 'gets', 'getters', 'getting',
'giant', 'gift', 'gifted', 'gifts', 'giggle', 'giggles', 'giggling',
'gingerbread', 'girl', 'girls', 'give', 'given', 'giver', 'gives', 'giving',
'glad', 'gladly', 'glance', 'glass', 'glasses', 'glassware', 'glaze', 'glazes',
'glimpse', 'glitter', 'global', 'globally', 'globe', 'globes', 'glockenspiels',
'gloves', 'glow', 'glue', 'glued', 'gluing', 'go', 'goal', 'goalie', 'goals',
'goats', 'god', 'goes', 'goggle', 'goggles', 'gogh', 'going', 'gold',
'goldberg', 'golden', 'goldfish', 'goldilocks', 'golf', 'gone', 'gonoodle',
'good', 'goodbye', 'goodies', 'goodness', 'goods', 'goofy', 'google', 'gopro',
'got', 'gotten', 'government', 'gpa', 'gps', 'grab', 'grabbing', 'grabs',
'grace', 'gracious', 'graciously', 'grade', 'graded', 'grader', 'graders',
'grades', 'grading', 'gradually', 'graduate', 'graduated', 'graduates',
'graduating', 'graduation', 'graffiti', 'grammar', 'grand', 'grandchildren',
'grandin', 'grandparent', 'grandparents', 'granola', 'grant', 'granted',
'grants', 'graph', 'graphic', 'graphics', 'graphing', 'graphs', 'grapple',
'grasp', 'grasped', 'grasping', 'grass', 'grateful', 'gratification',
'gratifying', 'gratitude', 'gravitate', 'gravity', 'gray', 'great', 'greater',
'greatest', 'greatly', 'greatness', 'greece', 'greek', 'green', 'greenhouse',
'greenville', 'greet', 'greeted', 'greeting', 'greetings', 'grew', 'grid',
'grip', 'grips', 'grit', 'groceries', 'grocery', 'grooming', 'groove',
'grooving', 'gross', 'ground', 'groundbreaking', 'grounded', 'grounds',
'groundwork', 'group', 'grouped', 'grouping', 'groupings', 'groups', 'grove',
'grow', 'growing', 'growling', 'grown', 'grows', 'growth', 'gruff', 'gt',
'guarantee', 'guaranteed', 'guard', 'guardian', 'guardians', 'guards',
'guatemala', 'guess', 'guest', 'guests', 'guidance', 'guide', 'guided',
'guidelines', 'guides', 'guiding', 'guinea', 'guitar', 'guitars', 'gulf', 'gun',
'guns', 'gusto', 'guy', 'guys', 'gym', 'gymnasium', 'gymnastics', 'gyms', 'ha',
'habit', 'habitat', 'habitats', 'habits', 'hail', 'hair', 'haiti', 'haitian',
'half', 'halfway', 'hall', 'halls', 'hallway', 'hallways', 'hamilton', 'hand',
'handed', 'handful', 'handheld', 'handicapped', 'handicaps', 'handing',
'handle', 'handled', 'handles', 'handling', 'handmade', 'handouts', 'hands',
'handshake', 'handwriting', 'handy', 'hang', 'hanging', 'happen', 'happened',
'happening', 'happenings', 'happens', 'happier', 'happiest', 'happily',
'happiness', 'happy', 'harbor', 'hard', 'hardcover', 'harder', 'hardest',
'hardly', 'hardship', 'hardships', 'hardware', 'hardworking', 'harlem', 'harm',
'harmful', 'harmony', 'harness', 'harry', 'harsh', 'hartford', 'harvest',
'harvested', 'harvesting', 'hassle', 'hat', 'hatch', 'hatched', 'hatching',
'hate', 'hates', 'hats', 'have', 'having', 'hawaii', 'hawaiian', 'hazard',
'hazards', 'hd', 'hdmi', 'he', 'head', 'headaches', 'headed', 'heading',
'headphone', 'headphones', 'heads', 'headset', 'headsets', 'healing', 'health',
'healthcare', 'healthful', 'healthier', 'healthiest', 'healthy', 'hear',
'heard', 'hearing', 'heart', 'heartbeat', 'heartbreaking', 'hearted', 'hearts',
'heartwarming', 'heat', 'heating', 'heaven', 'heavily', 'heavy', 'hectic',
'heidi', 'height', 'heighten', 'heightened', 'heights', 'held', 'helen',
'hello', 'helmets', 'helped', 'helper', 'helpers', 'helpful', 'helping',
'helps', 'hence', 'henry', 'her', 'herb', 'herbs', 'here', 'heritage', 'hermit',
'hero', 'heroes', 'hesitant', 'hesitate', 'hesitation', 'heterogeneous', 'hey',
'hi', 'hidden', 'hide', 'high', 'higher', 'highest', 'highlight', 'highlighted',
'highlighter', 'highlighters', 'highlighting', 'highlights', 'highly',
'highway', 'hiking', 'hilarious', 'hill', 'hills', 'hinder', 'hindered',
'hindering', 'hinders', 'hindi', 'hindrance', 'hints', 'hip', 'hired', 'his',
'hispanic', 'hispanics', 'historians', 'historic', 'historical', 'historically',
'histories', 'history', 'hit', 'hits', 'hitting', 'hmong', 'hobbies', 'hobby',
'hockey', 'hokki', 'hola', 'hold', 'holder', 'holders', 'holding', 'holds',
'hole', 'holes', 'holiday', 'holidays', 'holistic', 'hollywood', 'holocaust',
'home', 'homeless', 'homelessness', 'homemade', 'homeroom', 'homerooms',
'homes', 'hometown', 'homework', 'homey', 'honduras', 'hone', 'honest',
'honestly', 'honesty', 'honing', 'honor', 'honorable', 'honored', 'honoring',
'honors', 'hood', 'hook', 'hooked', 'hooki', 'hooks', 'hoop', 'hoops', 'hop',
'hope', 'hopeful', 'hopefully', 'hopes', 'hoping', 'hopping', 'hopscotch',
'horizon', 'horizons', 'horn', 'horrible', 'horrific', 'horse', 'horseshoe',
'hospital', 'host', 'hosted', 'hosting', 'hosts', 'hot', 'hotel', 'hotels',
'hour', 'hours', 'house', 'housed', 'household', 'households', 'housekeeping',
'houses', 'housing', 'houston', 'how', 'however', 'hp', 'hs', 'http', 'https',
'hub', 'hug', 'huge', 'hugely', 'hugs', 'hula', 'hum', 'human', 'humanities',
'humanity', 'humans', 'humble', 'humbled', 'humidity', 'humor', 'humorous',
'hundred', 'hundreds', 'hung', 'hunger', 'hungry', 'hunt', 'hunter', 'hunting',
'hunts', 'hurdle', 'hurdles', 'hurricane', 'hurt', 'hurting', 'hurts',
'husband', 'hybrid', 'hydrated', 'hydration', 'hydroponic', 'hydroponics',
'hygiene', 'hyper', 'hyperactive', 'hyperactivity', 'hypotheses', 'hypothesis',
'hypothesize', 'ib', 'ice', 'icons', 'ict', 'id', 'idaho', 'idea', 'ideal',
'ideally', 'ideals', 'ideas', 'identification', 'identified', 'identifies',
'identify', 'identifying', 'identities', 'identity', 'ie', 'iep', 'ieps', 'if',
'ignacio', 'ignite', 'ignited', 'ignites', 'igniting', 'ignore', 'ignored',
'ii', 'iii', 'il', 'ill', 'illinois', 'illiteracy', 'illiterate', 'illness',
'illnesses', 'illuminate', 'illustrate', 'illustrated', 'illustrating',
'illustration', 'illustrations', 'illustrator', 'illustrators', 'imac', 'image',
'imagery', 'images', 'imaginable', 'imaginary', 'imagination', 'imaginations',
'imaginative', 'imagine', 'imagined', 'imagining', 'imitate', 'immeasurable',
'immediate', 'immediately', 'immense', 'immensely', 'immerse', 'immersed',
'immersing', 'immersion', 'immersive', 'immigrant', 'immigrants', 'immigrated',
'immigration', 'imovie', 'imovies', 'impact', 'impacted', 'impactful',
'impacting', 'impacts', 'impaired', 'impairment', 'impairments', 'impart',
'impede', 'impedes', 'imperative', 'implement', 'implementation', 'implemented',
'implementing', 'implements', 'implications', 'importance', 'important',
'importantly', 'impossible', 'impoverished', 'impress', 'impressed',
'impression', 'impressionable', 'impressions', 'impressive', 'improve',
'improved', 'improvement', 'improvements', 'improves', 'improving',
'improvisation', 'improvise', 'impulse', 'impulsive', 'in', 'inability',
'inaccessible', 'inadequate', 'inappropriate', 'inaugural', 'incarcerated',
'incarceration', 'incentive', 'incentives', 'inch', 'inches', 'incidence',
'inclement', 'inclined', 'include', 'included', 'includes', 'including',
'inclusion', 'inclusive', 'income', 'incomes', 'incoming', 'incomplete',
'inconsistent', 'incorporate', 'incorporated', 'incorporates', 'incorporating',
'incorporation', 'incorrect', 'incorrectly', 'increase', 'increased',
'increases', 'increasing', 'increasingly', 'incredible', 'incredibly',
'incubator', 'indeed', 'independence', 'independent', 'independently', 'index',
'india', 'indian', 'indiana', 'indianapolis', 'indicate', 'indicated',
'indicates', 'indicator', 'indicators', 'individual', 'individuality',
'individualize', 'individualized', 'individually', 'individuals', 'indoor',
'indoors', 'industrial', 'industries', 'industry', 'inevitable', 'inevitably',
'inexpensive', 'infectious', 'infer', 'inference', 'inferences', 'infested',
'infinite', 'inflatable', 'influence', 'influenced', 'influences',
'influential', 'influx', 'inform', 'informal', 'information', 'informational',
'informative', 'informed', 'infrastructure', 'infuse', 'infused', 'infusing',
'infusion', 'ingenuity', 'ingredient', 'ingredients', 'inhabit', 'inherent',
'inherently', 'inherit', 'inherited', 'inhibit', 'initial', 'initially',
'initiate', 'initiated', 'initiative', 'initiatives', 'injured', 'injuries',
'injury', 'injustice', 'ink', 'innate', 'inner', 'innocence', 'innocent',
'innovate', 'innovated', 'innovating', 'innovation', 'innovations',
'innovative', 'innovators', 'input', 'inquire', 'inquirers', 'inquiries',
'inquiring', 'inquiry', 'inquisitive', 'inquisitiveness', 'ins', 'insatiable',
'insect', 'insects', 'insecure', 'insecurity', 'insert', 'inside', 'insight',
'insightful', 'insights', 'inspiration', 'inspirational', 'inspirations',
'inspire', 'inspired', 'inspires', 'inspiring', 'instability', 'instagram',
'install', 'installation', 'installed', 'instance', 'instances', 'instant',
'instantly', 'instead', 'instill', 'instilled', 'instilling', 'instills',
'institute', 'institution', 'institutions', 'instruct', 'instructed',
'instructing', 'instruction', 'instructional', 'instructions', 'instructor',
'instructors', 'instrument', 'instrumental', 'instruments', 'insufficient',
'insure', 'insurmountable', 'intact', 'intake', 'integral', 'integrate',
'integrated', 'integrates', 'integrating', 'integration', 'integrity',
'intellect', 'intellectual', 'intellectually', 'intelligence', 'intelligences',
'intelligent', 'intend', 'intended', 'intense', 'intensely', 'intensity',
'intensive', 'intensively', 'intent', 'intention', 'intentional',
'intentionally', 'intentions', 'inter', 'interact', 'interacting',
'interaction', 'interactions', 'interactive', 'interactively', 'interactivity',
'interdisciplinary', 'interest', 'interested', 'interesting', 'interests',
'interface', 'interfere', 'interferes', 'intermediate', 'internal',
'internalize', 'international', 'internationally', 'internet', 'internships',
'interpersonal', 'interpret', 'interpretation', 'interpretations',
'interpreting', 'interrupt', 'interrupted', 'interrupting', 'interruption',
'interruptions', 'intertwined', 'intervals', 'intervene', 'intervention',
'interventionist', 'interventions', 'interview', 'interviewed', 'interviewing',
'interviews', 'intimate', 'intimidated', 'intimidating', 'into', 'intonation',
'intramural', 'intricate', 'intrigue', 'intrigued', 'intriguing', 'intrinsic',
'intrinsically', 'introduce', 'introduced', 'introduces', 'introducing',
'introduction', 'introductory', 'intuitive', 'invaluable', 'invent', 'invented',
'inventing', 'invention', 'inventions', 'inventive', 'inventor', 'inventors',
'inventory', 'invest', 'invested', 'investigate', 'investigated',
'investigating', 'investigation', 'investigations', 'investigative',
'investigators', 'investing', 'investment', 'invigorating', 'invisible',
'invite', 'invited', 'invites', 'inviting', 'involve', 'involved',
'involvement', 'involves', 'involving', 'iowa', 'ipad', 'ipads', 'ipevo',
'iphone', 'ipod', 'ipods', 'iq', 'iran', 'iraq', 'iready', 'iron', 'is', 'isd',
'island', 'islander', 'islanders', 'islands', 'isolated', 'isolation', 'issue',
'issued', 'issues', 'istation', 'it', 'itching', 'item', 'items', 'its', 'ivan',
'ixl', 'jack', 'jacket', 'jackets', 'jacks', 'jackson', 'jacqueline', 'jail',
'jam', 'jamaica', 'james', 'jane', 'january', 'japan', 'japanese', 'jar',
'jars', 'jazz', 'jean', 'jefferson', 'jellyfish', 'jenga', 'jeopardy', 'jersey',
'jerseys', 'jet', 'jewelry', 'jitters', 'job', 'jobs', 'joe', 'john', 'johnson',
'join', 'joined', 'joining', 'joint', 'joke', 'jokes', 'jones', 'jordan',
'jose', 'joseph', 'jot', 'journal', 'journaling', 'journalism', 'journalists',
'journals', 'journey', 'journeys', 'joy', 'joyful', 'joyous', 'joys', 'jr',
'jrotc', 'judge', 'judged', 'judgement', 'judges', 'judgment', 'juggle',
'juggling', 'juice', 'juices', 'juliet', 'july', 'jumbo', 'jump', 'jumped',
'jumping', 'jumps', 'jumpstart', 'june', 'jungle', 'junie', 'junior', 'juniors',
'junk', 'just', 'justice', 'justify', 'juvenile', 'jv', 'k4', 'kahoot',
'kansas', 'karate', 'karen', 'kate', 'katrina', 'keen', 'keep', 'keeping',
'keeps', 'keepsake', 'keller', 'ken', 'kennedy', 'kentucky', 'kept', 'keva',
'key', 'keyboard', 'keyboarding', 'keyboards', 'keynote', 'keys', 'khan',
'kick', 'kickball', 'kicked', 'kicking', 'kid', 'kiddo', 'kiddos', 'kidney',
'kids', 'kill', 'killed', 'kiln', 'kind', 'kinder', 'kindergarten',
'kindergartener', 'kindergarteners', 'kindergartens', 'kindergartner',
'kindergartners', 'kinders', 'kindest', 'kindhearted', 'kindle', 'kindles',
'kindly', 'kindness', 'kinds', 'kinect', 'kinesthetic', 'kinesthetically',
'kinetic', 'king', 'kipp', 'kit', 'kitchen', 'kits', 'kleenex', 'knee', 'kneel',
'kneeling', 'knees', 'knew', 'knex', 'knit', 'knives', 'knock', 'know',
'knowing', 'knowledge', 'knowledgeable', 'known', 'knows', 'kore', 'korea',
'korean', 'la', 'lab', 'label', 'labeled', 'labeling', 'labels', 'labor',
'laboratory', 'labs', 'lacing', 'lack', 'lacked', 'lacking', 'lacks',
'lacrosse', 'ladder', 'ladders', 'ladies', 'lady', 'ladybugs', 'laid', 'lake',
'lakes', 'lakeshore', 'lakota', 'laminate', 'laminated', 'laminating',
'lamination', 'laminator', 'lamp', 'lamps', 'land', 'landforms', 'landmarks',
'lands', 'landscape', 'landscapes', 'language', 'languages', 'lap', 'lapboards',
'laps', 'laptop', 'laptops', 'large', 'largely', 'larger', 'largest', 'las',
'laser', 'last', 'lasting', 'lastly', 'lasts', 'latch', 'late', 'lately',
'later', 'latest', 'latin', 'latino', 'latinos', 'laugh', 'laughing', 'laughs',
'laughter', 'launch', 'launched', 'launching', 'laundry', 'lavender', 'law',
'laws', 'lawyers', 'lay', 'layer', 'layers', 'laying', 'layout', 'lays', 'lcd',
'lead', 'leader', 'leaders', 'leadership', 'leading', 'leads', 'leaf', 'league',
'lean', 'leaners', 'leaning', 'leap', 'leaps', 'learned', 'learner', 'learners',
'learnings', 'learns', 'least', 'leather', 'leave', 'leaves', 'leaving',
'lecture', 'lectures', 'lecturing', 'led', 'lee', 'left', 'leftover', 'leg',
'legacy', 'legal', 'legged', 'legible', 'lego', 'legos', 'legs', 'leisure',
'lemonade', 'lend', 'lending', 'lends', 'length', 'lengths', 'lengthy',
'lenovo', 'lens', 'lenses', 'less', 'lessen', 'lesson', 'lessons', 'let',
'lets', 'letter', 'letters', 'letting', 'lettuce', 'level', 'leveled',
'leveling', 'levels', 'leverage', 'lewis', 'lexia', 'lexile', 'lgbtq',
'liberty', 'librarian', 'librarians', 'libraries', 'library', 'lice', 'license',
'licenses', 'lids', 'lie', 'lies', 'life', 'lifelong', 'lifestyle',
'lifestyles', 'lifetime', 'lift', 'lifting', 'light', 'lightbulb', 'lighthouse',
'lighting', 'lightly', 'lightning', 'lights', 'lightweight', 'like', 'liked',
'likelihood', 'likely', 'likes', 'likewise', 'liking', 'limit', 'limitation',
'limitations', 'limited', 'limiting', 'limitless', 'limits', 'lincoln', 'line',
'linear', 'lined', 'lines', 'lingual', 'linguistic', 'linguistically', 'lining',
'link', 'linked', 'linking', 'links', 'linoleum', 'lion', 'lip', 'liquid',
'liquids', 'list', 'listed', 'listen', 'listened', 'listeners', 'listening',
'lists', 'lit', 'literacy', 'literally', 'literary', 'literate', 'literature',
'little', 'littlebits', 'littles', 'littlest', 'live', 'lived', 'lively',
'liven', 'lives', 'living', 'load', 'loaded', 'loads', 'loan', 'local',
'locally', 'locate', 'located', 'locating', 'location', 'locations', 'lock',
'locked', 'locker', 'lockers', 'locking', 'locks', 'lofty', 'log', 'logging',
'logic', 'logical', 'logically', 'login', 'logistics', 'logo', 'logs', 'lonely',
'long', 'longer', 'longest', 'longevity', 'longing', 'look', 'looked',
'looking', 'looks', 'loom', 'loop', 'looped', 'looping', 'loose', 'loosing',
'los', 'lose', 'losing', 'loss', 'lost', 'lot', 'lotion', 'lots', 'lottery',
'loud', 'louder', 'loudly', 'louis', 'louisiana', 'louisville', 'lounge',
'lovable', 'loved', 'lovely', 'lover', 'lovers', 'loves', 'loving', 'low',
'lower', 'lowercase', 'lowered', 'lowering', 'lowest', 'loyal', 'luck',
'luckiest', 'luckily', 'lucky', 'lucy', 'lunch', 'lunches', 'lunchroom',
'lunchtime', 'lungs', 'luther', 'luxuries', 'luxury', 'lying', 'lyrics',
'lysol', 'ma', 'mac', 'macbook', 'mache', 'machine', 'machines', 'mad', 'made',
'madison', 'magazine', 'magazines', 'magformers', 'magic', 'magical', 'magna',
'magnatiles', 'magnet', 'magnetic', 'magnetism', 'magnets', 'magnificent',
'magnifying', 'mail', 'mailbox', 'mailboxes', 'main', 'maine', 'mainly',
'mainstream', 'mainstreamed', 'maintain', 'maintained', 'maintaining',
'maintains', 'maintenance', 'major', 'majority', 'majors', 'makeover', 'maker',
'makers', 'makerspace', 'makerspaces', 'makes', 'makeup', 'makey', 'makeys',
'making', 'malala', 'male', 'males', 'mallet', 'mallets', 'man', 'manage',
'manageable', 'managed', 'management', 'managers', 'managing', 'mandarin',
'mandated', 'mandatory', 'mandela', 'maneuver', 'manga', 'manhattan',
'manifest', 'manipulate', 'manipulated', 'manipulates', 'manipulating',
'manipulation', 'manipulative', 'manipulatives', 'manner', 'mannered',
'manners', 'mantra', 'manual', 'manufacturing', 'map', 'mapping', 'maps',
'marble', 'marbles', 'march', 'marching', 'margaret', 'marginalized', 'maria',
'mariachi', 'marine', 'mark', 'marked', 'marker', 'markers', 'market',
'marketable', 'marketing', 'marking', 'marks', 'mars', 'marshall',
'marshallese', 'martian', 'martin', 'marvel', 'marvelous', 'mary', 'maryland',
'mascot', 'masks', 'mass', 'massachusetts', 'massive', 'master', 'mastered',
'mastering', 'masterpiece', 'masterpieces', 'masters', 'mastery', 'mat',
'match', 'matched', 'matches', 'matching', 'material', 'materials', 'mates',
'math', 'mathematic', 'mathematical', 'mathematically', 'mathematician',
'mathematicians', 'mathematics', 'mats', 'matter', 'matters', 'matthew',
'mature', 'maturing', 'maturity', 'max', 'maximize', 'maximized', 'maximizes',
'maximizing', 'maximum', 'may', 'maya', 'maybe', 'mayo', 'maze', 'mazes', 'md',
'me', 'meager', 'meal', 'meals', 'mean', 'meaning', 'meaningful',
'meaningfully', 'meanings', 'means', 'meant', 'meanwhile', 'measurable',
'measure', 'measured', 'measurement', 'measurements', 'measures', 'measuring',
'mechanical', 'mechanics', 'mechanisms', 'medal', 'media', 'median', 'medical',
'medically', 'medication', 'medicine', 'medieval', 'meditating', 'meditation',
'medium', 'mediums', 'meet', 'meeting', 'meetings', 'meets', 'melodic',
'melodies', 'melody', 'melt', 'melting', 'member', 'members', 'membership',
'memoir', 'memoirs', 'memorable', 'memorial', 'memories', 'memorization',
'memorize', 'memorized', 'memorizing', 'memory', 'memphis', 'men', 'mental',
'mentality', 'mentally', 'mention', 'mentioned', 'mentor', 'mentoring',
'mentors', 'menu', 'menus', 'mercy', 'mere', 'merely', 'merge', 'mesa', 'mesh',
'mess', 'message', 'messages', 'messes', 'messy', 'met', 'metabolism', 'metal',
'metallophones', 'metamorphosis', 'meter', 'method', 'methods', 'metric',
'metro', 'metropolitan', 'mexican', 'mexico', 'mi', 'miami', 'mic', 'mice',
'michael', 'michigan', 'micro', 'microorganisms', 'microphone', 'microphones',
'microscope', 'microscopes', 'microscopic', 'microsoft', 'microwave', 'mid',
'middle', 'midst', 'midwest', 'might', 'mighty', 'migrant', 'migration', 'mild',
'mile', 'mileage', 'miles', 'milestone', 'milestones', 'military', 'milk',
'mill', 'miller', 'million', 'millions', 'mills', 'milwaukee', 'mimic', 'mimio',
'min', 'mind', 'minded', 'mindedness', 'mindful', 'mindfulness', 'minds',
'mindset', 'mindsets', 'mindstorm', 'mindstorms', 'mine', 'minecraft',
'mineral', 'minerals', 'mini', 'miniature', 'minimal', 'minimize', 'minimizing',
'minimum', 'mining', 'minis', 'minneapolis', 'minnesota', 'minor', 'minorities',
'minority', 'minute', 'minutes', 'miracle', 'mirror', 'mirrors', 'misbehavior',
'miscellaneous', 'misconceptions', 'mismatched', 'misplaced', 'miss', 'missed',
'missing', 'mission', 'missions', 'mississippi', 'missouri', 'mistake',
'mistakes', 'misunderstood', 'mit', 'mitosis', 'mix', 'mixed', 'mixer',
'mixing', 'mixture', 'mn', 'mo', 'mobile', 'mobility', 'moby', 'mobymax',
'mock', 'mockingbird', 'modal', 'modalities', 'modality', 'mode', 'model',
'modeled', 'modeling', 'models', 'moderate', 'modern', 'modes', 'modest',
'modification', 'modifications', 'modified', 'modify', 'modifying', 'module',
'modules', 'mold', 'molded', 'molding', 'molds', 'molecular', 'molecules',
'mom', 'moment', 'moments', 'momentum', 'moms', 'monarch', 'monday', 'mondays',
'monetary', 'money', 'monies', 'monitor', 'monitored', 'monitoring', 'monitors',
'monkey', 'mono', 'monolingual', 'monopoly', 'monotony', 'monster', 'monsters',
'montana', 'montessori', 'month', 'monthly', 'months', 'monumental', 'mood',
'moods', 'moon', 'moral', 'morale', 'morals', 'more', 'moreover', 'morning',
'mornings', 'mosaic', 'mosaics', 'most', 'mostly', 'motels', 'mother',
'mothers', 'motion', 'motions', 'motivate', 'motivated', 'motivates',
'motivating', 'motivation', 'motivational', 'motivator', 'motivators', 'motor',
'motors', 'motto', 'mount', 'mountain', 'mountains', 'mounted', 'mounting',
'mouse', 'mouth', 'mouthpiece', 'mouthpieces', 'mouths', 'movable', 'move',
'moveable', 'moved', 'movement', 'movements', 'movers', 'moves', 'movie',
'movies', 'moving', 'mp3', 'mr', 'mrs', 'ms', 'mt', 'much', 'multi', 'multiage',
'multicultural', 'multilingual', 'multimedia', 'multiple', 'multiplication',
'multiply', 'multiplying', 'multipurpose', 'multiracial', 'multisensory',
'multitude', 'mundane', 'mural', 'murals', 'murder', 'muscle', 'muscles',
'muscular', 'museum', 'museums', 'music', 'musical', 'musically', 'musicals',
'musician', 'musicians', 'musicianship', 'must', 'mutual', 'myon', 'myriad',
'myself', 'mysteries', 'mysterious', 'mystery', 'mythology', 'myths', 'name',
'named', 'names', 'naming', 'nancy', 'nap', 'narrative', 'narratives',
'narrator', 'narrow', 'nasa', 'nashville', 'nation', 'national',
'nationalities', 'nationally', 'nations', 'nationwide', 'native', 'natives',
'natural', 'naturally', 'nature', 'navigate', 'navigating', 'navy', 'nc',
'near', 'nearby', 'nearest', 'nearing', 'nearly', 'nearpod', 'neat', 'neatly',
'nebraska', 'necessarily', 'necessary', 'necessities', 'necessity', 'neck',
'necklaces', 'needed', 'neediest', 'needing', 'needles', 'needless', 'needs',
'needy', 'negative', 'negatively', 'negatives', 'negativity', 'neglect',
'neglected', 'negotiate', 'neighbor', 'neighborhood', 'neighborhoods',
'neighboring', 'neighbors', 'neither', 'nelson', 'nepal', 'nepali', 'nerve',
'nervous', 'nest', 'nestled', 'net', 'nets', 'network', 'networking',
'neurological', 'nevada', 'never', 'nevertheless', 'new', 'newark', 'newbery',
'newcomer', 'newcomers', 'newer', 'newest', 'newfound', 'newly', 'news',
'newsela', 'newsletter', 'newsletters', 'newspaper', 'newspapers', 'newton',
'nex', 'next', 'ngss', 'nice', 'nicely', 'nicer', 'niche', 'nigeria', 'night',
'nightly', 'nightmare', 'nights', 'nine', 'nineteen', 'ninety', 'ninja',
'ninth', 'nj', 'no', 'nobody', 'noise', 'noises', 'noisy', 'nominated', 'non',
'none', 'nonetheless', 'nonexistent', 'nonfiction', 'nontraditional',
'nonverbal', 'noodle', 'noodles', 'nook', 'nooks', 'nor', 'norm', 'normal',
'normalcy', 'normally', 'norms', 'north', 'northeast', 'northern', 'northwest',
'nose', 'noses', 'notable', 'notation', 'notch', 'note', 'notebook',
'notebooks', 'noted', 'notes', 'nothing', 'notice', 'noticed', 'notices',
'noticing', 'notion', 'nots', 'nouns', 'nourish', 'nourished', 'nourishing',
'nourishment', 'novel', 'novels', 'novelty', 'november', 'novice', 'now',
'nowadays', 'nowhere', 'nuclear', 'number', 'numbers', 'numeracy', 'numerals',
'numerous', 'nurse', 'nursery', 'nurses', 'nursing', 'nurture', 'nurtured',
'nurtures', 'nurturing', 'nutrient', 'nutrients', 'nutrition', 'nutritional',
'nutritious', 'nuts', 'ny', 'nyc', 'oak', 'oakland', 'oasis', 'obama', 'obese',
'obesity', 'object', 'objective', 'objectives', 'objects', 'obligation',
'observant', 'observation', 'observational', 'observations', 'observe',
'observed', 'observing', 'obsessed', 'obsolete', 'obstacle', 'obstacles',
'obtain', 'obtained', 'obtaining', 'obvious', 'obviously', 'occasion',
'occasional', 'occasionally', 'occasions', 'occupational', 'occupations',
'occupied', 'occupy', 'occur', 'occurred', 'occurrence', 'occurring', 'occurs',
'ocean', 'oceans', 'october', 'odd', 'odds', 'odyssey', 'of', 'offer',
'offered', 'offering', 'offerings', 'offers', 'office', 'officer', 'officers',
'offices', 'official', 'officially', 'offset', 'often', 'oftentimes', 'oh',
'ohio', 'oil', 'oils', 'ok', 'okay', 'oklahoma', 'old', 'older', 'oldest',
'olds', 'olympiad', 'olympic', 'olympics', 'omaha', 'on', 'once', 'one', 'ones',
'oneself', 'ongoing', 'online', 'only', 'onto', 'open', 'opened', 'opening',
'openly', 'opens', 'opera', 'operate', 'operated', 'operates', 'operating',
'operation', 'operations', 'opinion', 'opinions', 'opportunities',
'opportunity', 'opposed', 'opposite', 'opt', 'opted', 'optimal', 'optimism',
'optimistic', 'optimize', 'optimum', 'option', 'optional', 'options', 'or',
'oral', 'orally', 'orange', 'orchestra', 'order', 'ordered', 'ordering',
'orderly', 'orders', 'ordinary', 'oregon', 'orff', 'org', 'organ', 'organic',
'organism', 'organisms', 'organization', 'organizational', 'organizations',
'organize', 'organized', 'organizer', 'organizers', 'organizing', 'organs',
'orientation', 'oriented', 'origami', 'origin', 'original', 'originally',
'origins', 'orlando', 'orleans', 'orthopedic', 'osmo', 'osmos', 'ot', 'other',
'others', 'otherwise', 'ottomans', 'ours', 'out', 'outbursts', 'outcome',
'outcomes', 'outdated', 'outdoor', 'outdoors', 'outer', 'outfit', 'outgoing',
'outlet', 'outlets', 'outline', 'outlined', 'outlook', 'output', 'outreach',
'outs', 'outside', 'outsiders', 'outskirts', 'outspoken', 'outstanding', 'oven',
'over', 'overall', 'overcome', 'overcoming', 'overcrowded', 'overflow',
'overflowing', 'overhead', 'overjoyed', 'overload', 'overlook', 'overlooked',
'overly', 'overnight', 'overrated', 'overseas', 'oversized', 'overstimulated',
'overweight', 'overwhelmed', 'overwhelming', 'overwhelmingly', 'owe', 'owl',
'owls', 'own', 'owned', 'owners', 'ownership', 'owning', 'oxygen', 'ozobot',
'ozobots', 'pa', 'pablo', 'pace', 'paced', 'paces', 'pacific', 'pacing', 'pack',
'package', 'packages', 'packed', 'packet', 'packets', 'packing', 'packs', 'pad',
'padded', 'padding', 'paddles', 'padlet', 'pads', 'page', 'pages', 'paid',
'pail', 'pain', 'painful', 'pains', 'paint', 'paintbrushes', 'painted',
'painters', 'painting', 'paintings', 'paints', 'pair', 'paired', 'pairing',
'pairs', 'pakistan', 'pal', 'palacio', 'palm', 'pals', 'palsy', 'panel',
'panels', 'pans', 'pantry', 'pants', 'paper', 'paperback', 'paperless',
'papers', 'paperwork', 'par', 'para', 'parachute', 'parachutes', 'parade',
'parades', 'paragraph', 'paragraphs', 'parallel', 'paramount',
'paraprofessional', 'paraprofessionals', 'parcc', 'parent', 'parental',
'parenting', 'parents', 'parish', 'park', 'parking', 'parks', 'part', 'partake',
'partial', 'partially', 'participant', 'participants', 'participate',
'participated', 'participates', 'participating', 'participation', 'particular',
'particularly', 'parties', 'partitions', 'partner', 'partnered', 'partnering',
'partners', 'partnership', 'partnerships', 'parts', 'party', 'pass', 'passage',
'passages', 'passed', 'passes', 'passing', 'passion', 'passionate',
'passionately', 'passions', 'passive', 'passively', 'past', 'paste', 'pastel',
'pastels', 'pasting', 'path', 'pathologist', 'paths', 'pathway', 'pathways',
'patience', 'patient', 'patiently', 'pattern', 'patterning', 'patterns', 'paul',
'pause', 'pave', 'paws', 'pay', 'paycheck', 'paying', 'pays', 'pbis', 'pbl',
'pbs', 'pc', 'pcs', 'pdf', 'pe', 'peace', 'peaceful', 'peak', 'peanut', 'pearl',
'pecs', 'pedagogy', 'pedal', 'pedaling', 'pedals', 'peddlers', 'pedometer',
'pedometers', 'peek', 'peer', 'peers', 'pegs', 'pellet', 'pellets', 'pen',
'pencil', 'pencils', 'penguins', 'peninsula', 'penmanship', 'pennies',
'pennsylvania', 'penny', 'pens', 'pent', 'people', 'peoples', 'pep', 'per',
'perceive', 'perceived', 'percent', 'percentage', 'percentages', 'percentile',
'perception', 'perceptions', 'percussion', 'percussionists', 'percy', 'perfect',
'perfection', 'perfectly', 'perform', 'performance', 'performances',
'performed', 'performers', 'performing', 'performs', 'perhaps', 'perimeter',
'period', 'periodic', 'periodically', 'periodicals', 'periods', 'perks',
'permanent', 'permanently', 'permission', 'permit', 'perseverance', 'persevere',
'persevering', 'persist', 'persistence', 'persistent', 'person', 'personal',
'personalities', 'personality', 'personalize', 'personalized', 'personalizing',
'personally', 'personnel', 'persons', 'perspective', 'perspectives', 'persuade',
'persuasive', 'pertain', 'pertaining', 'pertinent', 'peru', 'peruse', 'pet',
'pete', 'pets', 'ph', 'phase', 'phases', 'phenomena', 'phenomenal',
'phenomenon', 'philadelphia', 'philippines', 'philosophy', 'phoenix', 'phone',
'phonemic', 'phones', 'phonetic', 'phonic', 'phonics', 'phonological', 'photo',
'photograph', 'photographer', 'photographers', 'photographic', 'photographs',
'photography', 'photos', 'photoshop', 'photosynthesis', 'phrase', 'phrases',
'physical', 'physically', 'physics', 'physiology', 'pi', 'piano', 'pianos',
'picasso', 'pick', 'picked', 'picking', 'picks', 'picnic', 'picture',
'pictures', 'pie', 'piece', 'pieces', 'pig', 'pigeon', 'piggie', 'piggy',
'pigs', 'pile', 'piles', 'pillar', 'pillars', 'pillow', 'pillows', 'pilot',
'piloting', 'pin', 'pinch', 'ping', 'pink', 'pinnies', 'pins', 'pioneer',
'pioneers', 'pipe', 'pique', 'pirates', 'pit', 'pitch', 'pitched', 'pitches',
'pitching', 'pittsburgh', 'pivotal', 'pizza', 'pk', 'place', 'placed',
'placement', 'places', 'placing', 'plague', 'plagued', 'plain', 'plan', 'plane',
'planes', 'planet', 'planets', 'planks', 'planned', 'planner', 'planners',
'planning', 'plans', 'plant', 'planted', 'planters', 'planting', 'plants',
'plaster', 'plastic', 'plate', 'plates', 'platform', 'platforms', 'play',
'playback', 'playdoh', 'playdough', 'played', 'player', 'players', 'playful',
'playground', 'playgrounds', 'playing', 'plays', 'playtime', 'pleasant',
'please', 'pleased', 'pleasing', 'pleasurable', 'pleasure', 'pledge',
'plentiful', 'plenty', 'plethora', 'plickers', 'plot', 'plots', 'pltw', 'plug',
'plugged', 'plugs', 'plus', 'plush', 'pm', 'pocket', 'pockets', 'pod',
'podcast', 'podcasts', 'podium', 'pods', 'poem', 'poems', 'poetry', 'poets',
'point', 'pointed', 'pointers', 'pointing', 'points', 'pokemon', 'polar',
'polaroid', 'pole', 'poles', 'police', 'policy', 'polish', 'polished', 'polite',
'political', 'politicians', 'politics', 'poll', 'pollution', 'pom', 'poms',
'pond', 'pong', 'pool', 'poor', 'poorer', 'poorest', 'poorly', 'pop', 'popcorn',
'popped', 'popping', 'pops', 'popsicle', 'popular', 'popularity', 'populated',
'population', 'populations', 'port', 'portability', 'portable', 'portfolio',
'portfolios', 'portion', 'portions', 'portland', 'portrait', 'portraits',
'portray', 'portrayed', 'portuguese', 'pose', 'posed', 'poses', 'position',
'positions', 'positive', 'positively', 'positives', 'positivity', 'possess',
'possesses', 'possessions', 'possibilities', 'possibility', 'possible',
'possibly', 'post', 'posted', 'poster', 'posters', 'posting', 'posts',
'posture', 'postures', 'pot', 'potato', 'potential', 'potentially',
'potentials', 'pots', 'potter', 'pottery', 'pouch', 'pouches', 'pound', 'pour',
'pouring', 'poverty', 'power', 'powered', 'powerful', 'powerpoint',
'powerpoints', 'powers', 'practical', 'practically', 'practice', 'practiced',
'practices', 'practicing', 'praise', 'praised', 'pray', 'praying', 'pre',
'precious', 'precise', 'precisely', 'precision', 'precocious', 'predict',
'predictable', 'predicting', 'prediction', 'predictions', 'predictor',
'predominantly', 'predominately', 'prefer', 'preference', 'preferences',
'preferred', 'preform', 'preforming', 'pregnancy', 'prek', 'prekindergarten',
'premium', 'prep', 'preparation', 'preparatory', 'prepare', 'prepared',
'preparedness', 'prepares', 'preparing', 'prepping', 'prerequisite',
'preschool', 'preschoolers', 'presence', 'present', 'presentation',
'presentations', 'presented', 'presenter', 'presenters', 'presenting',
'presently', 'presents', 'preserve', 'preserved', 'preserving', 'president',
'presidential', 'presidents', 'press', 'pressing', 'pressure', 'pressures',
'prestigious', 'preteens', 'pretend', 'pretending', 'pretty', 'pretzels',
'prevalent', 'prevent', 'prevented', 'preventing', 'prevention', 'prevents',
'previous', 'previously', 'prey', 'prezi', 'price', 'priced', 'priceless',
'prices', 'pricey', 'pride', 'prides', 'primarily', 'primary', 'prime',
'principal', 'principals', 'principle', 'principles', 'print', 'printable',
'printed', 'printer', 'printers', 'printing', 'printmaking', 'prints', 'prior',
'priorities', 'prioritize', 'priority', 'prison', 'privacy', 'private',
'privately', 'privilege', 'privileged', 'privileges', 'prize', 'prizes', 'pro',
'proactive', 'probability', 'probably', 'probes', 'problem', 'problematic',
'problems', 'procedure', 'procedures', 'proceed', 'process', 'processed',
'processes', 'processing', 'processor', 'procure', 'prodigy', 'produce',
'produced', 'producers', 'produces', 'producing', 'product', 'production',
'productions', 'productive', 'productively', 'productivity', 'products',
'profession', 'professional', 'professionalism', 'professionally',
'professionals', 'professions', 'professor', 'professors', 'proficiency',
'proficient', 'proficiently', 'profile', 'profiles', 'profit', 'profits',
'profound', 'profoundly', 'program', 'programing', 'programmable', 'programmed',
'programmers', 'programming', 'programs', 'progress', 'progressed',
'progresses', 'progressing', 'progression', 'progressive', 'progressively',
'project', 'projected', 'projecting', 'projection', 'projector', 'projectors',
'projects', 'prolonged', 'promethean', 'prominent', 'promise', 'promised',
'promising', 'promote', 'promoted', 'promotes', 'promoting', 'promotion',
'prompt', 'prompted', 'prompting', 'prompts', 'prone', 'pronounce',
'pronounced', 'pronunciation', 'proof', 'prop', 'propel', 'proper', 'properly',
'properties', 'property', 'proportion', 'proportions', 'proposal', 'proposed',
'props', 'pros', 'prospect', 'prospects', 'prosper', 'protagonist',
'protagonists', 'protect', 'protected', 'protecting', 'protection',
'protective', 'protector', 'protectors', 'protein', 'prototype', 'prototypes',
'prototyping', 'protractors', 'proud', 'prouder', 'proudly', 'prove', 'proved',
'proven', 'proverb', 'proves', 'provide', 'provided', 'providence', 'provider',
'providers', 'provides', 'providing', 'proving', 'provoking', 'proximity', 'ps',
'psychological', 'psychology', 'pta', 'pto', 'puberty', 'public', 'publication',
'publications', 'publicly', 'publish', 'published', 'publishers', 'publishing',
'puerto', 'pull', 'pulled', 'pulling', 'pullout', 'pulls', 'pulse', 'pump',
'pumped', 'pumping', 'pumpkin', 'pumpkins', 'punch', 'punctuation', 'pupil',
'pupils', 'puppet', 'puppets', 'purchase', 'purchased', 'purchases',
'purchasing', 'pure', 'purely', 'purifier', 'purple', 'purpose', 'purposeful',
'purposefully', 'purposes', 'pursue', 'pursuing', 'pursuit', 'pursuits', 'push',
'pushed', 'pushes', 'pushing', 'put', 'puts', 'putting', 'putty', 'puzzle',
'puzzles', 'pyramid', 'pyramids', 'python', 'qr', 'quaint', 'qualified',
'qualifies', 'qualify', 'qualifying', 'qualities', 'quality', 'quantities',
'quantity', 'quarter', 'quarterly', 'quarters', 'queens', 'quench', 'quest',
'question', 'questioners', 'questioning', 'questions', 'quests', 'quick',
'quicker', 'quickly', 'quiet', 'quieter', 'quietly', 'quilt', 'quirky', 'quit',
'quite', 'quiz', 'quizlet', 'quizzes', 'quote', 'quotes', 'race', 'races',
'racial', 'racially', 'racing', 'racism', 'rack', 'rackets', 'racks', 'radio',
'radius', 'raffle', 'rage', 'railroad', 'rain', 'rainbow', 'rainforest',
'raining', 'rains', 'rainy', 'raise', 'raised', 'raises', 'raising', 'rallies',
'rally', 'ralph', 'rambunctious', 'ramp', 'ramps', 'ran', 'ranch', 'random',
'range', 'ranges', 'ranging', 'rank', 'ranked', 'ranking', 'ranks', 'rap',
'rapid', 'rapidly', 'rapport', 'raps', 'rare', 'rarely', 'raspberry', 'rate',
'rated', 'rates', 'rather', 'rating', 'ratings', 'ratio', 'ratios', 'raw',
'ray', 'rays', 'raz', 'razkids', 'reach', 'reached', 'reaches', 'reaching',
'react', 'reaction', 'reactions', 'read', 'readability', 'readable', 'reader',
'readers', 'readily', 'readiness', 'reading', 'readings', 'reads', 'ready',
'real', 'realia', 'realistic', 'realities', 'reality', 'realization', 'realize',
'realized', 'realizing', 'really', 'realm', 'realms', 'reams', 'reap',
'rearrange', 'reason', 'reasonable', 'reasoning', 'reasons', 'reassurance',
'rebuild', 'rebuilding', 'rebuilt', 'recall', 'receive', 'received', 'receives',
'receiving', 'recent', 'recently', 'receptive', 'recess', 'recesses',
'recharge', 'rechargeable', 'recieve', 'recipe', 'recipes', 'recipient',
'recipients', 'recite', 'recognition', 'recognize', 'recognized', 'recognizes',
'recognizing', 'recommend', 'recommendation', 'recommendations', 'recommended',
'record', 'recorded', 'recorder', 'recorders', 'recording', 'recordings',
'records', 'recovery', 'recreate', 'recreating', 'recreation', 'recreational',
'rectangle', 'rectangular', 'recyclable', 'recycle', 'recycled', 'recycling',
'red', 'redesign', 'redesigning', 'redirect', 'redirecting', 'redirection',
'reduce', 'reduced', 'reduces', 'reducing', 'reduction', 'reed', 'reeds',
'reef', 'reenact', 'refer', 'reference', 'references', 'referrals', 'referred',
'referring', 'refill', 'refills', 'refine', 'refining', 'reflect', 'reflected',
'reflecting', 'reflection', 'reflections', 'reflective', 'reflects', 'reflex',
'refocus', 'refresh', 'refreshed', 'refreshing', 'refrigerator', 'refuel',
'refuge', 'refugee', 'refugees', 'refurbished', 'refuse', 'regain', 'regard',
'regarding', 'regardless', 'regards', 'regents', 'reggio', 'region', 'regional',
'regions', 'register', 'registered', 'regroup', 'regrouping', 'regular',
'regularly', 'regulate', 'regulated', 'regulating', 'regulation', 'regulations',
'rehearsal', 'rehearsals', 'rehearse', 'rehearsing', 'reinforce', 'reinforced',
'reinforcement', 'reinforcements', 'reinforcers', 'reinforces', 'reinforcing',
'rekenrek', 'rekenreks', 'relatable', 'relate', 'related', 'relates',
'relating', 'relation', 'relational', 'relations', 'relationship',
'relationships', 'relative', 'relatively', 'relatives', 'relax', 'relaxation',
'relaxed', 'relaxing', 'relay', 'relays', 'release', 'released', 'releases',
'releasing', 'relentless', 'relentlessly', 'relevance', 'relevant', 'reliable',
'reliance', 'reliant', 'relief', 'relies', 'relieve', 'religion', 'religions',
'religious', 'relish', 'relocate', 'relocated', 'reluctant', 'rely', 'relying',
'remain', 'remainder', 'remained', 'remaining', 'remains', 'remarkable',
'remarkably', 'remedial', 'remediate', 'remediation', 'remedy', 'remember',
'remembered', 'remembering', 'remind', 'reminded', 'reminder', 'reminders',
'reminding', 'reminds', 'remote', 'removable', 'remove', 'removed', 'removes',
'removing', 'renaissance', 'renewable', 'renewed', 'renovated', 'rent',
'rental', 'repair', 'repaired', 'repairs', 'repeat', 'repeated', 'repeatedly',
'repeating', 'repertoire', 'repetition', 'repetitive', 'replace', 'replaced',
'replacement', 'replacements', 'replacing', 'replenish', 'replenished',
'replenishing', 'replica', 'replicas', 'replicate', 'replied', 'reply',
'report', 'reported', 'reporters', 'reporting', 'reports', 'represent',
'representation', 'representations', 'representative', 'represented',
'representing', 'represents', 'reproduce', 'reproducible', 'republic',
'reputation', 'request', 'requested', 'requesting', 'requests', 'require',
'required', 'requirement', 'requirements', 'requires', 'requiring', 'requisite',
'reread', 'rereading', 'rescue', 'research', 'researched', 'researchers',
'researching', 'reservation', 'reserve', 'reserved', 'reset', 'reside',
'residential', 'residents', 'resides', 'residing', 'resilience', 'resiliency',
'resilient', 'resist', 'resistance', 'resistant', 'resolution', 'resolve',
'resort', 'resource', 'resourced', 'resourceful', 'resourcefulness',
'resources', 'respect', 'respectable', 'respected', 'respectful',
'respectfully', 'respecting', 'respective', 'respects', 'respond', 'responded',
'responding', 'responds', 'response', 'responses', 'responsibilities',
'responsibility', 'responsible', 'responsibly', 'responsive', 'rest',
'restaurant', 'restaurants', 'restless', 'restlessness', 'restock',
'restorative', 'restraints', 'restricted', 'restrictions', 'restrictive',
'restroom', 'rests', 'result', 'resulted', 'resulting', 'results', 'resume',
'resumes', 'retail', 'retain', 'retained', 'retaining', 'reteach', 'reteaching',
'retell', 'retelling', 'retention', 'retired', 'retreat', 'retrieve', 'return',
'returned', 'returning', 'reusable', 'reuse', 'reused', 'reusing', 'reveal',
'revealed', 'reverse', 'review', 'reviewed', 'reviewing', 'reviews', 'revise',
'revising', 'revision', 'revisions', 'revisit', 'revolution', 'revolutionary',
'revolutionize', 'revolve', 'revolves', 'revolving', 'reward', 'rewarded',
'rewarding', 'rewards', 'rewrite', 'rhode', 'rhyme', 'rhymes', 'rhyming',
'rhythm', 'rhythmic', 'rhythms', 'ribbon', 'ribbons', 'rice', 'rich', 'richer',
'richest', 'richmond', 'richness', 'rico', 'rid', 'ridden', 'riddled',
'riddles', 'ride', 'rides', 'ridge', 'riding', 'right', 'rights', 'rigid',
'rigor', 'rigorous', 'rigors', 'ring', 'rings', 'rio', 'rip', 'ripped', 'rise',
'risen', 'risers', 'rises', 'rising', 'risk', 'risks', 'river', 'rivers',
'road', 'roadblocks', 'roads', 'roald', 'roam', 'rob', 'robert', 'robinson',
'robot', 'robotic', 'robotics', 'robots', 'robust', 'rochester', 'rock',
'rocker', 'rockers', 'rocket', 'rockets', 'rocking', 'rocks', 'rods', 'rogers',
'role', 'roles', 'roll', 'rolled', 'roller', 'rollers', 'rolling', 'rolls',
'romeo', 'roof', 'room', 'rooms', 'roosevelt', 'root', 'rooted', 'roots',
'rope', 'ropes', 'roping', 'rose', 'roster', 'rotate', 'rotated', 'rotating',
'rotation', 'rotations', 'rote', 'rouge', 'rough', 'roughly', 'round',
'rounded', 'rounds', 'route', 'routes', 'routine', 'routinely', 'routines',
'row', 'rowling', 'rows', 'rti', 'rubber', 'rube', 'rubik', 'rubric', 'rubrics',
'rug', 'rugs', 'ruined', 'rule', 'ruler', 'rulers', 'rules', 'run', 'runner',
'runners', 'running', 'runny', 'runs', 'runtz', 'rural', 'rush', 'rushing',
'russia', 'russian', 'sack', 'sacks', 'sacramento', 'sacrifice', 'sacrifices',
'sad', 'sadly', 'sadness', 'safe', 'safely', 'safer', 'safest', 'safety',
'said', 'sail', 'sake', 'salad', 'salary', 'sale', 'sales', 'salmon', 'salt',
'salvador', 'sample', 'samples', 'samsung', 'san', 'sanctuary', 'sand', 'sandy',
'sanitary', 'sanitize', 'sanitizer', 'sanitizing', 'santa', 'sat',
'satisfaction', 'satisfied', 'satisfy', 'satisfying', 'saturday', 'saucer',
'save', 'saved', 'saver', 'saves', 'saving', 'savings', 'savvy', 'saw',
'saxophone', 'saxophones', 'say', 'saying', 'says', 'sbac', 'sc', 'scaffold',
'scaffolded', 'scaffolding', 'scaffolds', 'scale', 'scales', 'scan', 'scanner',
'scanning', 'scarce', 'scare', 'scared', 'scarves', 'scary', 'scattered',
'scavenger', 'scenario', 'scenarios', 'scene', 'scenery', 'scenes', 'scented',
'schedule', 'scheduled', 'schedules', 'scheduling', 'schema', 'scholar',
'scholarly', 'scholars', 'scholarship', 'scholarships', 'scholastic',
'schoolers', 'schooling', 'schools', 'schoolwork', 'science', 'sciences',
'scientific', 'scientifically', 'scientist', 'scientists', 'scissors', 'scoop',
'scooter', 'scooters', 'scope', 'score', 'scored', 'scores', 'scoring',
'scotch', 'scott', 'scrambling', 'scrapbook', 'scraps', 'scratch', 'scratched',
'screaming', 'screen', 'screens', 'screws', 'script', 'scripts', 'sculpt',
'sculpting', 'sculpture', 'sculptures', 'sdc', 'se', 'sea', 'seal', 'seamless',
'seamlessly', 'seams', 'search', 'searched', 'searching', 'season', 'seasonal',
'seasons', 'seat', 'seated', 'seating', 'seats', 'seattle', 'second',
'secondary', 'secondly', 'seconds', 'secret', 'section', 'sections', 'secure',
'secured', 'securely', 'securing', 'security', 'sedentary', 'see', 'seed',
'seedlings', 'seeds', 'seeing', 'seek', 'seekers', 'seeking', 'seeks', 'seem',
'seemed', 'seemingly', 'seems', 'seen', 'sees', 'seesaw', 'segment', 'segments',
'segregated', 'sel', 'seldom', 'select', 'selected', 'selecting', 'selection',
'selections', 'selective', 'self', 'selfless', 'sell', 'selling', 'selves',
'semester', 'semi', 'seminar', 'seminars', 'send', 'sending', 'sends', 'senior',
'seniors', 'sensation', 'sense', 'senses', 'sensitive', 'sensitivity', 'sensor',
'sensors', 'sensory', 'sent', 'sentence', 'sentences', 'separate', 'separated',
'separation', 'september', 'sequence', 'sequences', 'sequencing', 'sequential',
'series', 'serious', 'seriously', 'serve', 'served', 'serves', 'service',
'serviced', 'services', 'servicing', 'serving', 'ses', 'session', 'sessions',
'set', 'setbacks', 'sets', 'setting', 'settings', 'settle', 'settled',
'settling', 'setup', 'seuss', 'seven', 'seventeen', 'seventh', 'seventy',
'several', 'severe', 'severely', 'severity', 'sew', 'sewing', 'sexual', 'shade',
'shades', 'shadow', 'shadows', 'shake', 'shakers', 'shakespeare', 'shaking',
'shall', 'shame', 'shampoo', 'shape', 'shaped', 'shapes', 'shaping', 'share',
'shared', 'shares', 'sharing', 'shark', 'sharks', 'sharp', 'sharpen',
'sharpened', 'sharpener', 'sharpeners', 'sharpening', 'sharpie', 'sharpies',
'she', 'shed', 'sheer', 'sheet', 'sheets', 'shelf', 'shell', 'shells',
'shelter', 'sheltered', 'shelters', 'shelves', 'shelving', 'shields', 'shift',
'shifted', 'shifting', 'shifts', 'shin', 'shine', 'shines', 'shining', 'shiny',
'ship', 'shirt', 'shirts', 'shock', 'shocked', 'shoe', 'shoes', 'shoot',
'shooting', 'shop', 'shopping', 'shops', 'shore', 'short', 'shortage',
'shortages', 'shortcomings', 'shorter', 'shorts', 'shot', 'shots', 'should',
'shoulder', 'shoulders', 'shout', 'shovels', 'show', 'showcase', 'showcased',
'showcases', 'showcasing', 'showed', 'shower', 'showing', 'shown', 'shows',
'shredder', 'shrinking', 'shuffles', 'shut', 'shy', 'sibling', 'siblings',
'sick', 'side', 'sided', 'sides', 'sidewalk', 'sierra', 'sight', 'sights',
'sign', 'signal', 'signed', 'significance', 'significant', 'significantly',
'signing', 'signs', 'silence', 'silent', 'silently', 'silhouette', 'silicon',
'silly', 'silver', 'similar', 'similarities', 'similarly', 'simple', 'simpler',
'simplest', 'simply', 'simulate', 'simulated', 'simulation', 'simulations',
'simultaneously', 'since', 'sincere', 'sincerely', 'sing', 'singers', 'singing',
'single', 'singled', 'sink', 'sir', 'sister', 'sisters', 'sit', 'site', 'sites',
'sits', 'sitting', 'situated', 'situation', 'situations', 'six', 'sixteen',
'sixth', 'sixty', 'size', 'sized', 'sizes', 'skeletal', 'skeleton', 'sketch',
'sketchbook', 'sketchbooks', 'sketches', 'sketching', 'skill', 'skilled',
'skills', 'skin', 'skip', 'skipping', 'skits', 'sky', 'skype', 'skyrocket',
'slam', 'slashed', 'slate', 'slates', 'slavery', 'sleep', 'sleeping', 'sleepy',
'sleeves', 'slide', 'slides', 'slideshow', 'slideshows', 'sliding', 'slight',
'slightly', 'slim', 'slip', 'slipping', 'slips', 'slogan', 'slot', 'slouching',
'slow', 'slower', 'slowing', 'slowly', 'slows', 'small', 'smaller', 'smallest',
'smart', 'smartboard', 'smartboards', 'smarter', 'smartest', 'smarties',
'smartphone', 'smartphones', 'smell', 'smelling', 'smells', 'smelly', 'smile',
'smiles', 'smiling', 'smith', 'smocks', 'smooth', 'smoother', 'smoothie',
'smoothies', 'smoothly', 'snack', 'snacking', 'snacks', 'snap', 'snapshot',
'snare', 'sneak', 'sneakers', 'snow', 'snowy', 'snuggle', 'so', 'soak',
'soaking', 'soap', 'soar', 'soaring', 'soars', 'soccer', 'social',
'socialization', 'socialize', 'socializing', 'socially', 'societal', 'society',
'socio', 'socioeconomic', 'socioeconomically', 'socks', 'socratic', 'soda',
'sofa', 'soft', 'softball', 'softer', 'software', 'soil', 'solace', 'solar',
'sold', 'soldering', 'sole', 'solely', 'solid', 'solidify', 'solids', 'solo',
'solution', 'solutions', 'solve', 'solved', 'solver', 'solvers', 'solving',
'somali', 'somalia', 'some', 'somebody', 'someday', 'somehow', 'someone',
'someplace', 'something', 'sometime', 'sometimes', 'somewhat', 'somewhere',
'son', 'song', 'songs', 'soon', 'sooner', 'soothe', 'soothing', 'sophisticated',
'sophomore', 'sophomores', 'soprano', 'sore', 'sorely', 'soreness', 'sorry',
'sort', 'sorted', 'sorting', 'sorts', 'sought', 'soul', 'souls', 'sound',
'sounding', 'sounds', 'soup', 'source', 'sources', 'south', 'southeast',
'southeastern', 'southern', 'southside', 'southwest', 'space', 'spaces',
'spacial', 'spacious', 'spain', 'span', 'spanish', 'spanning', 'spans', 'spare',
'spark', 'sparked', 'sparking', 'sparkle', 'sparks', 'sparse', 'spatial',
'speak', 'speaker', 'speakers', 'speaking', 'speaks', 'special', 'specialist',
'specialists', 'specialize', 'specialized', 'specializes', 'specially',
'specials', 'specialty', 'species', 'specific', 'specifically', 'specified',
'specimen', 'specimens', 'spectacular', 'spectrum', 'sped', 'speech',
'speeches', 'speed', 'speeds', 'spell', 'spelling', 'spend', 'spending',
'spends', 'spent', 'sphero', 'spheros', 'spice', 'spider', 'spill', 'spills',
'spin', 'spine', 'spinning', 'spiral', 'spirit', 'spirited', 'spirits', 'spite',
'splash', 'split', 'splitter', 'splitters', 'spoke', 'spoken', 'sponge',
'sponges', 'sponsor', 'sponsored', 'sponsors', 'spontaneous', 'spoon', 'spoons',
'sport', 'sporting', 'sports', 'sportsmanship', 'spot', 'spotlight', 'spots',
'sprawl', 'sprawled', 'spray', 'spread', 'spreading', 'spreads', 'spreadsheet',
'spreadsheets', 'spring', 'springboard', 'springfield', 'springs', 'sprk',
'sprout', 'spunk', 'spunky', 'spur', 'squad', 'square', 'squares', 'squeeze',
'squirm', 'squirming', 'squirmy', 'squished', 'squishy', 'st', 'staar',
'stability', 'stable', 'stack', 'stackable', 'stacked', 'stacking', 'stacks',
'staff', 'staffed', 'staffing', 'stage', 'stages', 'stagnant', 'stained',
'stains', 'stairs', 'stakeholders', 'stakes', 'stamina', 'stamp', 'stamping',
'stamps', 'stand', 'standard', 'standardized', 'standards', 'standing',
'stands', 'stanley', 'staple', 'stapler', 'staplers', 'staples', 'star',
'starbucks', 'starfall', 'staring', 'stars', 'start', 'started', 'starter',
'starters', 'starting', 'starts', 'starving', 'state', 'stated', 'statement',
'statements', 'staten', 'states', 'statewide', 'static', 'stating', 'station',
'stationary', 'stationed', 'stations', 'statistic', 'statistical',
'statistically', 'statistics', 'stats', 'status', 'statuses', 'stay', 'stayed',
'staying', 'stays', 'steadily', 'steady', 'steam', 'steel', 'steer', 'stellar',
'stem', 'stemming', 'stems', 'stencils', 'step', 'stephen', 'stepped',
'stepping', 'steps', 'stereo', 'stereotypes', 'steve', 'stewards',
'stewardship', 'stick', 'sticker', 'stickers', 'sticking', 'sticks', 'sticky',
'stiff', 'stifle', 'stifled', 'stigma', 'still', 'stimulate', 'stimulated',
'stimulates', 'stimulating', 'stimulation', 'stimuli', 'stimulus', 'stir',
'stix', 'stock', 'stocked', 'stocking', 'stolen', 'stomach', 'stomachs',
'stone', 'stones', 'stood', 'stool', 'stools', 'stop', 'stopped', 'stopping',
'stops', 'stopwatches', 'storage', 'store', 'stored', 'stores', 'stories',
'storing', 'storm', 'story', 'storyboards', 'storybook', 'storybooks',
'storytellers', 'storytelling', 'storyworks', 'stove', 'straight', 'strain',
'strange', 'strangers', 'strap', 'strapped', 'straps', 'strategic',
'strategically', 'strategies', 'strategy', 'straws', 'stream', 'streaming',
'streamline', 'street', 'streets', 'strength', 'strengthen', 'strengthened',
'strengthening', 'strengthens', 'strengths', 'stress', 'stressed', 'stresses',
'stressful', 'stressors', 'stretch', 'stretched', 'stretches', 'stretching',
'stricken', 'strict', 'strictly', 'stride', 'strides', 'strikes', 'striking',
'string', 'stringed', 'strings', 'strip', 'strips', 'strive', 'strives',
'striving', 'strong', 'stronger', 'strongest', 'strongly', 'struck',
'structural', 'structure', 'structured', 'structures', 'struggle', 'struggled',
'struggles', 'struggling', 'stuck', 'student', 'studied', 'studies', 'studio',
'studnets', 'study', 'studying', 'stuff', 'stuffed', 'stunning', 'stupid',
'sturdy', 'stuttering', 'style', 'styles', 'stylus', 'sub', 'subgroups',
'subject', 'subjects', 'submit', 'submitted', 'submitting', 'subscription',
'subscriptions', 'subsequent', 'subsequently', 'subsidized', 'substance',
'substances', 'substantial', 'substantially', 'substitute', 'subtle',
'subtract', 'subtracting', 'subtraction', 'suburb', 'suburban', 'suburbs',
'succeed', 'succeeded', 'succeeding', 'succeeds', 'success', 'successes',
'successful', 'successfully', 'such', 'sudan', 'sudden', 'suddenly', 'suffer',
'suffered', 'suffering', 'suffers', 'sufficient', 'sufficiently', 'sugar',
'sugary', 'suggest', 'suggested', 'suggestion', 'suggestions', 'suggests',
'suit', 'suitable', 'suite', 'suited', 'suits', 'sum', 'sumdog', 'summaries',
'summarize', 'summarizing', 'summary', 'summative', 'summer', 'summit', 'sums',
'sun', 'sunlight', 'sunny', 'sunshine', 'super', 'superhero', 'superheroes',
'superintendent', 'superior', 'superstars', 'supervised', 'supervision',
'supper', 'supplement', 'supplemental', 'supplementary', 'supplemented',
'supplements', 'supplied', 'supplies', 'supply', 'supplying', 'support',
'supported', 'supporter', 'supporters', 'supporting', 'supportive', 'supports',
'suppose', 'supposed', 'supreme', 'sure', 'surely', 'surf', 'surface',
'surfaces', 'surge', 'surpass', 'surprise', 'surprised', 'surprises',
'surprising', 'surprisingly', 'surround', 'surrounded', 'surrounding',
'surroundings', 'surrounds', 'survey', 'surveyed', 'surveys', 'survival',
'survive', 'survived', 'survivors', 'sustain', 'sustainability', 'sustainable',
'sustained', 'sustaining', 'swahili', 'swap', 'sway', 'sweat', 'sweeping',
'sweet', 'sweetest', 'swim', 'swimming', 'swing', 'swinging', 'swings',
'switch', 'switched', 'switches', 'switching', 'swivel', 'syllables', 'symbol',
'symbolism', 'symbols', 'symmetry', 'symphony', 'symptoms', 'syndrome',
'synonyms', 'synthesis', 'synthesize', 'synthesizing', 'syria', 'system',
'systematic', 'systems', 'tab', 'table', 'tables', 'tablet', 'tabletop',
'tablets', 'tabs', 'tackle', 'tackling', 'tactile', 'tag', 'tagalog', 'tags',
'tailor', 'tailored', 'take', 'taken', 'takers', 'takes', 'taking', 'tale',
'talent', 'talented', 'talents', 'tales', 'talk', 'talkative', 'talked',
'talkers', 'talking', 'talks', 'tall', 'taller', 'tampa', 'tangible', 'tangle',
'tangled', 'tangram', 'tangrams', 'tank', 'tap', 'tape', 'taped', 'tapes',
'taping', 'tapping', 'target', 'targeted', 'targeting', 'targets', 'task',
'tasked', 'tasks', 'taste', 'tastes', 'tasty', 'tattered', 'taught', 'tax',
'tea', 'teach', 'teachable', 'teacher', 'teachers', 'teaches', 'teaching',
'teachings', 'team', 'teammate', 'teammates', 'teams', 'teamwork', 'tear',
'tearing', 'tears', 'tech', 'technical', 'technically', 'technicians',
'technique', 'techniques', 'technological', 'technologically', 'technologies',
'technology', 'ted', 'tedious', 'tee', 'teen', 'teenage', 'teenager',
'teenagers', 'teens', 'teeth', 'television', 'tell', 'telling', 'tells',
'tempera', 'temperature', 'temperatures', 'templates', 'temple', 'tempo',
'temporarily', 'temporary', 'ten', 'tenacious', 'tenacity', 'tend',
'tendencies', 'tendency', 'tender', 'tends', 'tenmarks', 'tennessee', 'tennis',
'tens', 'tension', 'tent', 'tenth', 'term', 'terminology', 'terms', 'terrarium',
'terrible', 'terribly', 'terrific', 'territory', 'test', 'testament', 'tested',
'testing', 'tests', 'texas', 'text', 'textbook', 'textbooks', 'texting',
'texts', 'textual', 'texture', 'textured', 'textures', 'tfk', 'th', 'thailand',
'thank', 'thankful', 'thankfully', 'thanks', 'thanksgiving', 'that', 'thats',
'theater', 'theaters', 'theatre', 'theatrical', 'their', 'them', 'thematic',
'theme', 'themed', 'themes', 'then', 'theoretical', 'theories', 'theory',
'therapeutic', 'therapies', 'therapist', 'therapists', 'theraputty', 'therapy',
'there', 'thereby', 'therefor', 'therefore', 'thermal', 'thermometer',
'thermometers', 'thesaurus', 'thesauruses', 'these', 'theses', 'thick', 'thief',
'thier', 'thin', 'thing', 'things', 'think', 'thinker', 'thinkers', 'thinking',
'thinks', 'third', 'thirds', 'thirst', 'thirsting', 'thirsty', 'thirteen',
'thirty', 'thomas', 'thorough', 'thoroughly', 'those', 'though', 'thought',
'thoughtful', 'thoughtfully', 'thoughtfulness', 'thoughts', 'thousand',
'thousands', 'thread', 'threatening', 'three', 'thrift', 'thrill', 'thrilled',
'thrilling', 'thrive', 'thrives', 'thriving', 'through', 'throughout', 'throw',
'throwing', 'thrown', 'throws', 'thru', 'thrust', 'thursday', 'thus', 'ti',
'ticket', 'tickets', 'tidy', 'tie', 'tied', 'tier', 'ties', 'tiger', 'tigers',
'tiggly', 'tight', 'tightly', 'tile', 'tiles', 'till', 'tilt', 'time', 'timed',
'timeless', 'timeline', 'timely', 'timer', 'timers', 'times', 'timing', 'tin',
'tinker', 'tinkercad', 'tinkering', 'tiny', 'tip', 'tipping', 'tips', 'tire',
'tired', 'tirelessly', 'tissue', 'tissues', 'title', 'titled', 'titles', 'tk',
'tlc', 'tn', 'to', 'today', 'todays', 'toe', 'toes', 'together', 'toilet',
'token', 'tokens', 'told', 'tolerance', 'tolerant', 'toll', 'tom', 'tomatoes',
'tomorrow', 'tomorrows', 'ton', 'tone', 'toner', 'tones', 'tongue', 'tons',
'too', 'took', 'tool', 'toolbox', 'tools', 'toon', 'tooth', 'toothbrush',
'toothbrushes', 'toothpaste', 'top', 'topic', 'topics', 'tops', 'torn',
'tornado', 'toss', 'tossing', 'total', 'totally', 'tote', 'totes', 'touch',
'touched', 'touches', 'touching', 'touchscreen', 'tough', 'tougher', 'toughest',
'tour', 'tourism', 'tournament', 'tournaments', 'tours', 'toward', 'towards',
'towel', 'towels', 'tower', 'towers', 'town', 'towns', 'township', 'toxic',
'toy', 'toys', 'trace', 'tracing', 'track', 'tracked', 'tracker', 'trackers',
'tracking', 'tracks', 'trade', 'trades', 'trading', 'tradition', 'traditional',
'traditionally', 'traditions', 'traffic', 'tragedies', 'tragedy', 'tragic',
'trail', 'trailer', 'trailers', 'trails', 'train', 'trained', 'trainer',
'trainers', 'training', 'trainings', 'trains', 'trait', 'traits', 'trajectory',
'trampoline', 'transcend', 'transfer', 'transferable', 'transferred',
'transferring', 'transfers', 'transform', 'transformation', 'transformations',
'transformative', 'transformed', 'transforming', 'transforms', 'transient',
'transition', 'transitional', 'transitioned', 'transitioning', 'transitions',
'translate', 'translated', 'translates', 'translation', 'translations',
'translators', 'translucent', 'transparent', 'transplant', 'transport',
'transportation', 'transported', 'transporting', 'trapped', 'trash', 'trauma',
'traumatic', 'travel', 'traveled', 'travelers', 'traveling', 'travels', 'tray',
'trays', 'treasure', 'treasured', 'treasures', 'treat', 'treated', 'treating',
'treatment', 'treats', 'tree', 'trees', 'tremendous', 'tremendously', 'trend',
'trends', 'tri', 'trial', 'trials', 'triangle', 'tribe', 'tribes',
'tribulations', 'trick', 'tricks', 'tricky', 'tricycle', 'tricycles', 'tried',
'tries', 'trigger', 'triggers', 'trikes', 'trimester', 'trip', 'triple',
'tripod', 'tripods', 'tripping', 'trips', 'triumph', 'trombone', 'trouble',
'troubled', 'troubles', 'troubleshoot', 'troubleshooting', 'trucks', 'true',
'truly', 'trumpet', 'trumpets', 'trunk', 'trust', 'trusted', 'trusting',
'truth', 'try', 'trying', 'tub', 'tuba', 'tubano', 'tube', 'tubes', 'tubs',
'tuck', 'tucked', 'tuesday', 'tuition', 'tulsa', 'tumble', 'tumbling',
'tummies', 'tummy', 'tumultuous', 'tune', 'tuned', 'tuners', 'tunes', 'tuning',
'tunnel', 'turkey', 'turmoil', 'turn', 'turnaround', 'turned', 'turning',
'turns', 'turtle', 'tutor', 'tutorial', 'tutorials', 'tutoring', 'tutors', 'tv',
'tvs', 'tweezers', 'twelfth', 'twelve', 'twenty', 'twice', 'twinkle', 'twist',
'twitter', 'two', 'tx', 'tying', 'type', 'typed', 'types', 'typical',
'typically', 'typing', 'ugly', 'ukulele', 'ukuleles', 'ultimate', 'ultimately',
'umbrella', 'umbrellas', 'un', 'unable', 'unacceptable', 'unattainable',
'unavailable', 'unaware', 'unbelievable', 'unbelievably', 'uncertainty',
'uncles', 'uncomfortable', 'uncommon', 'unconditional', 'unconventional',
'uncover', 'under', 'underfunded', 'undergo', 'undergoing', 'undergone',
'underground', 'underline', 'underlying', 'underneath', 'underprivileged',
'underrepresented', 'underserved', 'understand', 'understandable',
'understanding', 'understandings', 'understands', 'understatement',
'understood', 'undertaking', 'underwater', 'underwear', 'undoubtedly',
'unemployed', 'unemployment', 'uneven', 'unexpected', 'unfair', 'unfamiliar',
'unfocused', 'unfold', 'unforgettable', 'unfortunate', 'unfortunately',
'unhealthy', 'unified', 'unifix', 'uniform', 'uniforms', 'unimaginable',
'uninterested', 'uninteresting', 'uninterrupted', 'union', 'unique', 'uniquely',
'uniqueness', 'unit', 'unite', 'united', 'unites', 'units', 'unity',
'universal', 'universe', 'universities', 'university', 'unknown', 'unleash',
'unless', 'unlike', 'unlikely', 'unlimited', 'unlock', 'unlocking', 'unlocks',
'unmatched', 'unmotivated', 'unnatural', 'unnecessary', 'unnoticed', 'unpack',
'unparalleled', 'unprecedented', 'unpredictable', 'unprepared', 'unrealistic',
'unreliable', 'unsafe', 'unseen', 'unstable', 'unstoppable', 'unstructured',
'unsuccessful', 'unsure', 'untapped', 'unthinkable', 'until', 'unusable',
'unused', 'unusual', 'unwanted', 'unwind', 'up', 'upbeat', 'upbringings',
'upcoming', 'update', 'updated', 'updates', 'updating', 'upfront', 'upgrade',
'upgraded', 'upgrades', 'upgrading', 'uphill', 'uphold', 'uplift', 'uplifting',
'upload', 'uploaded', 'upon', 'upper', 'uppercase', 'upright', 'ups', 'upset',
'upside', 'upstate', 'urban', 'urdu', 'urge', 'urgency', 'urgent', 'us', 'usa',
'usable', 'usage', 'usb', 'used', 'useful', 'useless', 'user', 'users', 'uses',
'using', 'usual', 'usually', 'utah', 'utensil', 'utensils', 'utility',
'utilization', 'utilize', 'utilized', 'utilizes', 'utilizing', 'utmost', 'uv',
'va', 'vacation', 'vacations', 'vacuum', 'valid', 'validate', 'validated',
'valley', 'valuable', 'value', 'valued', 'values', 'van', 'variable',
'variables', 'variation', 'variations', 'varied', 'varies', 'varieties',
'variety', 'various', 'varsity', 'vary', 'varying', 'vast', 'vastly', 'vegas',
'vegetable', 'vegetables', 'veggies', 'vehicle', 'vehicles', 'velcro',
'velocity', 'venture', 'venue', 'venues', 'verb', 'verbal', 'verbalize',
'verbally', 'verbs', 'verge', 'vermont', 'vernier', 'versa', 'versatile',
'versatility', 'verse', 'versed', 'version', 'versions', 'versus', 'vertical',
'very', 'vessel', 'vest', 'vested', 'vestibular', 'vests', 'vet', 'veteran',
'veterans', 'veterinarians', 'vex', 'via', 'viable', 'vibe', 'vibrant',
'vibrating', 'vice', 'victim', 'victims', 'victories', 'victory', 'video',
'videos', 'vietnam', 'vietnamese', 'view', 'viewed', 'viewer', 'viewers',
'viewing', 'viewpoints', 'views', 'vigor', 'vigorous', 'village', 'vinyl',
'viola', 'violence', 'violent', 'violin', 'violins', 'virginia', 'virtual',
'virtually', 'visible', 'vision', 'visions', 'visit', 'visited', 'visiting',
'visitors', 'visits', 'vista', 'visual', 'visualization', 'visualize',
'visualizing', 'visually', 'visuals', 'vital', 'vitally', 'vivacious', 'vivid',
'vocabularies', 'vocabulary', 'vocal', 'vocational', 'voice', 'voiced',
'voices', 'void', 'volcano', 'volcanoes', 'volleyball', 'volleyballs', 'volume',
'volumes', 'voluntarily', 'voluntary', 'volunteer', 'volunteered',
'volunteering', 'volunteers', 'voracious', 'vote', 'voted', 'voting', 'vowel',
'vowels', 'vr', 'vs', 'vulnerable', 'wage', 'wagon', 'wait', 'waiting', 'wake',
'waking', 'walk', 'walked', 'walkers', 'walking', 'walks', 'wall', 'walls',
'walt', 'wander', 'wandering', 'wands', 'want', 'wanted', 'wanting', 'wants',
'war', 'ward', 'warlick', 'warm', 'warmer', 'warming', 'warms', 'warmth',
'warning', 'warrior', 'warriors', 'wars', 'was', 'wash', 'washable', 'washed',
'washing', 'washington', 'waste', 'wasted', 'wastes', 'wasting', 'watch',
'watched', 'watches', 'watching', 'water', 'watercolor', 'watercolors',
'watering', 'waters', 'watershed', 'watson', 'wave', 'waves', 'wax', 'way',
'ways', 'weak', 'weakness', 'weaknesses', 'wealth', 'wealthier', 'wealthy',
'weapon', 'wear', 'wearable', 'wearing', 'weather', 'weave', 'weaving', 'web',
'webquests', 'webs', 'website', 'websites', 'wednesday', 'wedo', 'weeds',
'week', 'weekend', 'weekends', 'weekly', 'weeks', 'weigh', 'weight', 'weighted',
'weights', 'weird', 'welcome', 'welcomed', 'welcomes', 'welcoming', 'welding',
'well', 'wellness', 'went', 'were', 'west', 'western', 'wet', 'what',
'whatever', 'whats', 'wheel', 'wheelchair', 'wheelchairs', 'wheels', 'when',
'whenever', 'where', 'whereas', 'wherever', 'whether', 'which', 'whichever',
'while', 'whisper', 'whistles', 'white', 'whiteboard', 'whiteboards', 'who',
'whole', 'wholeheartedly', 'whose', 'why', 'wi', 'wide', 'widely', 'widen',
'wider', 'wifi', 'wiggle', 'wigglers', 'wiggles', 'wiggling', 'wiggly', 'wii',
'wiki', 'wikki', 'wild', 'wildest', 'wildlife', 'wildly', 'will', 'willed',
'willems', 'william', 'williams', 'willing', 'willingly', 'willingness',
'wilson', 'wimpy', 'win', 'wind', 'window', 'windows', 'wing', 'wings', 'winn',
'winner', 'winners', 'winning', 'wins', 'winter', 'winters', 'wipe', 'wiped',
'wipes', 'wire', 'wired', 'wireless', 'wirelessly', 'wires', 'wiring',
'wisconsin', 'wisdom', 'wise', 'wisely', 'wish', 'wished', 'wishes', 'wishing',
'with', 'within', 'without', 'withstand', 'witness', 'witnessed', 'witnessing',
'witty', 'wo', 'wobble', 'wobbling', 'wobbly', 'wolf', 'woman', 'women',
'wonder', 'wondered', 'wonderful', 'wonderfully', 'wondering', 'wonderland',
'wonderment', 'wonders', 'wondrous', 'wont', 'wood', 'wooden', 'woods',
'woodwind', 'woodworking', 'word', 'words', 'workbook', 'workbooks', 'worked',
'worker', 'workers', 'workforce', 'working', 'workout', 'workouts', 'workplace',
'works', 'worksheet', 'worksheets', 'workshop', 'workshops', 'workspace',
'workstation', 'workstations', 'world', 'worldly', 'worlds', 'worldwide',
'worm', 'worms', 'worn', 'worried', 'worries', 'worry', 'worrying', 'worse',
'worst', 'worth', 'worthwhile', 'worthy', 'would', 'woven', 'wow', 'wrap',
'wrapped', 'wrestle', 'wrestlers', 'wrestling', 'wrist', 'wristbands', 'write',
'writer', 'writers', 'writes', 'writing', 'writings', 'written', 'wrong',
'wrote', 'wwii', 'www', 'wyoming', 'xbox', 'xtramath', 'xylophone',
'xylophones', 'ya', 'yard', 'yards', 'yarn', 'yay', 'yeah', 'year', 'yearbook',
'yearbooks', 'yearly', 'yearn', 'yearning', 'years', 'yelling', 'yellow',
'yemen', 'yes', 'yesterday', 'yet', 'yield', 'yoga', 'york', 'you', 'younannan',
'young', 'younger', 'youngest', 'youngsters', 'your', 'yourself', 'youth',
'youtube', 'yummy', 'zeal', 'zearn', 'zen', 'zenergy', 'zero', 'zest', 'zip',
'zippers', 'zone', 'zones', 'zoo', 'zoom', 'zumba']
```

[0]:

```python
from tqdm import tqdm
def get_text_on_idf_features(essay_text):
  '''
  This method collects words within 25 to 75 percentile
  '''
  preprocessed_text = []
  # Iterate each essay text
  for sent in tqdm(essay_text):
    words = sent.split()
    final_sent = ''
    # Iterate for each esssay word
    for word in words:
      # Check word exists in idf corpus
      if(word in feature_name):
        final_sent += ' ' + word

    preprocessed_text.append(final_sent)

  return preprocessed_text
```

[71]:
```
# sample 1
sample_data = get_text_on_idf_features(['I am here', 'I am here', 'I am here',␣
 ↪'I am here'])
sample_data
```

```
100%|| 4/4 [00:00<00:00, 1378.80it/s]
```

[71]:
```
[' am here', ' am here', ' am here', ' am here']
```

[72]:
```
print(X_train['essay'].shape)
print(type(X_train['essay']))
print(X_train['essay'][0:5])
print(X_train['essay'][0:1])
```

```
(69918,)
<class 'pandas.core.series.Series'>
51379    the amazing students class diverse area rural ...
21528    my students best best i work school nearly stu...
91495    i work phenomenal group fifth graders incredib...
42853    this time year i like show students much i car...
28802    as teacher low income high poverty school i se...
Name: essay, dtype: object
51379    the amazing students class diverse area rural ...
Name: essay, dtype: object
```

[73]:
```
# sample 2
sample_data_2 = get_text_on_idf_features(X_train['essay'][0:5])
sample_data_2
```

```
100%|| 5/5 [00:00<00:00, 61.45it/s]
```

```
[73]: [' amazing diverse area rural suburban neighborhoods families diverse ethnic
      socioeconomic backgrounds it diversity guides diverse talents skills
      cooperatively inquire world around these curious curiosity want take action
      projects community new focus ask questions world around us try solve problems
      tools investigate math science questions these chromebooks allow internet
      explore discover answers questions technology tool allows explore experience
      world around these tools necessary cooperative groups answer questions variety
      life experiences never traveled outside rural community others traveled across
      globe throughout lives technology allows common experiences construct meaning
      together',
       ' best best nearly receive free reduced lunch yet kindergarteners every ready
      may always clean clothes fully belly ready as teacher consider duty meet needs
      care deserve best consider responsibility give deserve feel safe time project
      bring safe secure spot project fund cozy furniture safe space developmentally
      kindergarteners ready regulate emotions 100 time expect when emotions get hand
      place cool safe space safe space short term feel safe cool long term invaluable
      skill regulating emotions donations monumental task self regulation control
      imagine wonderful world would adults know developed skills better young
      children',
       ' phenomenal group fifth graders incredible boston full service fully inclusive
      70 english language learners nearly 90 live poverty line limited resources
      spirit diverse language culture ability all inquisitive thoughtful eager however
      bright shining star scholars tools experience success regular basis welcome
      attention hyperactivity challenges socio emotional challenges self regulation
      challenges fully inclusive currently enough sensory tools support level activity
      stimuli control varying levels focused movement fully attend instruction
      maintain focus academically demanding tasks provided daily try best deserve
      better special seating tools noise canceling headphones hand fidget tools
      provide appropriate levels support fully confidently participate activities
      ultimately reach full potential ask tools regular basis sharing handful sensory
      tools old barely functioning know hearts minds light receives tools delight fact
      opportunity get fully achieve',
       ' time year like show much care want see safe place social skills academic
      skills want give access much organization possible take show stronger peers go
      life rug meet reading math writing science lessons carpet area focus best want
      feel rug safe share thoughts ideas here meet together couch offer chance reading
      center feel comfortable read quietly reading buddy in reading area choose couch
      place read quietly stay focused',
       ' as teacher low income high poverty see first hand struggles face inside
      outside excited struggle stay focused goal find variety different strategies
      stay focused increase knowledge high expectations achieve want overcome
      obstacles stand way success most boys tons energy extra energy makes difficult
      sit still stay focused wobble cushions stay focused reduces excess energy wobble
      cushions working whole group small group individual instruction without wobble
      cushions require frequent breaks lessons by wobble cushions sit instruction
      focus meeting goals']
```

[74]:

```
# sample 3 final testing
sample_data_3 = get_text_on_idf_features(['piano specially paramount␣
 ↪microscopes figuring'])
sample_data_3
```

```
100%|| 1/1 [00:00<00:00, 589.42it/s]
```

[74]:

```
[' piano specially paramount microscopes figuring']
```

[75]:

```
print(X_train['essay'][0:10])
```

```
51379    the amazing students class diverse area rural ...
21528    my students best best i work school nearly stu...
91495    i work phenomenal group fifth graders incredib...
42853    this time year i like show students much i car...
28802    as teacher low income high poverty school i se...
22824    our school located urban area composed 70 engl...
98401    i work low income school district located nort...
30533    community cornerstone school district this phi...
22791    i honor teaching intelligent enthusiastic know...
14379    my school title 1 school kids receive free lun...
Name: essay, dtype: object
```

[76]:

```
# filter train data
X_train_essay_orig = X_train['essay']
X_train_assign_2 = []
X_train_assign_2 = get_text_on_idf_features(X_train['essay'])
len(X_train_assign_2)
```

```
100%|| 69918/69918 [19:58<00:00, 58.32it/s]
```

[76]:

```
69918
```

[77]:

```
# filter cv data
X_cv_essay_orig = X_cv['essay']
X_cv_assign_2 = []
X_cv_assign_2 = get_text_on_idf_features(X_cv['essay'])
len(X_cv_assign_2)
```

```
100%|| 17480/17480 [04:59<00:00, 58.35it/s]
```

[77]:

```
17480
```

[78]:

```
# filter test data
X_test_essay_orig = X_test['essay']
X_test_assign_2 = []
X_test_assign_2 = get_text_on_idf_features(X_test['essay'])
```

```
len(X_test_assign_2)
```

```
100%|| 21850/21850 [06:15<00:00, 56.57it/s]
```

[78]:
```
21850
```

[79]:
```
print(X_train['essay'])
print(X_cv['essay'])
print(X_test['essay'])
print(X_train_essay_orig)
print(X_cv_essay_orig)
print(X_test_essay_orig)
```

```
51379    the amazing students class diverse area rural ...
21528    my students best best i work school nearly stu...
91495    i work phenomenal group fifth graders incredib...
42853    this time year i like show students much i car...
28802    as teacher low income high poverty school i se...
                               ...
34888    my scholars quirky passionate energetic learne...
68639    my kind enthusiastic students lives simply see...
50108    our school smaller school area filled great ki...
60386    students not proper exposure early ideas mathe...
65463    my students come low income high poverty area ...
Name: essay, Length: 69918, dtype: object
101372    the read things know the learn places go dr se...
18965     my students little i want give much all studen...
35173     my students far east side chicago they come lo...
13800     i yet meet eighty 5th graders walk room year h...
98655     i wonderful group 5th graders this first year ...
                                ...
34478     the school year started kids love reading in f...
20810     i teach title i school washington d c high pop...
20408     during 2015 2016 school year i pleasure servic...
41302     kindergarten foundation child lifelong educati...
38098     i teach low income area city richmond virginia...
Name: essay, Length: 17480, dtype: object
20512    my students face many challenges realize educa...
83509    my students diverse we 19 languages english sp...
24232    my kindergartenrs energetic eager full life th...
15867    i teach prek sped urban school district most s...
1063     we first grade classroom kindle fire classroom...
                               ...
52904    i teach building 100 years old nestled small c...
87507    my group students full energy love move around...
2243     i teach class 24 kindergarten students title 1...
1037     everyday i want students excited learning my s...
```

```
94631    my students best brightest district last year ...
Name: essay, Length: 21850, dtype: object
51379    the amazing students class diverse area rural ...
21528    my students best best i work school nearly stu...
91495    i work phenomenal group fifth graders incredib...
42853    this time year i like show students much i car...
28802    as teacher low income high poverty school i se...
                               ...
34888    my scholars quirky passionate energetic learne...
68639    my kind enthusiastic students lives simply see...
50108    our school smaller school area filled great ki...
60386    students not proper exposure early ideas mathe...
65463    my students come low income high poverty area ...
Name: essay, Length: 69918, dtype: object
101372    the read things know the learn places go dr se...
18965     my students little i want give much all studen...
35173     my students far east side chicago they come lo...
13800     i yet meet eighty 5th graders walk room year h...
98655     i wonderful group 5th graders this first year ...
                                ...
34478     the school year started kids love reading in f...
20810     i teach title i school washington d c high pop...
20408     during 2015 2016 school year i pleasure servic...
41302     kindergarten foundation child lifelong educati...
38098     i teach low income area city richmond virginia...
Name: essay, Length: 17480, dtype: object
20512    my students face many challenges realize educa...
83509    my students diverse we 19 languages english sp...
24232    my kindergartenrs energetic eager full life th...
15867    i teach prek sped urban school district most s...
1063     we first grade classroom kindle fire classroom...
                               ...
52904    i teach building 100 years old nestled small c...
87507    my group students full energy love move around...
2243     i teach class 24 kindergarten students title 1...
1037     everyday i want students excited learning my s...
94631    my students best brightest district last year ...
Name: essay, Length: 21850, dtype: object
```

[0]:

```
# # convert review word as well i.e. tokenize review text
# # this step is mandatory as we converted essay data
# # - text : text data
# print(X_train.shape, y_train.shape)
# print(X_cv.shape, y_cv.shape)
# print(X_test.shape, y_test.shape)

# print("*"*100)
# (easy_features, rank_dict)  = fit_transform_train_data(X_train_assign_2) #␣
 ↪fit has to happen only on train data

# # we use the fitted CountVectorizer to convert the text to vector
# X_train_essay = transform_data(X_train_assign_2, rank_dict)
# X_cv_essay = transform_data(X_cv_assign_2, rank_dict)
# X_test_essay = transform_data(X_test_assign_2, rank_dict)

# print("After vectorizations")
# print(len(X_train_essay), y_train.shape)
# print(len(X_cv_essay), y_cv.shape)
# print(len(X_test_essay), y_test.shape)
# # print(easy_features)
# print(len(easy_features))
# print("*"*100)
```

[0]:

```
#https://stackoverflow.com/posts/51956230/revisions
t = Tokenizer()
t.fit_on_texts(X_train_assign_2)
vocab_size = len(t.word_index) + 1
# integer encode the documents
encoded_docs = t.texts_to_sequences(X_train_assign_2)
X_train_essay = padded(encoded_docs)
```

[0]:

```
#t = Tokenizer()
#t.fit_on_texts(x_cross.cleaned_essay)
#vocab_size = len(t.word_index) + 1
# integer encode the documents
encoded_docs = t.texts_to_sequences(X_cv_assign_2)
X_cv_essay = padded(encoded_docs)
```

[0]:

```
#t = Tokenizer()
#t.fit_on_texts(x_test.cleaned_essay)
#vocab_size = len(t.word_index) + 1
# integer encode the documents
encoded_docs = t.texts_to_sequences(X_test_assign_2)
X_test_essay = padded(encoded_docs)
```

[0]:

```
embedding_matrix = np.zeros((vocab_size, 300))
for word, i in t.word_index.items():
        embedding_vector = embeddings_index.get(word)
        if embedding_vector is not None:
                embedding_matrix[i] = embedding_vector
```

[85]:

```
project_data['essay']
```

[85]:

```
0          i fortunate enough use fairy tale stem kits cl...
1          imagine 8 9 years old you third grade classroo...
2          having class 24 students comes diverse learner...
```

```
3         i recently read article giving students choice...
4         my students crave challenge eat obstacles brea...
                                ...
109243    our day starts 100 students athletes low incom...
109244    my students range age four five years old atte...
109245    we title 1 school 650 total students our eleme...
109246    i teach many different types students my class...
109247    my first graders eager learn world around they...
Name: essay, Length: 109248, dtype: object
```

[86]:

```python
print(X_train['essay'][0:5].values[0])
string = str(X_train['essay'][0:5].values[0])
type(string)
```

```
the amazing students class diverse area rural suburban neighborhoods they come
families diverse ethnic socioeconomic backgrounds it diversity guides learning
the students class work use diverse talents skills cooperatively inquire world
around these students curious use curiosity learn they want use learn take
action projects help classroom community our class new focus we learning ask
questions world around us try solve problems my students need tools investigate
math science questions these chromebooks allow students use internet explore
discover answers questions technology tool allows students explore experience
world around classroom these tools necessary students work cooperative groups
answer questions my students variety life experiences many students never
traveled outside rural community others traveled across globe throughout lives
the use technology allows students common experiences construct meaning together
nannan
```

[86]:

```
str
```

[0]:

```python
# # zero padding newly created data
# #padding zeros at the begining of each easy to make max len as 250
# max_easy_length = 250
# X_train_essay = pad_sequences(X_train_essay, maxlen=max_easy_length)
# X_cv_essay = pad_sequences(X_cv_essay, maxlen=max_easy_length)
# X_test_essay = pad_sequences(X_test_essay, maxlen=max_easy_length)
# print(X_train_essay.shape)
# print(X_train_essay[0])
# print(X_train_essay[1])
# print(X_cv_essay.shape)
# print(X_cv_essay[0])
# # print(X_cv_essay[1])
# print(X_test_essay.shape)
# print(X_test_essay[0])
# # print(X_test_essay[1])
```

[88]:

```python
print(type(X_train['essay']))
print(X_train['essay'])
```

```
<class 'pandas.core.series.Series'>
51379    the amazing students class diverse area rural ...
21528    my students best best i work school nearly stu...
91495    i work phenomenal group fifth graders incredib...
42853    this time year i like show students much i car...
28802    as teacher low income high poverty school i se...
                               ...
34888    my scholars quirky passionate energetic learne...
68639    my kind enthusiastic students lives simply see...
50108    our school smaller school area filled great ki...
60386    students not proper exposure early ideas mathe...
65463    my students come low income high poverty area ...
Name: essay, Length: 69918, dtype: object
```

[89]:

```
print(type(X_train_assign_2))
print(X_train_assign_2[0:5])
print(X_train['essay'][0:5])
```

```
<class 'list'>
[' amazing diverse area rural suburban neighborhoods families diverse ethnic
socioeconomic backgrounds it diversity guides diverse talents skills
cooperatively inquire world around these curious curiosity want take action
projects community new focus ask questions world around us try solve problems
tools investigate math science questions these chromebooks allow internet
explore discover answers questions technology tool allows explore experience
world around these tools necessary cooperative groups answer questions variety
life experiences never traveled outside rural community others traveled across
globe throughout lives technology allows common experiences construct meaning
together', ' best best nearly receive free reduced lunch yet kindergarteners
every ready may always clean clothes fully belly ready as teacher consider duty
meet needs care deserve best consider responsibility give deserve feel safe time
project bring safe secure spot project fund cozy furniture safe space
developmentally kindergarteners ready regulate emotions 100 time expect when
emotions get hand place cool safe space safe space short term feel safe cool
long term invaluable skill regulating emotions donations monumental task self
regulation control imagine wonderful world would adults know developed skills
better young children', ' phenomenal group fifth graders incredible boston full
service fully inclusive 70 english language learners nearly 90 live poverty line
limited resources spirit diverse language culture ability all inquisitive
thoughtful eager however bright shining star scholars tools experience success
regular basis welcome attention hyperactivity challenges socio emotional
challenges self regulation challenges fully inclusive currently enough sensory
tools support level activity stimuli control varying levels focused movement
fully attend instruction maintain focus academically demanding tasks provided
daily try best deserve better special seating tools noise canceling headphones
hand fidget tools provide appropriate levels support fully confidently
participate activities ultimately reach full potential ask tools regular basis
sharing handful sensory tools old barely functioning know hearts minds light
receives tools delight fact opportunity get fully achieve', ' time year like
show much care want see safe place social skills academic skills want give
access much organization possible take show stronger peers go life rug meet
reading math writing science lessons carpet area focus best want feel rug safe
share thoughts ideas here meet together couch offer chance reading center feel
comfortable read quietly reading buddy in reading area choose couch place read
quietly stay focused', ' as teacher low income high poverty see first hand
struggles face inside outside excited struggle stay focused goal find variety
different strategies stay focused increase knowledge high expectations achieve
want overcome obstacles stand way success most boys tons energy extra energy
makes difficult sit still stay focused wobble cushions stay focused reduces
excess energy wobble cushions working whole group small group individual
instruction without wobble cushions require frequent breaks lessons by wobble
cushions sit instruction focus meeting goals']
51379    the amazing students class diverse area rural ...
21528    my students best best i work school nearly stu...
91495    i work phenomenal group fifth graders incredib...
42853    this time year i like show students much i car...
28802    as teacher low income high poverty school i se...
Name: essay, dtype: object
```

[0]:
```
# create weight matrix on newly created list
# X_train_essay_mat = embedding_mat(pd.
 ↪Series(X_train_assign_2))#X_train['essay']
# print(X_train_essay_mat.shape)
# X_train_essay_mat[5]
```

[93]:
```
X_train_essay_mat = embedding_matrix
X_train_essay_mat.shape
```

[93]:
```
(9976, 300)
```

[94]:
```
# Create model 1 for assignment 2

#input 1 essay text
input1 = Input(shape=(250,))
# x1 = Embedding(input_dim=49042,output_dim= 300)(input1)
x1 = Embedding(input_dim=9976,output_dim=␣
 ↪300,weights=[X_train_essay_mat],trainable=False)(input1)
x1 = SpatialDropout1D(0.3)(x1)
x1 = CuDNNLSTM(128,return_sequences=True)(x1)
# x1 = LSTM(128,return_sequences=True)(x1)
x1 = Flatten()(x1)

#https://medium.com/@davidheffernan_99410/
 ↪an-introduction-to-using-categorical-embeddings-ee686ed7e7f9
cat_vars =␣
 ↪["teacher_prefix","school_state","project_grade_category","clean_categories","clean_subcate
```

```
cat_sizes = {}
cat_embsizes = {}
for cat in cat_vars:
    cat_sizes[cat] = X_train[cat].nunique()
    cat_embsizes[cat] = min(50, cat_sizes[cat]//2+1)

# input 2 school_state 51
input2 = Input(shape=(1,))
x2 = Embedding(input_dim=cat_sizes['school_state']+1,␣
 ↪output_dim=cat_embsizes['school_state'])(input2)#input_dim=52
x2 = Flatten()(x2)

# input 3 project_grade_cat 4
input3 = Input(shape=(1,))
x3 = Embedding(input_dim=cat_sizes['project_grade_category']+1,␣
 ↪output_dim=cat_embsizes['project_grade_category'])(input3)
x3 = Flatten()(x3)

# input 4 clean_categories 9
input4 = Input(shape=(1,))
x4 = Embedding(input_dim=cat_sizes['clean_categories']+1,␣
 ↪output_dim=cat_embsizes['clean_categories'])(input4)
x4 = Flatten()(x4)

# input 5 clean_subcategories 30
input5 = Input(shape=(1,))
x5 = Embedding(input_dim=cat_sizes['clean_subcategories']+1,␣
 ↪output_dim=cat_embsizes['clean_subcategories'])(input5)
x5 = Flatten()(x5)

# input 6 teacher_prefix 5
input6 = Input(shape=(1,))
x6 = Embedding(input_dim=cat_sizes['teacher_prefix']+1,␣
 ↪output_dim=cat_embsizes['teacher_prefix'])(input6)
x6 = Flatten()(x6)

# print(cat_concat)

#input 7 remaining inout
input7 = Input(shape=(1,))
x7 = Dense(16,kernel_initializer=he_normal(),kernel_regularizer=l2(0.
 ↪0001))(input7)
x7 = LeakyReLU()(x7)

concat = concatenate([x1, x2, x3, x4, x5, x6, x7])
```

```
x = Dense(128,kernel_initializer=he_normal(),kernel_regularizer=l2(0.
 ↪0001))(concat)
x = Dropout(0.5)(x)
x = LeakyReLU()(x)
x = Dense(64,kernel_initializer=he_normal(),kernel_regularizer=l2(0.0001))(x)
x = Dropout(0.5)(x)
x = BatchNormalization()(x)
x = LeakyReLU()(x)
x = Dense(32,kernel_initializer=he_normal(),kernel_regularizer=l2(0.0001))(x)
x = LeakyReLU()(x)
# x = Dropout(0.5)(x)
output = Dense(2, activation = 'softmax')(x)

# create model with seven inputs
model = Model([input1,input2,input3,input4,input5,input6,input7], output)#
model.run_eagerly = True
tensorboard = TensorBoard(log_dir='/content/drive/My Drive/LSTM Output/logs/{}'.
 ↪format(time()))
# tensorboard = TensorBoard(log_dir='logs')
model.compile(loss='categorical_crossentropy', optimizer=keras.optimizers.
 ↪Adam(lr=0.0006,decay = 1e-4),metrics=['accuracy', auc])
print(model.summary())
```

```
Model: "model_3"
________________________________________________________________________________
__________________
Layer (type)                    Output Shape         Param #     Connected to
================================================================================
==================
input_17 (InputLayer)           (None, 250)          0

________________________________________________________________________________
__________________
embedding_15 (Embedding)        (None, 250, 300)     2992800     input_17[0][0]

________________________________________________________________________________
__________________
spatial_dropout1d_3 (SpatialDro (None, 250, 300)     0
embedding_15[0][0]

________________________________________________________________________________
__________________
input_18 (InputLayer)           (None, 1)            0

________________________________________________________________________________
__________________
input_19 (InputLayer)           (None, 1)            0

________________________________________________________________________________
__________________
input_20 (InputLayer)           (None, 1)            0

________________________________________________________________________________
```

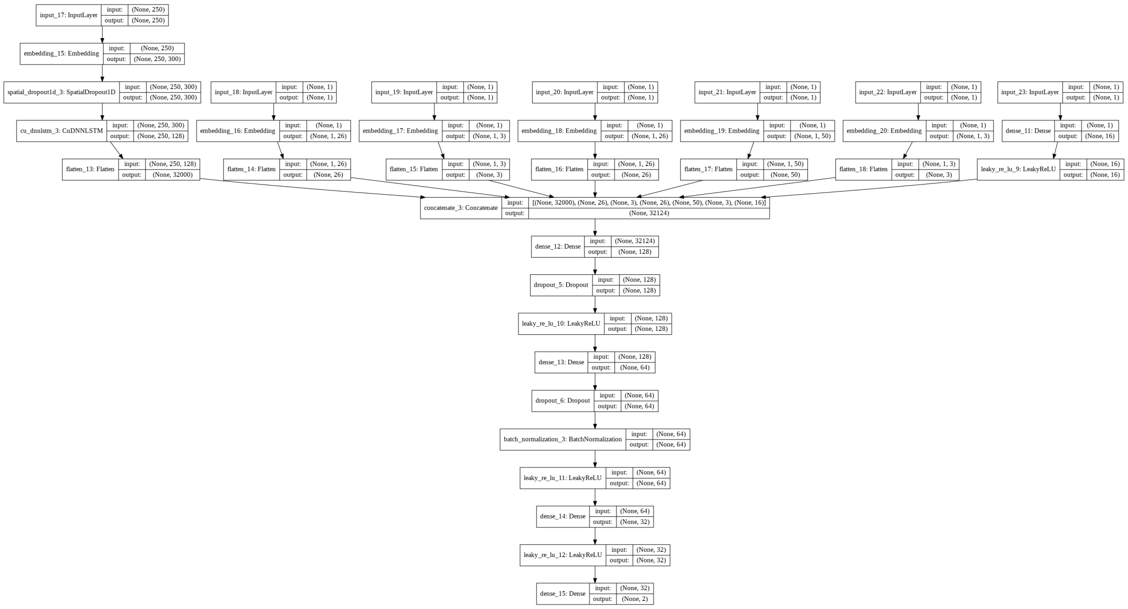
```
------------------
input_21 (InputLayer)           (None, 1)             0
--------------------------------------------------------------------------------
------------------
input_22 (InputLayer)           (None, 1)             0
--------------------------------------------------------------------------------
------------------
input_23 (InputLayer)           (None, 1)             0
--------------------------------------------------------------------------------
------------------
cu_dnnlstm_3 (CuDNNLSTM)        (None, 250, 128)      220160
spatial_dropout1d_3[0][0]
--------------------------------------------------------------------------------
------------------
embedding_16 (Embedding)        (None, 1, 26)         1352        input_18[0][0]
--------------------------------------------------------------------------------
------------------
embedding_17 (Embedding)        (None, 1, 3)          15          input_19[0][0]
--------------------------------------------------------------------------------
------------------
embedding_18 (Embedding)        (None, 1, 26)         1326        input_20[0][0]
--------------------------------------------------------------------------------
------------------
embedding_19 (Embedding)        (None, 1, 50)         19700       input_21[0][0]
--------------------------------------------------------------------------------
------------------
embedding_20 (Embedding)        (None, 1, 3)          18          input_22[0][0]
--------------------------------------------------------------------------------
------------------
dense_11 (Dense)                (None, 16)            32          input_23[0][0]
--------------------------------------------------------------------------------
------------------
flatten_13 (Flatten)            (None, 32000)         0
cu_dnnlstm_3[0][0]
--------------------------------------------------------------------------------
------------------
flatten_14 (Flatten)            (None, 26)            0
embedding_16[0][0]
--------------------------------------------------------------------------------
------------------
flatten_15 (Flatten)            (None, 3)             0
embedding_17[0][0]
--------------------------------------------------------------------------------
------------------
flatten_16 (Flatten)            (None, 26)            0
embedding_18[0][0]
--------------------------------------------------------------------------------
------------------
```

```
flatten_17 (Flatten)            (None, 50)           0
embedding_19[0][0]

--------------------------------------------------------------------------------
------------------
flatten_18 (Flatten)            (None, 3)            0
embedding_20[0][0]

--------------------------------------------------------------------------------
------------------
leaky_re_lu_9 (LeakyReLU)       (None, 16)           0           dense_11[0][0]

--------------------------------------------------------------------------------
------------------
concatenate_3 (Concatenate)     (None, 32124)        0
flatten_13[0][0]
flatten_14[0][0]
flatten_15[0][0]
flatten_16[0][0]
flatten_17[0][0]
flatten_18[0][0]
leaky_re_lu_9[0][0]

--------------------------------------------------------------------------------
------------------
dense_12 (Dense)                (None, 128)          4112000
concatenate_3[0][0]

--------------------------------------------------------------------------------
------------------
dropout_5 (Dropout)             (None, 128)          0           dense_12[0][0]

--------------------------------------------------------------------------------
------------------
leaky_re_lu_10 (LeakyReLU)      (None, 128)          0           dropout_5[0][0]

--------------------------------------------------------------------------------
------------------
dense_13 (Dense)                (None, 64)           8256
leaky_re_lu_10[0][0]

--------------------------------------------------------------------------------
------------------
dropout_6 (Dropout)             (None, 64)           0           dense_13[0][0]

--------------------------------------------------------------------------------
------------------
batch_normalization_3 (BatchNor (None, 64)           256         dropout_6[0][0]

--------------------------------------------------------------------------------
------------------
leaky_re_lu_11 (LeakyReLU)      (None, 64)           0
batch_normalization_3[0][0]

--------------------------------------------------------------------------------
------------------
dense_14 (Dense)                (None, 32)           2080
leaky_re_lu_11[0][0]

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

```
------------------
leaky_re_lu_12 (LeakyReLU)      (None, 32)           0           dense_14[0][0]

--------------------------------------------------------------------------------
------------------
dense_15 (Dense)                (None, 2)            66
leaky_re_lu_12[0][0]
================================================================================
==================
Total params: 7,358,061
Trainable params: 4,365,133
Non-trainable params: 2,992,928

--------------------------------------------------------------------------------
------------------
None
```

[95]:
```python
#https://machinelearningmastery.com/
 ↪visualize-deep-learning-neural-network-model-keras/
from keras.utils.vis_utils import plot_model
plot_model(model, to_file='/content/drive/My Drive/LSTM Output/model_2.png',␣
 ↪show_shapes=True, show_layer_names=True)
```

[95]:


input_17: InputLayer | input: (None, 250) | output: (None, 250)
embedding_15: Embedding | input: (None, 250) | output: (None, 250, 300)
spatial_dropout1d_3: SpatialDropout1D | input: (None, 250, 300) | output: (None, 250, 300)
cu_dnnlstm_3: CuDNNLSTM | input: (None, 250, 300) | output: (None, 250, 128)
flatten_13: Flatten | input: (None, 250, 128) | output: (None, 32000)
input_18: InputLayer | input: (None, 1) | output: (None, 1)
embedding_16: Embedding | input: (None, 1) | output: (None, 1, 26)
flatten_14: Flatten | input: (None, 1, 26) | output: (None, 26)
input_19: InputLayer | input: (None, 1) | output: (None, 1)
embedding_17: Embedding | input: (None, 1) | output: (None, 1, 3)
flatten_15: Flatten | input: (None, 1, 3) | output: (None, 3)
input_20: InputLayer | input: (None, 1) | output: (None, 1)
embedding_18: Embedding | input: (None, 1) | output: (None, 1, 26)
flatten_16: Flatten | input: (None, 1, 26) | output: (None, 26)
input_21: InputLayer | input: (None, 1) | output: (None, 1)
embedding_19: Embedding | input: (None, 1) | output: (None, 1, 50)
flatten_17: Flatten | input: (None, 1, 50) | output: (None, 50)
input_22: InputLayer | input: (None, 1) | output: (None, 1)
embedding_20: Embedding | input: (None, 1) | output: (None, 1, 3)
flatten_18: Flatten | input: (None, 1, 3) | output: (None, 3)
input_23: InputLayer | input: (None, 1) | output: (None, 1)
dense_11: Dense | input: (None, 1) | output: (None, 16)
leaky_re_lu_9: LeakyReLU | input: (None, 16) | output: (None, 16)
concatenate_3: Concatenate | input: [(None, 32000), (None, 26), (None, 3), (None, 26), (None, 50), (None, 3), (None, 16)] | output: (None, 32124)
dense_12: Dense | input: (None, 32124) | output: (None, 128)
dropout_5: Dropout | input: (None, 128) | output: (None, 128)
leaky_re_lu_10: LeakyReLU | input: (None, 128) | output: (None, 128)
dense_13: Dense | input: (None, 128) | output: (None, 64)
dropout_6: Dropout | input: (None, 64) | output: (None, 64)
batch_normalization_3: BatchNormalization | input: (None, 64) | output: (None, 64)
leaky_re_lu_11: LeakyReLU | input: (None, 64) | output: (None, 64)
dense_14: Dense | input: (None, 64) | output: (None, 32)
leaky_re_lu_12: LeakyReLU | input: (None, 32) | output: (None, 32)
dense_15: Dense | input: (None, 32) | output: (None, 2)

[96]:
```python
# print shape of train data
print(X_train_essay.shape)
print(X_train_school_state_ohe.shape)
print(X_train_project_grade_category_ohe.shape)
print(X_train_clean_categories_ohe.shape)
print(X_train_clean_subcategories_ohe.shape)
print(X_train_teacher_prefix_ohe.shape)
print(X_train['remaining_input'].shape)
print('*'*100)
# print shape of cv data
print(X_cv_essay.shape)
print(X_cv_school_state_ohe.shape)
print(X_cv_project_grade_category_ohe.shape)
print(X_cv_clean_categories_ohe.shape)
print(X_cv_clean_subcategories_ohe.shape)
print(X_cv_teacher_prefix_ohe.shape)
print(X_cv['remaining_input'].shape)
print('*'*100)
# print shape of test data
print(X_test_essay.shape)
print(X_test_school_state_ohe.shape)
print(X_test_project_grade_category_ohe.shape)
print(X_test_clean_categories_ohe.shape)
print(X_test_clean_subcategories_ohe.shape)
print(X_test_teacher_prefix_ohe.shape)
print(X_test['remaining_input'].shape)
print('*'*100)
```

```
(69918, 250)
(69918, 1)
(69918, 1)
(69918, 1)
(69918, 1)
(69918, 1)
(69918,)
********************************************************************************
********************
(17480, 250)
(17480, 1)
(17480, 1)
(17480, 1)
(17480, 1)
(17480, 1)
(17480,)
********************************************************************************
********************
(21850, 250)
(21850, 1)
(21850, 1)
(21850, 1)
(21850, 1)
(21850, 1)
(21850,)
********************************************************************************
```

```
********************
```

[97]:

```
# with open('/content/drive/My Drive/LSTM Output/abc.gdoc', 'w') as f:
#   f.write('content')

#model fitting
#https://machinelearningmastery.com/check-point-deep-learning-models-keras/
filepath="/content/drive/My Drive/LSTM Output/weights_copy_new_23_2.best.hdf5"

earlystopping_1 = EarlyStopping(monitor='val_loss', patience=2, verbose=1)

checkpoint = ModelCheckpoint(filepath, monitor='val_auc', verbose=1,␣
 ↪save_best_only=True, mode='max')

callbacks_list = [checkpoint,tensorboard,earlystopping_1]

model.fit([X_train_essay, X_train_school_state_ohe,␣
 ↪X_train_project_grade_category_ohe, X_train_clean_categories_ohe,␣
 ↪X_train_clean_subcategories_ohe, X_train_teacher_prefix_ohe,␣
 ↪X_train['remaining_input']], y_train, nb_epoch=50, verbose=1,␣
 ↪batch_size=256, validation_data=([X_cv_essay, X_cv_school_state_ohe,␣
 ↪X_cv_project_grade_category_ohe, X_cv_clean_categories_ohe,␣
 ↪X_cv_clean_subcategories_ohe, X_cv_teacher_prefix_ohe,␣
 ↪X_cv['remaining_input']]  , y_cv), callbacks = callbacks_list)
```

```
Train on 69918 samples, validate on 17480 samples
Epoch 1/50
69918/69918 [==============================] - 36s 511us/step - loss: 0.4862 -
acc: 0.8319 - auc: 0.5976 - val_loss: 0.5006 - val_acc: 0.8456 - val_auc: 0.6741

Epoch 00001: val_auc improved from -inf to 0.67413, saving model to
/content/drive/My Drive/LSTM Output/weights_copy_new_23_2.best.hdf5
Epoch 2/50
69918/69918 [==============================] - 34s 486us/step - loss: 0.4293 -
acc: 0.8475 - auc: 0.6776 - val_loss: 0.4584 - val_acc: 0.8504 - val_auc: 0.7081

Epoch 00002: val_auc improved from 0.67413 to 0.70813, saving model to
/content/drive/My Drive/LSTM Output/weights_copy_new_23_2.best.hdf5
Epoch 3/50
69918/69918 [==============================] - 34s 485us/step - loss: 0.4179 -
acc: 0.8480 - auc: 0.7020 - val_loss: 0.4299 - val_acc: 0.8509 - val_auc: 0.7069

Epoch 00003: val_auc did not improve from 0.70813
Epoch 4/50
69918/69918 [==============================] - 34s 485us/step - loss: 0.4102 -
acc: 0.8489 - auc: 0.7138 - val_loss: 0.4130 - val_acc: 0.8526 - val_auc: 0.7018
```

```
Epoch 00004: val_auc did not improve from 0.70813
Epoch 5/50
69918/69918 [==============================] - 34s 484us/step - loss: 0.4054 -
acc: 0.8495 - auc: 0.7196 - val_loss: 0.4343 - val_acc: 0.8504 - val_auc: 0.7160

Epoch 00005: val_auc improved from 0.70813 to 0.71605, saving model to
/content/drive/My Drive/LSTM Output/weights_copy_new_23_2.best.hdf5
Epoch 6/50
69918/69918 [==============================] - 34s 483us/step - loss: 0.4007 -
acc: 0.8497 - auc: 0.7279 - val_loss: 0.4104 - val_acc: 0.8530 - val_auc: 0.7311

Epoch 00006: val_auc improved from 0.71605 to 0.73106, saving model to
/content/drive/My Drive/LSTM Output/weights_copy_new_23_2.best.hdf5
Epoch 7/50
69918/69918 [==============================] - 34s 483us/step - loss: 0.3961 -
acc: 0.8506 - auc: 0.7347 - val_loss: 0.4145 - val_acc: 0.8533 - val_auc: 0.7316

Epoch 00007: val_auc improved from 0.73106 to 0.73163, saving model to
/content/drive/My Drive/LSTM Output/weights_copy_new_23_2.best.hdf5
Epoch 8/50
69918/69918 [==============================] - 34s 483us/step - loss: 0.3918 -
acc: 0.8523 - auc: 0.7411 - val_loss: 0.3904 - val_acc: 0.8543 - val_auc: 0.7443

Epoch 00008: val_auc improved from 0.73163 to 0.74430, saving model to
/content/drive/My Drive/LSTM Output/weights_copy_new_23_2.best.hdf5
Epoch 9/50
69918/69918 [==============================] - 34s 483us/step - loss: 0.3893 -
acc: 0.8516 - auc: 0.7445 - val_loss: 0.4014 - val_acc: 0.8539 - val_auc: 0.7315

Epoch 00009: val_auc did not improve from 0.74430
Epoch 10/50
69918/69918 [==============================] - 34s 484us/step - loss: 0.3853 -
acc: 0.8541 - auc: 0.7518 - val_loss: 0.3927 - val_acc: 0.8510 - val_auc: 0.7393

Epoch 00010: val_auc did not improve from 0.74430
Epoch 00010: early stopping
```

[97]:

```
<keras.callbacks.History at 0x7f9310cbedd8>
```

[0]:

```
# Load model with best weights for assignment 2

#input 1 essay text
input1 = Input(shape=(250,))
# x1 = Embedding(input_dim=49042,output_dim= 300)(input1)
x1 = Embedding(input_dim=9976,output_dim=␣
 ↪300,weights=[X_train_essay_mat],trainable=False)(input1)
x1 = SpatialDropout1D(0.3)(x1)
x1 = CuDNNLSTM(128,return_sequences=True)(x1)
```

```
# x1 = LSTM(128,return_sequences=True)(x1)
x1 = Flatten()(x1)

#https://medium.com/@davidheffernan_99410/
 ↪an-introduction-to-using-categorical-embeddings-ee686ed7e7f9
cat_vars =␣
 ↪["teacher_prefix","school_state","project_grade_category","clean_categories","clean_subcate
cat_sizes = {}
cat_embsizes = {}
for cat in cat_vars:
    cat_sizes[cat] = X_train[cat].nunique()
    cat_embsizes[cat] = min(50, cat_sizes[cat]//2+1)

# input 2 school_state 51
input2 = Input(shape=(1,))
x2 = Embedding(input_dim=cat_sizes['school_state']+1,␣
 ↪output_dim=cat_embsizes['school_state'])(input2)#input_dim=52
x2 = Flatten()(x2)

# input 3 project_grade_cat 4
input3 = Input(shape=(1,))
x3 = Embedding(input_dim=cat_sizes['project_grade_category']+1,␣
 ↪output_dim=cat_embsizes['project_grade_category'])(input3)
x3 = Flatten()(x3)

# input 4 clean_categories 9
input4 = Input(shape=(1,))
x4 = Embedding(input_dim=cat_sizes['clean_categories']+1,␣
 ↪output_dim=cat_embsizes['clean_categories'])(input4)
x4 = Flatten()(x4)

# input 5 clean_subcategories 30
input5 = Input(shape=(1,))
x5 = Embedding(input_dim=cat_sizes['clean_subcategories']+1,␣
 ↪output_dim=cat_embsizes['clean_subcategories'])(input5)
x5 = Flatten()(x5)

# input 6 teacher_prefix 5
input6 = Input(shape=(1,))
x6 = Embedding(input_dim=cat_sizes['teacher_prefix']+1,␣
 ↪output_dim=cat_embsizes['teacher_prefix'])(input6)
x6 = Flatten()(x6)

# print(cat_concat)

#input 7 remaining inout
```

```
input7 = Input(shape=(1,))
x7 = Dense(16,kernel_initializer=he_normal(),kernel_regularizer=l2(0.
 ↪0001))(input7)
x7 = LeakyReLU()(x7)

concat = concatenate([x1, x2, x3, x4, x5, x6, x7])

x = Dense(128,kernel_initializer=he_normal(),kernel_regularizer=l2(0.
 ↪0001))(concat)
x = Dropout(0.5)(x)
x = LeakyReLU()(x)
x = Dense(64,kernel_initializer=he_normal(),kernel_regularizer=l2(0.0001))(x)
x = Dropout(0.5)(x)
x = BatchNormalization()(x)
x = LeakyReLU()(x)
x = Dense(32,kernel_initializer=he_normal(),kernel_regularizer=l2(0.0001))(x)
x = LeakyReLU()(x)
# x = Dropout(0.5)(x)
output = Dense(2, activation = 'softmax')(x)

# create model with seven inputs
model = Model([input1,input2,input3,input4,input5,input6,input7], output)#
model.run_eagerly = True
tensorboard = TensorBoard(log_dir='/content/drive/My Drive/LSTM Output/logs/{}'.
 ↪format(time()))
# tensorboard = TensorBoard(log_dir='logs')
model.load_weights(filepath)
model.compile(loss='categorical_crossentropy', optimizer=keras.optimizers.
 ↪Adam(lr=0.0006,decay = 1e-4),metrics=['accuracy', auc])
```

[99]:

```
print("AUC for test data: %0.3f"%roc_auc_score(y_test,model.
 ↪predict([X_test_essay, X_test_school_state_ohe,␣
 ↪X_test_project_grade_category_ohe, X_test_clean_categories_ohe,␣
 ↪X_test_clean_subcategories_ohe, X_test_teacher_prefix_ohe,␣
 ↪X_test['remaining_input']])))
print("AUC for CV data: %0.3f"%roc_auc_score(y_cv,model.predict([X_cv_essay,␣
 ↪X_cv_school_state_ohe, X_cv_project_grade_category_ohe,␣
 ↪X_cv_clean_categories_ohe, X_cv_clean_subcategories_ohe,␣
 ↪X_cv_teacher_prefix_ohe, X_cv['remaining_input']])))
print("AUC for train data: %0.3f"%roc_auc_score(y_train,model.
 ↪predict([X_train_essay, X_train_school_state_ohe,␣
 ↪X_train_project_grade_category_ohe, X_train_clean_categories_ohe,␣
 ↪X_train_clean_subcategories_ohe, X_train_teacher_prefix_ohe,␣
 ↪X_train['remaining_input']])))
```

```
AUC for test data: 0.748
AUC for CV data: 0.744
AUC for train data: 0.763
```

Assignment 2 Tensorboard Image
Ref: https://i.imgur.com/MclYcac.png
Assignment 3

- **input_seq_total_text_data**:
- **Other_than_text_data**:
  . Convert all your Categorical values to onehot coded and then concatenate all these onehot vectors . Neumerical values and use CNN1D as shown in above figure. . You are free to choose all CNN parameters like kernel sizes, stride.

[100]:
```
# process other than text data
# Convert all your Categorical values to onehot coded and then concatenate all these onehot vectors
# Neumerical values

# One hot encoding of Categorical Feature
# - school_state : categorical data
vectorizer = CountVectorizer()
vectorizer.fit(X_train['school_state'].values)# Fit has to happen only on train data

X_train_school_state_ohe = vectorizer.transform(X_train['school_state'].values)
X_cv_school_state_ohe = vectorizer.transform(X_cv['school_state'].values)
X_test_school_state_ohe = vectorizer.transform(X_test['school_state'].values)

school_state_features = vectorizer.get_feature_names()

print(X_train_school_state_ohe.shape, y_train.shape)
print(X_cv_school_state_ohe.shape, y_cv.shape)
print(X_test_school_state_ohe.shape, y_test.shape)
print(vectorizer.get_feature_names())
print('*'*100)
```

```
(69918, 51) (69918, 2)
(17480, 51) (17480, 2)
(21850, 51) (21850, 2)
['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']
********************************************************************************
********************
```

[101]:
```
# One hot encoding of Categorical Feature
# - clean_categories : categorical data
```

```
vectorizer = CountVectorizer()
vectorizer.fit(X_train['clean_categories'].values)# Fit has to happen only on␣
 ↪train data

X_train_clean_categories_ohe = vectorizer.transform(X_train['clean_categories'].
 ↪values)
X_cv_clean_categories_ohe = vectorizer.transform(X_cv['clean_categories'].
 ↪values)
X_test_clean_categories_ohe = vectorizer.transform(X_test['clean_categories'].
 ↪values)

clean_categories_features = vectorizer.get_feature_names()

print(X_train_clean_categories_ohe.shape, y_train.shape)
print(X_cv_clean_categories_ohe.shape, y_cv.shape)
print(X_test_clean_categories_ohe.shape, y_test.shape)
print(vectorizer.get_feature_names())
print('*'*100)
```

```
(69918, 9) (69918, 2)
(17480, 9) (17480, 2)
(21850, 9) (21850, 2)
['appliedlearning', 'care_hunger', 'health_sports', 'history_civics',
'literacy_language', 'math_science', 'music_arts', 'specialneeds', 'warmth']
********************************************************************************
********************
```

[102]:
```
# One hot encoding of Categorical Feature
# - clean_subcategories : categorical data
vectorizer = CountVectorizer()
vectorizer.fit(X_train['clean_subcategories'].values)# Fit has to happen only␣
 ↪on train data

X_train_clean_subcategories_ohe = vectorizer.
 ↪transform(X_train['clean_subcategories'].values)
X_cv_clean_subcategories_ohe = vectorizer.transform(X_cv['clean_subcategories'].
 ↪values)
X_test_clean_subcategories_ohe = vectorizer.
 ↪transform(X_test['clean_subcategories'].values)

clean_subcategories_features = vectorizer.get_feature_names()

print(X_train_clean_subcategories_ohe.shape, y_train.shape)
print(X_cv_clean_subcategories_ohe.shape, y_cv.shape)
print(X_test_clean_subcategories_ohe.shape, y_test.shape)
print(vectorizer.get_feature_names())
print('*'*100)
```

```
(69918, 30) (69918, 2)
(17480, 30) (17480, 2)
(21850, 30) (21850, 2)
['appliedsciences', 'care_hunger', 'charactereducation', 'civics_government',
'college_careerprep', 'communityservice', 'earlydevelopment', 'economics',
'environmentalscience', 'esl', 'extracurricular', 'financialliteracy',
'foreignlanguages', 'gym_fitness', 'health_lifescience', 'health_wellness',
'history_geography', 'literacy', 'literature_writing', 'mathematics', 'music',
'nutritioneducation', 'other', 'parentinvolvement', 'performingarts',
'socialsciences', 'specialneeds', 'teamsports', 'visualarts', 'warmth']
********************************************************************************
********************
```

[103]:

```
print(X_train['project_grade_category'])
# One hot encoding of Categorical Feature
# - project_grade_category : categorical data
# Convert one hot encoding for project grade category
vectorizer = CountVectorizer()
vectorizer.fit(X_train['project_grade_category'].values)# Fit has to happen␣
 ↪only on train data

X_train_project_grade_category_ohe = vectorizer.
 ↪transform(X_train['project_grade_category'].values)
X_cv_project_grade_category_ohe = vectorizer.
 ↪transform(X_cv['project_grade_category'].values)
X_test_project_grade_category_ohe = vectorizer.
 ↪transform(X_test['project_grade_category'].values)

project_grade_category_features = vectorizer.get_feature_names()

print(X_train_project_grade_category_ohe.shape, y_train.shape)
print(X_cv_project_grade_category_ohe.shape, y_cv.shape)
print(X_test_project_grade_category_ohe.shape, y_test.shape)
print(vectorizer.get_feature_names())
print('*'*100)
```

```
51379    grades_prek_2
21528    grades_prek_2
91495       grades_3_5
42853    grades_prek_2
28802       grades_3_5
             ...
34888       grades_6_8
68639      grades_9_12
50108      grades_9_12
60386    grades_prek_2
65463       grades_6_8
```

```
Name: project_grade_category, Length: 69918, dtype: object
(69918, 4) (69918, 2)
(17480, 4) (17480, 2)
(21850, 4) (21850, 2)
['grades_3_5', 'grades_6_8', 'grades_9_12', 'grades_prek_2']
********************************************************************************
********************
```

[104]:

```
# One hot encoding of Categorical Feature
# - teacher_prefix : categorical data
print(X_train['teacher_prefix'])
vectorizer = CountVectorizer()
vectorizer.fit(X_train['teacher_prefix'].values)# Fit has to happen only on␣
 ↪train data

X_train_teacher_prefix_ohe = vectorizer.transform(X_train['teacher_prefix'].
 ↪values)
X_cv_clean_teacher_prefix_ohe = vectorizer.transform(X_cv['teacher_prefix'].
 ↪values)
X_test_clean_teacher_prefix_ohe = vectorizer.transform(X_test['teacher_prefix'].
 ↪values)

teacher_prefix_features = vectorizer.get_feature_names()

print(X_train_teacher_prefix_ohe.shape, y_train.shape)
print(X_cv_clean_teacher_prefix_ohe.shape, y_cv.shape)
print(X_test_clean_teacher_prefix_ohe.shape, y_test.shape)
print(vectorizer.get_feature_names())
print('*'*100)
```

```
51379    mrs
21528     ms
91495    mrs
42853     ms
28802    mrs
        ...
34888     ms
68639     ms
50108     ms
60386     ms
65463    mrs
Name: teacher_prefix, Length: 69918, dtype: object
(69918, 5) (69918, 2)
(17480, 5) (17480, 2)
(21850, 5) (21850, 2)
['dr', 'mr', 'mrs', 'ms', 'teacher']
********************************************************************************
********************
```

[105]:

```python
# You no need to perform standardization/normalization on numerical data,
# because you will classify data by using gini impurity in decision tree classifier.
# - remaining_input : numerical

X_train_remaining_input_norm = X_train['remaining_input'].values.reshape(-1,1)
X_cv_remaining_input_norm = X_cv['remaining_input'].values.reshape(-1,1)
X_test_remaining_input_norm = X_test['remaining_input'].values.reshape(-1,1)

print("After vectorizations")
print(X_train_remaining_input_norm.shape, y_train.shape)
print(X_cv_remaining_input_norm.shape, y_cv.shape)
print(X_test_remaining_input_norm.shape, y_test.shape)
print("="*100)
```

```
After vectorizations
(69918, 1) (69918, 2)
(17480, 1) (17480, 2)
(21850, 1) (21850, 2)
================================================================================
====================
```

[106]:

```python
# print(categories_one_hot.shape)
# print(sub_categories_one_hot.shape)
# print(text_bow.shape)
# print(price_standardized.shape)
print('Categorical Features')
print('*'*100)
print(X_train_school_state_ohe.shape, y_train.shape)
print(X_cv_school_state_ohe.shape, y_cv.shape)
print(X_test_school_state_ohe.shape, y_test.shape)
print('*'*100)
print(X_train_clean_categories_ohe.shape, y_train.shape)
print(X_cv_clean_categories_ohe.shape, y_cv.shape)
print(X_test_clean_categories_ohe.shape, y_test.shape)
print('*'*100)
print(X_train_clean_subcategories_ohe.shape, y_train.shape)
print(X_cv_clean_subcategories_ohe.shape, y_cv.shape)
print(X_test_clean_subcategories_ohe.shape, y_test.shape)
print('*'*100)
print(X_train_project_grade_category_ohe.shape, y_train.shape)
print(X_cv_project_grade_category_ohe.shape, y_cv.shape)
print(X_test_project_grade_category_ohe.shape, y_test.shape)
print('*'*100)
print(X_train_teacher_prefix_ohe.shape, y_train.shape)
print(X_cv_clean_teacher_prefix_ohe.shape, y_cv.shape)
print(X_test_clean_teacher_prefix_ohe.shape, y_test.shape)
print('*'*100)
print('Numerical Features')
print('*'*100)
print(X_train_remaining_input_norm.shape, y_train.shape)
print(X_cv_remaining_input_norm.shape, y_cv.shape)
print(X_test_remaining_input_norm.shape, y_test.shape)
```

```
Categorical Features
********************************************************************************
********************
(69918, 51) (69918, 2)
(17480, 51) (17480, 2)
(21850, 51) (21850, 2)
********************************************************************************
********************
(69918, 9) (69918, 2)
(17480, 9) (17480, 2)
(21850, 9) (21850, 2)
********************************************************************************
********************
(69918, 30) (69918, 2)
(17480, 30) (17480, 2)
(21850, 30) (21850, 2)
********************************************************************************
********************
(69918, 4) (69918, 2)
(17480, 4) (17480, 2)
(21850, 4) (21850, 2)
********************************************************************************
********************
(69918, 5) (69918, 2)
(17480, 5) (17480, 2)
(21850, 5) (21850, 2)
********************************************************************************
********************
Numerical Features
********************************************************************************
********************
(69918, 1) (69918, 2)
(17480, 1) (17480, 2)
(21850, 1) (21850, 2)
```

[107]:

```
# merge two sparse matrices: https://stackoverflow.com/a/19710648/4084039
from scipy.sparse import hstack
# with the same hstack function we are concatinating a sparse matrix and a␣
 ↪dense matirx :)
```

```
# X = hstack((categories_one_hot, sub_categories_one_hot, text_bow,␣
 ↪price_standardized))
# X.shape

X_train_real = X_train
X_cv_real = X_cv
X_test_real = X_test

X_train = hstack((X_train_school_state_ohe, X_train_clean_categories_ohe,␣
 ↪X_train_clean_subcategories_ohe, X_train_project_grade_category_ohe,␣
 ↪X_train_teacher_prefix_ohe, X_train_remaining_input_norm)).tocsr()
X_cv = hstack((X_cv_school_state_ohe, X_cv_clean_categories_ohe,␣
 ↪X_cv_clean_subcategories_ohe, X_cv_project_grade_category_ohe,␣
 ↪X_cv_clean_teacher_prefix_ohe, X_cv_remaining_input_norm)).tocsr()
X_test = hstack((X_test_school_state_ohe, X_test_clean_categories_ohe,␣
 ↪X_test_clean_subcategories_ohe, X_test_project_grade_category_ohe,␣
 ↪X_test_clean_teacher_prefix_ohe, X_test_remaining_input_norm)).tocsr()

print(X_train_real.shape)
print(X_cv_real.shape)
print(X_test_real.shape)
print(X_train.shape)
print(X_cv.shape)
print(X_test.shape)
```

```
(69918, 9)
(17480, 9)
(21850, 9)
(69918, 100)
(17480, 100)
(21850, 100)
```

[0]:

```
# # convert review word as well i.e. tokenize review text
# # this step is mandatory as we converted essay data
# # - text : text data
# print(X_train.shape, y_train.shape)
# print(X_cv.shape, y_cv.shape)
# print(X_test.shape, y_test.shape)

# print("*"*100)

# (easy_features, rank_dict)  = fit_transform_train_data(X_train_essay_orig) #␣
 ↪fit has to happen only on train data

# # we use the fitted CountVectorizer to convert the text to vector
# X_train_essay = transform_data(X_train_essay_orig, rank_dict)
# X_cv_essay = transform_data(X_cv_essay_orig, rank_dict)
# X_test_essay = transform_data(X_test_essay_orig, rank_dict)

# print("After vectorizations")
# print(len(X_train_essay), y_train.shape)
# print(len(X_cv_essay), y_cv.shape)
# print(len(X_test_essay), y_test.shape)
# # print(easy_features)
# print(len(easy_features))
# print("*"*100)
```

[0]:

```
#https://stackoverflow.com/posts/51956230/revisions
t = Tokenizer()
t.fit_on_texts(X_train_essay_orig)
vocab_size = len(t.word_index) + 1
# integer encode the documents
encoded_docs = t.texts_to_sequences(X_train_essay_orig)
X_train_essay = padded(encoded_docs)
```

[0]:

```
#t = Tokenizer()
#t.fit_on_texts(x_cross.cleaned_essay)
#vocab_size = len(t.word_index) + 1
# integer encode the documents
encoded_docs = t.texts_to_sequences(X_cv_essay_orig)
X_cv_essay = padded(encoded_docs)
```

[0]:

```
#t = Tokenizer()
#t.fit_on_texts(x_test.cleaned_essay)
#vocab_size = len(t.word_index) + 1
# integer encode the documents
encoded_docs = t.texts_to_sequences(X_test_essay_orig)
X_test_essay = padded(encoded_docs)
```

[118]:

```
print("After vectorizations")
print(len(X_train_essay), y_train.shape)
print(len(X_cv_essay), y_cv.shape)
print(len(X_test_essay), y_test.shape)
# print(easy_features)
# print(len(easy_features))
print("*"*100)
```

```
After vectorizations
69918 (69918, 2)
17480 (17480, 2)
21850 (21850, 2)
********************************************************************************
********************
```

[0]:

```
embedding_matrix = np.zeros((vocab_size, 300))
for word, i in t.word_index.items():
        embedding_vector = embeddings_index.get(word)
        if embedding_vector is not None:
                embedding_matrix[i] = embedding_vector
```

[120]:

```
X_train_essay_mat = embedding_matrix
X_train_essay_mat.shape
```

[120]:

```
(47268, 300)
```

[0]:

```
# # zero padding newly created data
# #padding zeros at the begining of each easy to make max len as 250
# max_easy_length = 250
# X_train_essay = pad_sequences(X_train_essay, maxlen=max_easy_length)
# X_cv_essay = pad_sequences(X_cv_essay, maxlen=max_easy_length)
# X_test_essay = pad_sequences(X_test_essay, maxlen=max_easy_length)
# print(X_train_essay.shape)
# print(X_train_essay[0])
# print(X_train_essay[1])
# print(X_cv_essay.shape)
# print(X_cv_essay[0])
# # print(X_cv_essay[1])
# print(X_test_essay.shape)
# print(X_test_essay[0])
# # print(X_test_essay[1])
```

[0]:

```
# # create weight matrix on newly created list
# X_train_essay_mat = embedding_mat(X_train_essay_orig)#X_train['essay']
# print(X_train_essay_mat.shape)
# X_train_essay_mat[5]
```

[0]:

```
# X_train = X_train.todense()
# X_cv = X_cv.todense()
# X_test = X_test.todense()
```

[127]:

```
print(X_train.shape)
print(X_cv.shape)
print(X_test.shape)
```

```
(69918, 99, 1)
(17480, 99, 1)
(21850, 99, 1)
```

[0]:

```
X_train = np.resize(X_train,new_shape=(69918,99,1))
X_cv = np.resize(X_cv,new_shape=(17480,99,1))
X_test = np.resize(X_test,new_shape=(21850,99,1))
```

[129]:

```
print(X_train.shape)
print(X_cv.shape)
print(X_test.shape)
```

```
(69918, 99, 1)
(17480, 99, 1)
(21850, 99, 1)
```

[133]:

```
# Assignment #3 model 1

# input 1
input1 = Input(batch_shape=(None,250))
x1 = Embedding(input_dim=47268,output_dim=␣
 ↪300,weights=[X_train_essay_mat],trainable = False)(input1)
x1 = SpatialDropout1D(0.3)(x1)
x1 = CuDNNLSTM(256,return_sequences=True)(x1)
x1 = Flatten()(x1)

# input 2
input2 = Input(shape=(99,1))
x2 = Conv1D(filters=64,kernel_size=3,strides=1)(input2)
x2 = Conv1D(filters=64,kernel_size=3,strides=1)(x2)
x2 = Flatten()(x2)

# merging both the inputs
concat = concatenate([x1,x2])
x = Dense(512,kernel_initializer=he_normal(),kernel_regularizer=l2(0.
 ↪0001))(concat)
x = Dropout(0.4)(x)
x = LeakyReLU()(x)
x = Dense(256,kernel_initializer=he_normal(),kernel_regularizer=l2(0.0001))(x)
x = Dropout(0.5)(x)
x = BatchNormalization()(x)
x = LeakyReLU()(x)
x = Dense(128,kernel_initializer=he_normal(),kernel_regularizer=l2(0.0001))(x)
x = LeakyReLU()(x)
# x = Dropout(0.6)(x)
output = Dense(2, activation = 'softmax')(x)

# create model with two inputs
model = Model([input1,input2], output)
model.run_eagerly = True
tensorboard = TensorBoard(log_dir='/content/drive/My Drive/LSTM Output/logs/{}'.
 ↪format(time()))
model.compile(loss='categorical_crossentropy', optimizer=keras.optimizers.
 ↪adam(lr=0.0006,decay = 1e-4), metrics=['accuracy', auc])
print(model.summary())
```

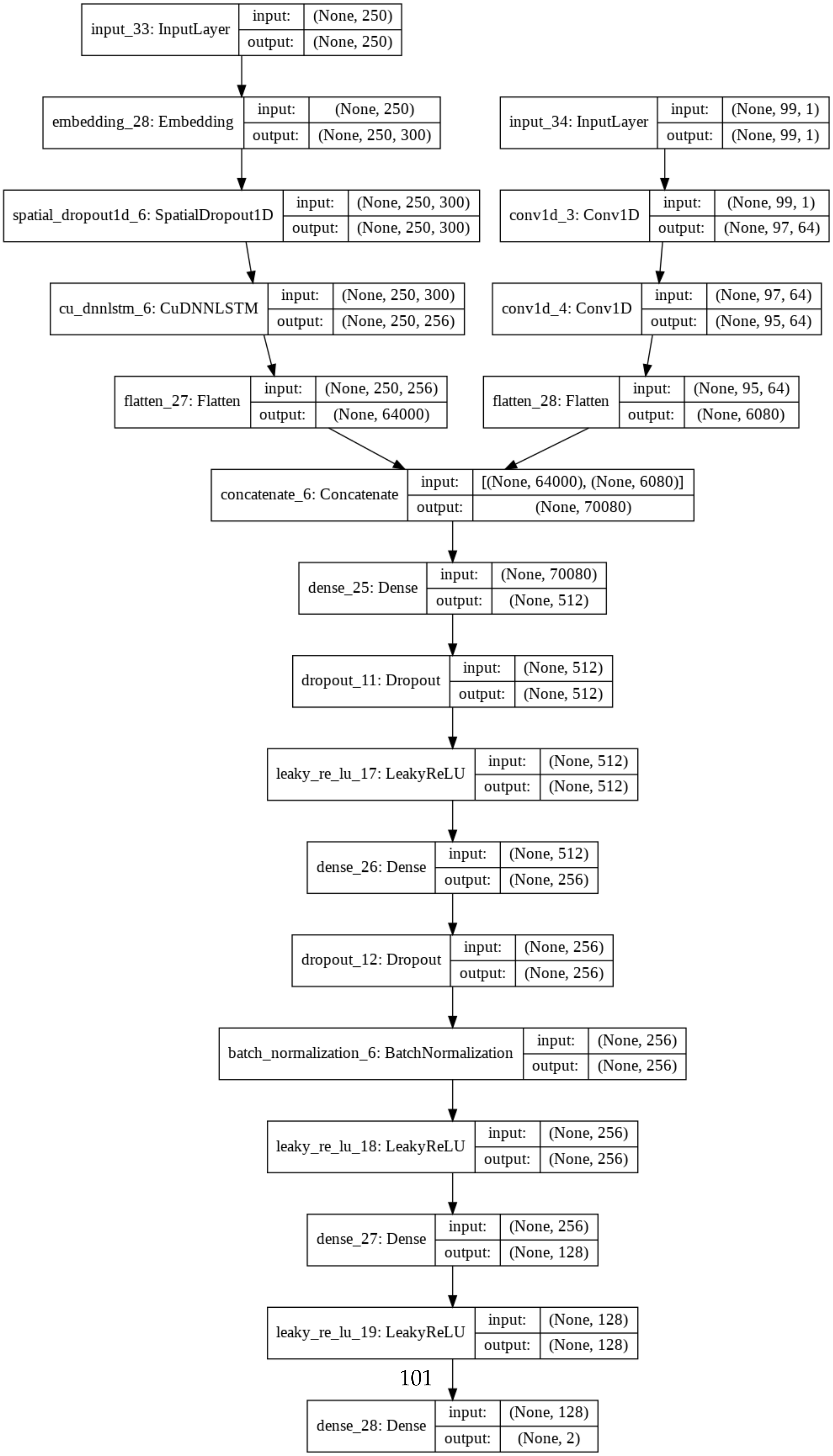
```
Model: "model_6"
________________________________________________________________________________
__________________
```

```
Layer (type)                    Output Shape         Param #     Connected to
================================================================================
==================
input_33 (InputLayer)           (None, 250)          0
--------------------------------------------------------------------------------
------------------
embedding_28 (Embedding)        (None, 250, 300)     14180400    input_33[0][0]
--------------------------------------------------------------------------------
------------------
input_34 (InputLayer)           (None, 99, 1)        0
--------------------------------------------------------------------------------
------------------
spatial_dropout1d_6 (SpatialDro (None, 250, 300)     0
embedding_28[0][0]
--------------------------------------------------------------------------------
------------------
conv1d_3 (Conv1D)               (None, 97, 64)       256         input_34[0][0]
--------------------------------------------------------------------------------
------------------
cu_dnnlstm_6 (CuDNNLSTM)        (None, 250, 256)     571392
spatial_dropout1d_6[0][0]
--------------------------------------------------------------------------------
------------------
conv1d_4 (Conv1D)               (None, 95, 64)       12352       conv1d_3[0][0]
--------------------------------------------------------------------------------
------------------
flatten_27 (Flatten)            (None, 64000)        0
cu_dnnlstm_6[0][0]
--------------------------------------------------------------------------------
------------------
flatten_28 (Flatten)            (None, 6080)         0           conv1d_4[0][0]
--------------------------------------------------------------------------------
------------------
concatenate_6 (Concatenate)     (None, 70080)        0
flatten_27[0][0]
flatten_28[0][0]
--------------------------------------------------------------------------------
------------------
dense_25 (Dense)                (None, 512)          35881472
concatenate_6[0][0]
--------------------------------------------------------------------------------
------------------
dropout_11 (Dropout)            (None, 512)          0           dense_25[0][0]
--------------------------------------------------------------------------------
------------------
leaky_re_lu_17 (LeakyReLU)      (None, 512)          0
dropout_11[0][0]
--------------------------------------------------------------------------------
```

```
------------------
dense_26 (Dense)                (None, 256)          131328
leaky_re_lu_17[0][0]

________________________________________________________________________________
------------------
dropout_12 (Dropout)            (None, 256)          0           dense_26[0][0]
________________________________________________________________________________
------------------
batch_normalization_6 (BatchNor (None, 256)          1024
dropout_12[0][0]

________________________________________________________________________________
------------------
leaky_re_lu_18 (LeakyReLU)      (None, 256)          0
batch_normalization_6[0][0]

________________________________________________________________________________
------------------
dense_27 (Dense)                (None, 128)          32896
leaky_re_lu_18[0][0]

________________________________________________________________________________
------------------
leaky_re_lu_19 (LeakyReLU)      (None, 128)          0           dense_27[0][0]
________________________________________________________________________________
------------------
dense_28 (Dense)                (None, 2)            258
leaky_re_lu_19[0][0]
================================================================================
==================
Total params: 50,811,378
Trainable params: 36,630,466
Non-trainable params: 14,180,912

________________________________________________________________________________
------------------
None
```

[134]:
```python
#https://machinelearningmastery.com/
 ↪visualize-deep-learning-neural-network-model-keras/
from keras.utils.vis_utils import plot_model
plot_model(model, to_file='/content/drive/My Drive/LSTM Output/model_3.png',␣
 ↪show_shapes=True, show_layer_names=True)
```

[134]:


input_33: InputLayer
input:
(None, 250)
output:
(None, 250)
embedding_28: Embedding
input:
(None, 250)
output:
(None, 250, 300)
input_34: InputLayer
input:
(None, 99, 1)
output:
(None, 99, 1)
spatial_dropout1d_6: SpatialDropout1D
input:
(None, 250, 300)
output:
(None, 250, 300)
conv1d_3: Conv1D
input:
(None, 99, 1)
output:
(None, 97, 64)
cu_dnnlstm_6: CuDNNLSTM
input:
(None, 250, 300)
output:
(None, 250, 256)
conv1d_4: Conv1D
input:
(None, 97, 64)
output:
(None, 95, 64)
flatten_27: Flatten
input:
(None, 250, 256)
output:
(None, 64000)
flatten_28: Flatten
input:
(None, 95, 64)
output:
(None, 6080)
concatenate_6: Concatenate
input:
[(None, 64000), (None, 6080)]
output:
(None, 70080)
dense_25: Dense
input:
(None, 70080)
output:
(None, 512)
dropout_11: Dropout
input:
(None, 512)
output:
(None, 512)
leaky_re_lu_17: LeakyReLU
input:
(None, 512)
output:
(None, 512)
dense_26: Dense
input:
(None, 512)
output:
(None, 256)
dropout_12: Dropout
input:
(None, 256)
output:
(None, 256)
batch_normalization_6: BatchNormalization
input:
(None, 256)
output:
(None, 256)
leaky_re_lu_18: LeakyReLU
input:
(None, 256)
output:
(None, 256)
dense_27: Dense
input:
(None, 256)
output:
(None, 128)
leaky_re_lu_19: LeakyReLU
input:
(None, 128)
output:
(None, 128)
dense_28: Dense
input:
(None, 128)
output:
(None, 2)

[135]:

```
# with open('/content/drive/My Drive/LSTM Output/abc.gdoc', 'w') as f:
#   f.write('content')

#model fitting
#https://machinelearningmastery.com/check-point-deep-learning-models-keras/
filepath="/content/drive/My Drive/LSTM Output/weights_copy_assig_3.best.hdf5"

earlystopping_1 = EarlyStopping(monitor='val_loss', patience=2, verbose=1)

checkpoint = ModelCheckpoint(filepath, monitor='val_auc', verbose=1,␣
 ↪save_best_only=True, mode='max')

callbacks_list = [checkpoint,tensorboard, earlystopping_1]

model.fit([X_train_essay, X_train], y_train, nb_epoch=50, verbose=1,␣
 ↪batch_size=256, validation_data=([X_cv_essay, X_cv]  , y_cv), callbacks =␣
 ↪callbacks_list)
```

```
Train on 69918 samples, validate on 17480 samples
Epoch 1/50
69918/69918 [==============================] - 85s 1ms/step - loss: 0.5818 -
acc: 0.8393 - auc: 0.5976 - val_loss: 0.5242 - val_acc: 0.8477 - val_auc: 0.6663

Epoch 00001: val_auc improved from -inf to 0.66626, saving model to
/content/drive/My Drive/LSTM Output/weights_copy_assig_3.best.hdf5
Epoch 2/50
69918/69918 [==============================] - 82s 1ms/step - loss: 0.4999 -
acc: 0.8493 - auc: 0.6868 - val_loss: 0.5229 - val_acc: 0.8496 - val_auc: 0.7009

Epoch 00002: val_auc improved from 0.66626 to 0.70087, saving model to
/content/drive/My Drive/LSTM Output/weights_copy_assig_3.best.hdf5
Epoch 3/50
69918/69918 [==============================] - 82s 1ms/step - loss: 0.4790 -
acc: 0.8505 - auc: 0.7143 - val_loss: 0.5538 - val_acc: 0.8217 - val_auc: 0.7215

Epoch 00003: val_auc improved from 0.70087 to 0.72150, saving model to
/content/drive/My Drive/LSTM Output/weights_copy_assig_3.best.hdf5
Epoch 4/50
69918/69918 [==============================] - 82s 1ms/step - loss: 0.4666 -
acc: 0.8532 - auc: 0.7264 - val_loss: 0.5217 - val_acc: 0.8270 - val_auc: 0.7395

Epoch 00004: val_auc improved from 0.72150 to 0.73954, saving model to
/content/drive/My Drive/LSTM Output/weights_copy_assig_3.best.hdf5
Epoch 5/50
69918/69918 [==============================] - 82s 1ms/step - loss: 0.4562 -
```

```
acc: 0.8536 - auc: 0.7372 - val_loss: 0.4584 - val_acc: 0.8525 - val_auc: 0.7278

Epoch 00005: val_auc did not improve from 0.73954
Epoch 6/50
69918/69918 [==============================] - 82s 1ms/step - loss: 0.4443 -
acc: 0.8548 - auc: 0.7423 - val_loss: 0.4512 - val_acc: 0.8547 - val_auc: 0.7424

Epoch 00006: val_auc improved from 0.73954 to 0.74238, saving model to
/content/drive/My Drive/LSTM Output/weights_copy_assig_3.best.hdf5
Epoch 7/50
69918/69918 [==============================] - 82s 1ms/step - loss: 0.4328 -
acc: 0.8553 - auc: 0.7521 - val_loss: 0.4454 - val_acc: 0.8527 - val_auc: 0.7354

Epoch 00007: val_auc did not improve from 0.74238
Epoch 8/50
69918/69918 [==============================] - 82s 1ms/step - loss: 0.4232 -
acc: 0.8572 - auc: 0.7590 - val_loss: 0.4406 - val_acc: 0.8555 - val_auc: 0.7267

Epoch 00008: val_auc did not improve from 0.74238
Epoch 9/50
69918/69918 [==============================] - 82s 1ms/step - loss: 0.4152 -
acc: 0.8594 - auc: 0.7669 - val_loss: 0.4316 - val_acc: 0.8545 - val_auc: 0.7502

Epoch 00009: val_auc improved from 0.74238 to 0.75017, saving model to
/content/drive/My Drive/LSTM Output/weights_copy_assig_3.best.hdf5
Epoch 10/50
69918/69918 [==============================] - 82s 1ms/step - loss: 0.4067 -
acc: 0.8584 - auc: 0.7744 - val_loss: 0.4351 - val_acc: 0.8528 - val_auc: 0.7496

Epoch 00010: val_auc did not improve from 0.75017
Epoch 11/50
69918/69918 [==============================] - 82s 1ms/step - loss: 0.3972 -
acc: 0.8617 - auc: 0.7836 - val_loss: 0.4360 - val_acc: 0.8528 - val_auc: 0.7492

Epoch 00011: val_auc did not improve from 0.75017
Epoch 00011: early stopping
```

[135]:

```
<keras.callbacks.History at 0x7f92a6788550>
```

[0]:

```python
# Assignment #3 model 2 with optimized weights

# input 1
input1 = Input(batch_shape=(None,250))
x1 = Embedding(input_dim=47268,output_dim=␣
 ↪300,weights=[X_train_essay_mat],trainable = False)(input1)
x1 = SpatialDropout1D(0.3)(x1)
x1 = CuDNNLSTM(256,return_sequences=True)(x1)
x1 = Flatten()(x1)
```

```
# input 2
input2 = Input(shape=(99,1))
x2 = Conv1D(filters=64,kernel_size=3,strides=1)(input2)
x2 = Conv1D(filters=64,kernel_size=3,strides=1)(x2)
x2 = Flatten()(x2)

# merging both the inputs
concat = concatenate([x1,x2])
x = Dense(512,kernel_initializer=he_normal(),kernel_regularizer=l2(0.
 ↪0001))(concat)
x = Dropout(0.4)(x)
x = LeakyReLU()(x)
x = Dense(256,kernel_initializer=he_normal(),kernel_regularizer=l2(0.0001))(x)
x = Dropout(0.5)(x)
x = BatchNormalization()(x)
x = LeakyReLU()(x)
x = Dense(128,kernel_initializer=he_normal(),kernel_regularizer=l2(0.0001))(x)
x = LeakyReLU()(x)
# x = Dropout(0.6)(x)
output = Dense(2, activation = 'softmax')(x)

# create model with two inputs
model = Model([input1,input2], output)
model.run_eagerly = True
tensorboard = TensorBoard(log_dir='/content/drive/My Drive/LSTM Output/logs/{}'.
 ↪format(time()))
model.compile(loss='categorical_crossentropy', optimizer=keras.optimizers.
 ↪adam(lr=0.0006,decay = 1e-4), metrics=['accuracy', auc])
model.load_weights(filepath)
# print(model.summary())
```

[137]:

```
print("Auc for Test data: %0.3f"%roc_auc_score(y_test,model.
 ↪predict([X_test_essay, X_test])))
print("Auc for CV data: %0.3f"%roc_auc_score(y_cv,model.predict([X_cv_essay,␣
 ↪X_cv])))
print("Auc for Train data: %0.3f"%roc_auc_score(y_train,model.
 ↪predict([X_train_essay, X_train])))
```

```
Auc for Test data: 0.760
Auc for CV data: 0.750
Auc for Train data: 0.795
```

Assignment 3 Tensor Board Image
Ref: https://i.imgur.com/mIhsdiX.png

[138]:

```
# Please compare all your models using Prettytable library
#http://zetcode.com/python/prettytable/
```

```python
from prettytable import PrettyTable


x = PrettyTable()
x.field_names = ["Features", "Model", "Epochs", "Train AUC", "CV AUC", "Test␣
 ↪AUC"]
x.add_row(["Assignment 1 (Embedding Layer Encoding)", "CuDNNLSTM", 50, 0.780, 0.
 ↪755, 0.761])
x.add_row(["Assignment 2 (TFIDF based essay text + Embedding Layer Encoding)",␣
 ↪"CuDNNLSTM", 50, 0.763, 0.744, 0.748])
x.add_row(["Assignment 3 (Essay text + Other features combined)",␣
 ↪"CuDNNLSTM+Conv1D", 50, 0.795, 0.750, 0.760])

print(x)
```

```
+------------------------------------------------------------------+------------
------+--------+-----------+--------+----------+
|                             Features                             |      Model
| Epochs | Train AUC | CV AUC | Test AUC |
+------------------------------------------------------------------+------------
------+--------+-----------+--------+----------+
|             Assignment 1 (Embedding Layer Encoding)              |
CuDNNLSTM     |   50   |    0.78   | 0.755  |  0.761   |
| Assignment 2 (TFIDF based essay text + Embedding Layer Encoding) |
CuDNNLSTM     |   50   |   0.763   | 0.744  |  0.748   |
|       Assignment 3 (Essay text + Other features combined)        |
CuDNNLSTM+Conv1D |   50   |   0.795   |  0.75  |   0.76   |
+------------------------------------------------------------------+------------
------+--------+-----------+--------+----------+
```

Conclusion

- TensorBoard is a nice tools for deep learning model profiling and generating graphs.
- GPUs performance degrades when do CPU intensive work like removing higher or lower tfidf words from sentences but it works best for deep learning stuffs.
- EarlyStopping is very nice feature while training deep learning models

[0]: