

TEXT CLASSIFICATION HANDBOOK

***Author:***

***Akash Swamy***

***Machine Learning Engineer***

***Co-Founder: arythmattic.com***

# Software Requirements

## Software

1. Python 3.5
2. Anaconda latest version
3. Pycharm IDE

## Downloading libraries

1. Keras

Pip install keras

1. Tensorflow

Pip install tensorflow

1. Tqdm

Pip install tqdm

1. Nltk

pip install nltk

## Downloading data for libraries

1. Nltk

Open python console and run following commands:

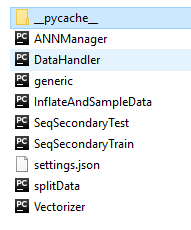
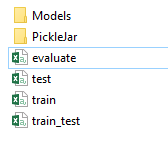
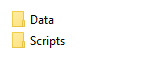
import nltk

nltk.download()

***OR***

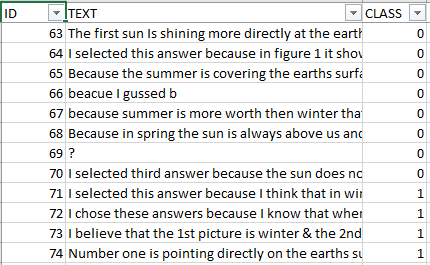
Run file “nltkSetup.py” given in the Scripts folder

# Project Structure

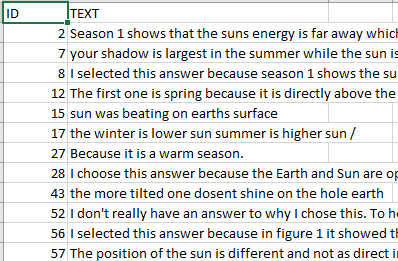


# Data Templates

Train, Train\_test & Evaluate files Data Template:



Test Data Template:



# Instructions

## Load Project

1. Open Project in PyCharm IDE & Set project interpreter as Anaconda

## Set-up Data

1. To Divide the training data into train & evaluate files, Run Script “splitData.py”
   1. Input file: train\_test.csv
   2. Output file: train.csv, evaluate.csv

Variables to enter:

Randomize the division of data: yes/no, if selected yes ‘y’, enter any number greater than 0

Training data fraction: any fraction between 0-1 (e.g 0.8 divides the data into 80% training and 20% evaluation samples)

1. To inflate data by creating duplicates, Run script “InflateAndSampleData.py”.
   1. Input file: train.csv
   2. Output file: train\_samples.csv
   3. Change the train\_sample.csv file format as per the train data template, add ID column and save file as train.csv

Variables to set:

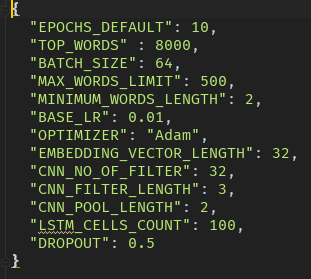


samples\_count : Number of training samples per class

REMOVE\_EXTRA: remove extra samples if samples count for any class is greater than the given samples\_count number

## Model parameters

All the model parameters can be set/changed using the “settings.json” file:



1. EPOCHS\_DEFAULT: Default epochs count for training
2. TOP\_WORDS : Maximum number of words in Bag of words
3. BATCH\_SIZE: Training batch size
4. MAX\_WORDS\_LIMIT: Maximum number of words in one text sample/answer
5. MINIMUM\_WORDS\_LENGTH: Minimum length of a word to be added to Bag of words
6. BASE\_LR: Base learning rate
7. OPTIMIZER: Training optimizer
8. EMBEDDING\_VECTOR\_LENGTH: Length of embedding vector
9. CNN\_NO\_OF\_FILTER: Number of filter in CNN
10. CNN\_FILTER\_LENGTH: filter length in CNN
11. CNN\_POOL\_LENGTH: Pooling size for max pooling
12. LSTM\_CELLS\_COUNT: Number of LSTM cells
13. DROPOUT: Drop out in the model

## Train Model

Run Script “trainModel.py”

1. Input file: train.csv
2. Output file: Model & Data Pickles to be used to predictions (Model & PickleJar folder)

Once the file is executed user is prompted to enter following options:

1. Train new Model or Continue training previously trained Model:

To continue training using the previously trained Model, enter “y” in the console. To train new model enter “n”

1. Number of Epochs

Enter an integer number to set epochs for training the model.

Leave blank to select default value from the settings.json file.

More classes can be added for training by adding more classes in the “CLASS” column of train.csv file. And the run trainModel script to train the Model with the updated classes structure.

## Test/Predict

Run Script “predictModel.py”

1. Input file: test.csv,
2. Default inputs: Trained Model & Pickled data (Model & PickleJar folder)
3. Output file: predictions.csv