Foundation of Intelligent System
Project 2 - Exploring the discourse in reddit
Rishab Lalwani

1. Introduction

1.1 Overview

This report concentrates on the techniques used to recognize which subreddit a title belongs to.
Reddit website is one of the largest growing collections of user-submitted content. Reddit is an American social news aggregation, web content rating, and discussion website. Where subreddits are collected using some inbuilt libraries and them splitted into training, development and testing dataset for further processing.

1.2 Approach used

Selection of appropriate data is important for the study. So we chose the subreddits carefully and the one which are not dependent on each other to get good accuracy. Later the dataset has been cleaned i.e. done bagging on it and splitted it into training, testing and development dataset for further processing, classifiers were used to do so.
Classification is the technique used to identify given a title which subreddit it belongs to. Classification is a supervised learning approach in which the computer program learns from the data input given to it and then uses this learning to classify new observation. The main goal of a classification problem is to identify the category/class to which a new data will fall under. SVM and Random forest is the algorithm used to identify which reddit the post belongs to. SVM is primarily a classier method that performs classification tasks by constructing hyper planes in a multidimensional space that separates cases of different class labels. Random forest classifier creates a set of decision trees from randomly selected subset of training set. It then aggregates the votes from different decision trees to decide the final class of the test object.
Long Short term memory (LSTM) is also performed using tensor flow and keras on the same data set to compare the accuracy.

2. Methods

2.1 Dataset description

The dataset that we selected for analysis came from reddit.com. Choosing the appropriate data is the important factor. The dataset is made up of reddit posts. A reddit scraper was created to pull posts from Reddit website. The reddits we chose were yoga, Chinese and games of thrones and extracted top 500 posts from each of it. Features available in the reddit posts are title, created and body. The dataset is in .json format
The final data set used in the analysis includes 1500 samples from three sub reddit categories. These three categories were chosen to make the task more tractable by selecting three distinct categories where domain specific vocabularies would be less likely to overlap.

2.1.1 Libraries used:

a. Praw:

Praw is a python package which allows for simple access to reddit's API. Praw is a python wrapper which enables us to scrape data from subreddits.

Following are the steps m used to pull the post from reddit using praw library.

1. To install praw
   pip install praw

2. Import praw
   import praw

3. Authentication:
   To have the authentication information, create reddit app and save the authentication information for further process.

```
reddit=praw.Reddit(client_id='UUj3Iby86ia3YQ',client_secret="Ufkis-
VI9jG5_JT6_QlQLnM0Y9E", password='reddit 123', user_agent='Project 2',
        username='rl9703')
```

The above code is for creation of reddit instance and providing it with client_id, client_secret and user_agent.

4. Getting subreddit posts:

```
subreddit 1 = reddit.subreddit('chinese')
subreddit 2 = reddit.subreddit('game of thrones')
subreddit 3 = reddit.subreddit('yoga')
```

Using praw.reddit command now can access all the data and to just scrap the top 500 post from each of the subreddit .top(limit = 500) is used in for loop.
The data collected is a total of 1500 tuples from the different subreddits. Idea of choosing different subreddit to increase the accuracy.

2.2 Data pre-processing

Here the dataset had around 1500 rows of data, which was then lemmatized, stemming was done on raw words and the stop words were removed.
Before performing all the above, the dataset has been transformed into the data frame and later all the subreddits e were concatenated into one data-frame and shuffled. The title was been extracted from it. Lemmatizaton is done to reduce redundant vocabulary, normally to remove inflectional endings and returns the base or dictionary form of a word. Stemming is the process for reducing the inflected words to their stem and stop words such as the,a,an,in were removed. Lemmatization is similar to stemming but it brings context to the words. So it links words with similar meaning to one word.
Text pre-processing includes both Stemming as well as Lemmatization.

To perform above data pre-processing the Natural language toolkit (NLTK) was used:

NLTK is a python library .The Natural Language Toolkit (**NLTK**) is a platform used for building **Python** programs that work with human language data for applying in statistical natural language processing (NLP). It contains text processing libraries for tokenization, parsing, classification, stemming, tagging and semantic reasoning. Following are the steps to use nltk package for data pre-processing:

1. Install NLTK package
   pip install nltk

2. Import NLTK package
   import nltk

3. Call the nltk downloader.
   nltk.download('punkt')

4. Stemming: nltk.stem is a package that performs the stemming
   using different classes.

   Here PorterStemmer is used, which do suffix stripping to
   produce stems from nltk.stem.porter import PorterStemmer

5. Stop words : following command is used remove stop words.
   Stops words are commonly used words.
   This downloads and imports the stopword package/library.
   nltk.download('stopwords')
   from nltk.corpus import stopwords

2.2.1 Vectorization of data

Vectorization is the process of converting an algorithm from operating on a single value at a time to operating on a set of values at one time. Vectorization is the process of executing operations on entire array. There are many techniques used to do so, we have used the Bag of words(BoW) model. In bag of words approach words counts were collected separately over training, development and testing sets and words with count below a frequency threshold value were removed.

All the above steps were performed efficiently and the pre-processing and rules for the number and frequency of token were done using a function called as Countvectorizer, which counts the numbers of times a token appears in the documents.

2.3 Modelling the data

Here 1500 posts from all subreddits was used and was split as follows:

Training : 50%
Development:25%
Testing:25%

The models used are SVM, Random forest classifier, LSTM using Keras and Tensor flow to recognize which reddit the post belongs to. Features that were used to recognize the prediction was Title and found out that the classification done on titles were sufficient instead of using the body. Let's go through all of them:

a. Support Vector Machine

Support vector machine is a set of supervised learning methods used for classification.
It provides the highest accuracy compare to other algorithms. To separate different classes the multidimensional space is produced. To implement the SVM we have used the SciKit - learn library. Where the model is trained, developed and tested.
Here the input to the SVM is the count-vectorized output, i.e. the vector x, vector y which is nothing but the vector count of the titles extracted from the subreddits.

b. Random Forest classifier

Random forest classifier is ensemble algorithm. A random forest is a meta estimator that fits a number of decision tree classifiers on various sub-samples of the dataset and uses averaging to improve the predictive accuracy and control over-fitting. Here the input to the Random forest is the count-vectorized output, i.e. the vector x, vector y which is nothing but the vector count of the titles extracted from the subreddits.

c. Long short term memory using keras and tensor flow:

LSTM is a recurrent neural model. An LSTM network is a recurrent neural network that has LSTM cell blocks in place of our standard neural network layers. These cells have various components called the input gate, the forget gate and the output gate .

2.4 Evaluate the model:

Word cloud

As part of exploring our data set, we used the word cloud Python library to create word-cloud graphics of the most common words in the titles and texts of the Reddit posts that we scraped. We created word clouds which depict most common words in the entire data set, and of certain subreddits to find interesting patterns.

Word Cloud outputs:

- A word cloud depicts the frequency of words appearing in the text. The larger the size of the word, the greater the frequency.
- We have plotted the word clouds for titles of the subreddits
- The argument interpolation="bilinear" in the plt.imshow() is used. This is to make the displayed image appear more smoothly.

Word cloud subreddit Chinese:

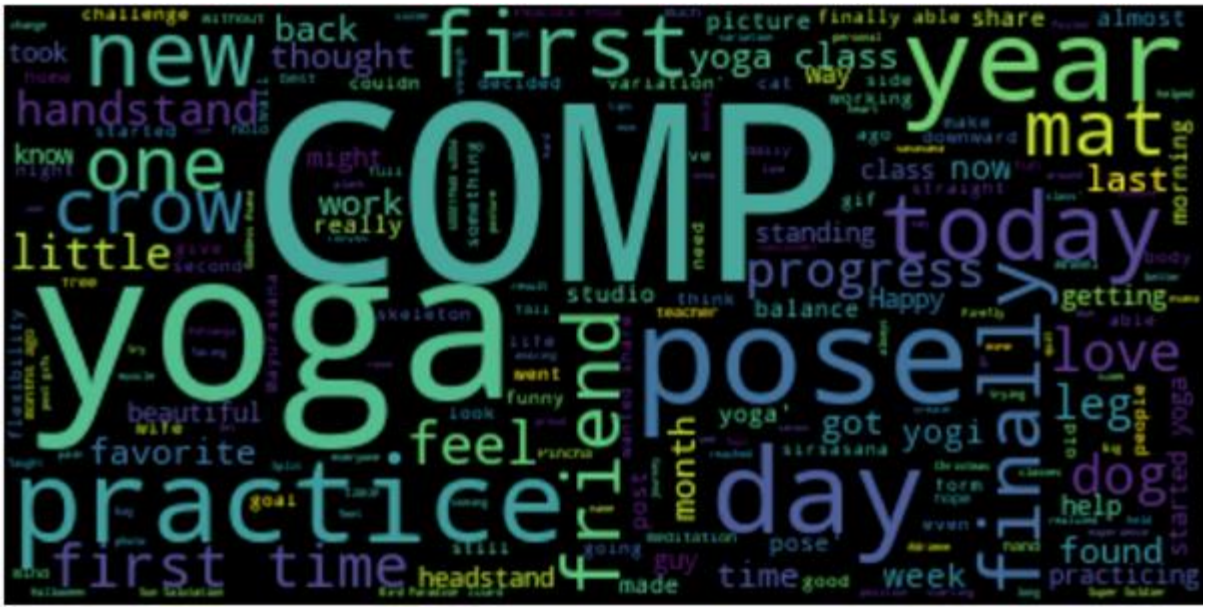
- Max frequency words observed are: ‘Chinese’, ’Mandarin’, ’translate’.

Word cloud reddit gameofthrones:

- Max frequency words observed are: ‘SPOILERS’, ’everything’, ’throne’.

Word Cloud subreddit Yoga:

- Max frequency words observed is: ‘Comp’, ’pose’, ’practice’.



3. Results:

3.1 Classification Output

```
SVM
 Confusion matrix for SVM:
 [[238   5   6]
 [ 12 240   4]
 [ 32   4 209]]

Classification Report:
                precision    recall  f1-score   support

       chinese       0.84      0.96      0.90       249
gameofthrones       0.96      0.94      0.95       256
          yoga       0.95      0.85      0.90       245

    micro avg       0.92      0.92      0.92       750
    macro avg       0.92      0.92      0.92       750
 weighted avg       0.92      0.92      0.92       750

Support Vector Classifier accuracy: 91.6 %

Random Forest
 Confusion matrix for RandomForest:
 [[123   3   4]
 [  7 117   2]
 [ 13   3 103]]

Classification Report:
              precision    recall  f1-score   support

           0       0.86      0.95      0.90       130
           1       0.95      0.93      0.94       126
           2       0.94      0.87      0.90       119

   micro avg       0.91      0.91      0.91       375
   macro avg       0.92      0.91      0.91       375
weighted avg       0.92      0.91      0.91       375

Random Forest Classifier accuracy: 89.86666666666666 %
```

3.1.1 SVM:

- We have used the OnevsRest classifier for multiclass classification.
- Looking at the confusion matrix we analyze the following:
    - The classifier misclassified 48 + 15 = 63 subreddits out of 750 test classifications performed.
    - Which is 91.6% accuracy?

- Precision is a measure of result relevancy,
- While recall is a measure of how many truly relevant results are returned
    - Precision obtained is 84%,96% and 95% for the three subreddits 'chinese', 'gameofthrones' and 'yoga'
    - Recall obtained is 96%,94% and 85% for the three subreddits 'chinese', 'gameofthrones' and 'yoga'.

- We have also performed the f1 score and support for statistics
- F1 score is the harmonic mean,
- The support is the number of occurrences of each class in y_true.
    - F1 score obtained is 86%,95% and 94% for the three subreddits 'chinese',' gameofthrones' and 'yoga'.
    - Support obtained is 249,256 and 245 for the three subreddits 'chinese', 'gameofthrones' and 'yoga'.

- The micro average, macro average and weighted average is shown, which depicts:
    - Micro average: Calculate metrics globally by counting the total true positives, false negatives and false positives.
    - Macro average: Calculate metrics for each label and finds their unweighted mean.
    - Weighted average: Calculate metrics for each label and finds their average weighted by support

3.1.2 Random Forest:

- We have built a forest of trees from the training set (Vector_X ,Vector_Y).
- Looking at the confusion matrix we analyze the following:
    - The classifier misclassified 23 + 9 = 32 subreddits out of 375 test classifications performed.
    - Which is ~90% accuracy.

- Precision is a measure of result relevancy,
- While recall is a measure of how many truly relevant results are returned
    - Precision obtained is 86%,95% and 94% for the three subreddits 'chinese', 'gameofthrones' and 'yoga'

- Recall obtained is 95%,93% and 87% for the three subreddits ‘chinese’, ’gameofthrones’ and ‘yoga’.

- We have also performed the f1 score and support for statistics
- F1 score is the harmonic mean,
- The support is the number of occurrences of each class in y_true.
  - F1 score obtained is 90%,94% and 90% for the three subreddits ‘chinese’,’ gameofthrones’ and ‘yoga’.
  - Support obtained is 130,126 and 119 for the three subreddits ‘chinese’, ’gameofthrones’ and ‘yoga’.

- The micro average, macro average and weighted average is shown, which depicts:
  - Micro average: Calculate metrics globally by counting the total true positives, false negatives and false positives.
  - Macro average: Calculate metrics for each label and finds their unweighted mean.
  - Weighted average: Calculate metrics for each label and finds their average weighted by support

3.2 LSTM using keras and tensor flow

```
Train on 900 samples, validate on 225 samples
Epoch 1/10
```

We perform training on 900 titles of 3 shuffled subreddits (‘Chinese’, ‘gameofthrones’, ‘yoga’)
With 10 epoch’s

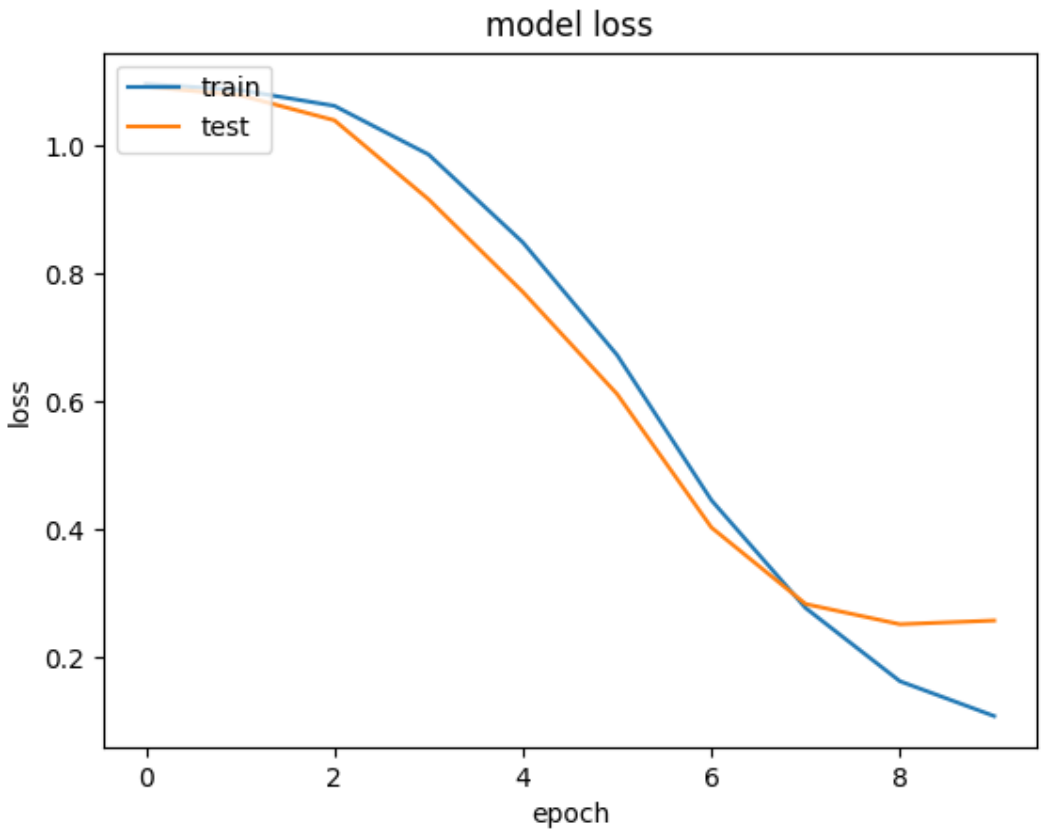
```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
inputs (InputLayer)          (None, 150)               0
_________________________________________________________________
embedding_1 (Embedding)      (None, 150, 50)           50000
_________________________________________________________________
lstm_1 (LSTM)                (None, 64)                29440
_________________________________________________________________
FC1 (Dense)                  (None, 256)               16640
_________________________________________________________________
activation_1 (Activation)    (None, 256)               0
_________________________________________________________________
dropout_1 (Dropout)          (None, 256)               0
_________________________________________________________________
out_layer (Dense)            (None, 3)                 771
_________________________________________________________________
activation_2 (Activation)    (None, 3)                 0
=================================================================
Total params: 96,851
Trainable params: 96,851
Non-trainable params: 0
```

Keras provides this capability with parameters on the LSTM layer, the dropout for configuring the input dropout and recurrent_dropout for configuring the recurrent dropout.

For our RNN model we have used two dropout layers with dropout rate=0.5 and the softmax function for multi-classification as seen in the code.

```
 32/375 [=>............................] - ETA: 0s
 96/375 [======>.......................] - ETA: 0s
160/375 [===========>..................] - ETA: 0s
224/375 [================>.............] - ETA: 0s
288/375 [=====================>........] - ETA: 0s
352/375 [==========================>...] - ETA: 0s
375/375 [==============================] - 0s 1ms/step
Accuracy: 93.33333339691163 %
```

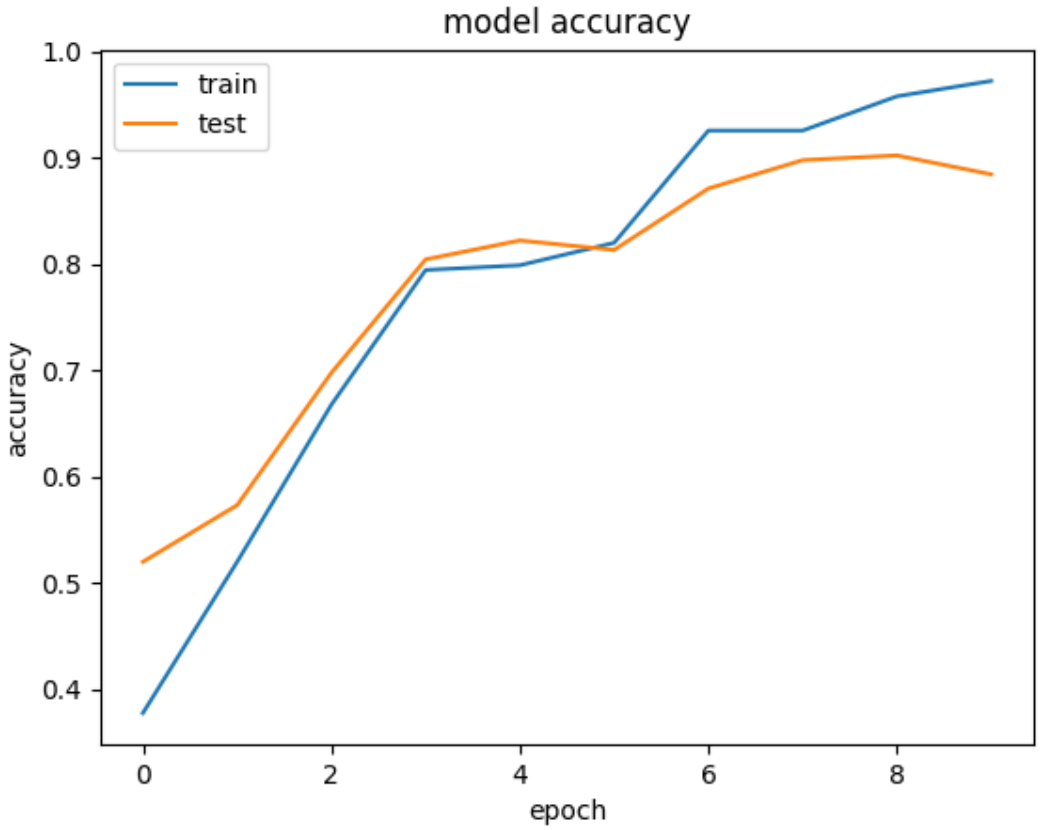
We can see dropout having the desired impact on training.



We achieved an accuracy of 93.33% when testing on 25% of data , which is pretty efficient using this model

3.4 Precision Curve:

From the plot of accuracy we can see that the model has not over-learned the training dataset for the last few epochs. We can also see that the model is showing comparable skill on both datasets.



From the plot of loss, we can see that the model has comparable performance on both train and validation datasets (labelled test).

3.5 Precision and Calls

During information retrieval, precision is a measure of result relevancy, while recall is a measure of how many truly relevant results are returned

The learning curve shows the tradeoff between precision and recall for different threshold. A high area under the curve represents both high recall and high precision, where high precision relates to a low false positive rate, and high recall relates to a low false negative rate.

High scores for both (high recall and high precision) show that the classifier is returning accurate results (high precision), as well as returning a majority of all positive results (high recall).

In our data set the classification is shown as follows:

- precision (P) is calculated as follows:
    - P = Tp/Tp + Fp
- Recall (R) is defined as
    - R = Tp/Tp+fn
- F1 score is the harmonic mean, calculated as shown below:
    - f1-score= 2.(P X R) / P +R

For the Svm classifier:

	Precision (P)	Recall (R)	f1-score	support
Chinese	0.84	0.96	0.90	249
Gameofthrones	0.96	0.94	0.95	256
yoga	0.95	0.85	0.90	245

For the Random Forest Classifier:

	Precision (P)	Recall (R)	f1-score	support
chinese	0.86	0.95	0.90	130
gameofthrones	0.95	0.93	0.94	126
yoga	0.94	0.87	0.90	119

4. Conclusion:

Various binary classifier were to recognize which reddit the post belongs to.
From the results it can be observed that SVM slightly out performed the Random Forest Classification. Model has comparable performance on both train and validation datasets. Classifier returns the accurate result was seen.

5. References:

1. https://en.wikipedia.org/wiki/Reddit
2. https://www.cs.ubc.ca/~nando/540-2013/projects/p38.pdf
3.https://www.google.com/search?rlz=1C1RLNS_enIN788IN788&q=What+is+classification+in+machine+learning%3F&sa=X&ved=2ahUKEwj8_7rLm_bhAhVOn-AKHQ9pAd8Qzmd6BAgMEAk&biw=1189&bih=949
4. https://adventuresinmachinelearning.com/keras-lstm-tutorial/
5.https://engineering.upside.com/a-beginners-guide-to-optimizing-pandas-code-for-speed-c09ef2c6a4d6
6. https://www.datacamp.com/community/tutorials/wordcloud-python
7. https://praw.readthedocs.io/en/v3.6.0/
8. https://towardsdatascience.com/machine-learning-nlp-text-classification-using-scikit-learn-python-and-nltk-c52b92a7c73a

9. https://medium.com/machine-learning-101/chapter-5-random-forest-classifier-56dc7425c3e1

10 https://engineering.upside.com/a-beginners-guide-to-optimizing-pandas-code-for-speed-c09ef2c6a4d6