Classification of Reddit Posts using CNN & RNN with LSTM

Renjun Cheng
UC Berkeley School of Information
renjuncheng@ischool.berkeley.edu

Jacky Wong
UC Berkeley School of Information
jackymars@ischool.berkeley.edu

Abstract

The inherent nature of online forum content poses challenges to applications of language classification problems. Reddit is an online forum that displays a rich source of user-generated content. In this paper, we analyzed the contents of submissions on Reddit and built a recurrent neural network model to learn the type of posts from different sub-channels, which are called subreddits. To be specific, we used contents of 1 million posts from 40 most popular subreddits and fitted Convolutional Neural Networks and single-layer LSTM classifier models to predict the subreddit with a certain type of content. Our RNN classification algorithm performs quite well and achieves an average test accuracy of 46.1%. Our CNN classification performs slightly worse and achieves an average test accuracy of 40.24%. As you may know, Classification in 40 different categories is complicated. We tried applying the same model on the simplified task: classification in 10 subreddit instead of 40, the RNN model achieves a 76.8% accuracy. And the CNN model achieves a 73.54% accuracy.

1. Introduction

Reddit is a social news aggregation, web content rating, and discussion website. Registered members submit content to the site such as links, text posts, images, and videos, which are then voted up or down by other members. Posts are organized by topic into user-created channels called "subreddits", which cover a variety of topics such as news, politics, science, movies, video games, music, books, sports, fitness, and cooking[3]. Users may sometimes find it hard to categorize the posts after they finish writing the contents. A built-in recommendation system in Reddit that gives them advice on the subreddit where they might want to post would make things easier for the users. Thus, a classification algorithm that takes in the contents of the posts and outputs the suggested subreddit would be highly valuable, and deep learning approaches offer some way to achieve this goal.

For this project, we work on semantic analysis of Reddit post contents. Using a training dataset of 780,000 posts in 40 most popular subreddits, we design a RNN classification model with LSTM that is able to determine the subreddit a particular post is from by examining its content. The trained network is evaluated on a test set of 195,000 posts. With this method, an appropriate subreddit can be generated for the contents of a certain post, and this may offer insight for a recommendation system that can be built on Reddit to make smart suggestions for users on subreddit where they should post.

2. Background and Related Work

[1] Tyler, Rolland, and William tried to classify the subreddit based on the post's title and did not include any information about the content of the posts. They also only focused on the 10 hand-selected subreddit and that could potentially cause bias.

[2] Prasanna, Radhakrishnan, and Varun tried to perform Reddit post content analysis. They focused on sentiment analysis instead of content classification.

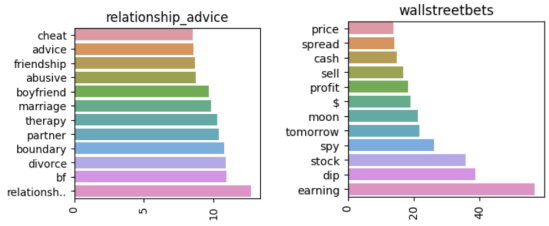
3. Methodology

3.1 Dataset

The dataset we use comes from Kaggle 1 million Reddit comments from 40 subreddits[4]. The dataset is an extract from a bigger reddit dataset (All reddit comments from May 2019, 157Gb or data uncompressed) that contains both more comments and more associated information (timestamps, author, flairs etc.). The dataset picked the first 25,000 comments for each of the 40 most frequented subreddits (May 2019), the volumes are balanced. The dataset excluded any removed comments / comments whose author got deleted and comments deemed too short (less than 4 tokens). The information kept here is: 1. Subreddit (categorical): on which subreddit the comment was posted; 2. Body (str): comment content; 3. Controversiality (binary): a reddit aggregated metric; 4. Score (scalar): upvotes minus downvotes.

3.2 Exploratory Data Analysis

Here we investigate the dataset by plotting the distributions of content words in two subreddits of interest: relationship_advice and wallstreetbets. Figure 1(a) shows that words "relationship_advice", "bf", "divorce", and "boundary" show up most frequently in the "relationship_adive" subreddit. Figure 1(b) shows that words "earning", "dip", and "stock" show up most frequently in the subreddit named "wallstreetbets".



(a) (b)

Figure 1: Distributions of Content Words in two subreddits: relationship_advice and wallstreetbets

4. Architect

4.1 RNN with LSTM

The Recurrent neural networks with Long Short-Term Memory Models (RNN with LSTM) extend the traditional recurrent neural network architecture and have been one of the most popular architectures for training language models. Specifically, most previous work has used the sequence-to-sequence approach to train models that are capable of generating textual output, either in the form of novel new phrases or in translation tasks [5][6][7].

The advantages of RNN with LSTM models over the traditional RNN models are that they can persist and discard information over long time sequences through the input gate and the forget gate ft. A cell graphically showing this equation structure is shown in figure 2. In classification tasks, the outputs of each LSTM cell have a linear transformation applied to them, followed by a softmax function in order to calculate the likelihood of a given outcome category[10].

For our classification task, our first approach is using a Recurrent Neural Network with LSTM. The original input of this model is a sequence of words in the content of the post, then the model converts the words to embeddings, and the embedded words serve as inputs to the LSTM cell. A subreddit prediction is generated at the end of the series of LSTM cells.

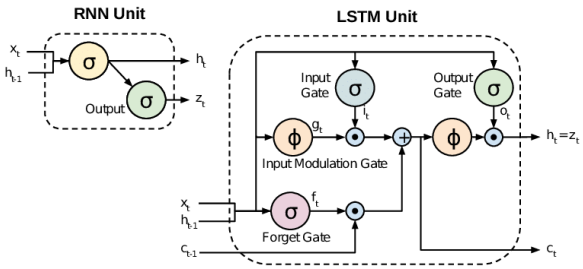
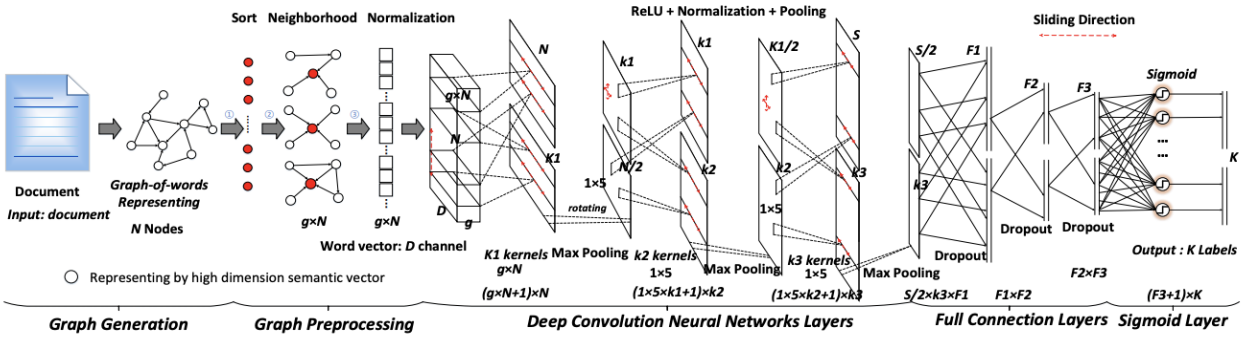


Figure 2: Graphical representation of the equations of an LSTM cell.

4.2 CNN

The Convolutional neural network is a class of deep, feed-forward artificial neural networks & uses a variation of multilayer perceptrons designed to require minimal preprocessing[8][9]. The original idea of CNN is inspired by the animal visual cortex. CNN is generally used in computer vision, however, they've recently been



applied to various NLP tasks and the results were promising.

Figure 3:Graphical representation of CNN in NLP

Our second approach to our classification task uses a convolutional neural network model to perform the classification. The Architect of CNN is shown in figure 3. We slide over input data convolution to extract features by applying a filter or kernel. In this feature extraction step, there are two key parameters associated with that sliding filter: how much input to take at once and by what extent the input should be overlapped. Stride here means the size of the step filter moves every instance of time. And filter count is the number of filters we want to use. After we apply the filter over input and have generated multiple feature maps, an activation function will be passed over the output to provide a non-linear relationship for our output. For our model, the activation function we choose is ReLU.

5. Evaluation

Our evaluation metric will be the classification accuracy for training and testing. Note that we use 80% of the 975,000 posts to train models, and 20% for final testing.

5.1 RNN with LSTM performance

For the RNN with LSTM model. We limit the dataset to the top 5,0000 words and set the max number of words in each post at 250. For predicting the subreddit origin for a post title we implemented a LSTM of length 250 and depth 1. This model contains 100-dimension hidden layers. During training, optimization is carried out over 10 epochs with a batch size of 128 posts. The model is trained on 70% of the 1,000,000 post content, with 10% of the posts

left for optimizing select hyper-parameters, and 20% for final testing. We determine the optimal dropout rate to be 0.2. Then we determine the optimal learning rate to be 0.01.

After fitting the model on the training data with adjusted hyper-parameters, we evaluate our model on the test data. Our model achieved a training accuracy of 52.63% and a test accuracy of 46.1%. The confusion matrix of the model predictions on the test set is shown in figure 4.

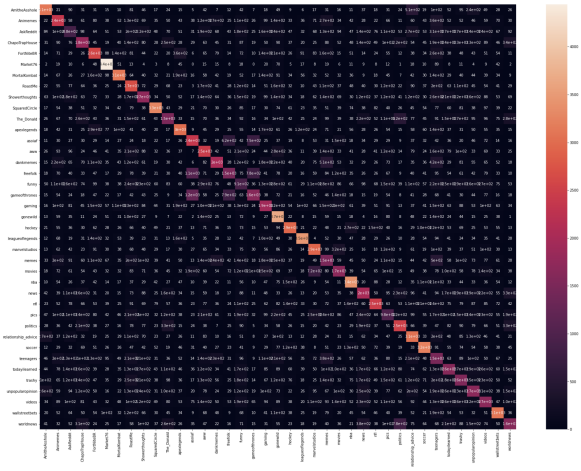


Figure 4: RNN with LSTM confusion matrix

46.1% is not a great accuracy at first look. But we need to consider the fact that we are classified 1 in 40 and the task is complicated. To see how it performs on less category classification. We used the same architect on task to classify the 10 subreddits instead of 40 subreddits.

We fit the model on 10 subreddits with highest accuracy. The model achieved 84.05% training accuracy and 76.80% testing accuracy. The confusion matrix of the model prediction on the test set is shown in figure 5.

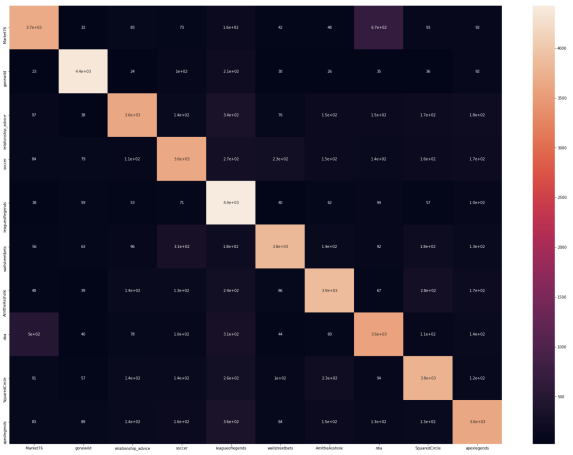


Figure 5: RNN with LSTM 10 highest accuracy subreddit classification confusion matrix

We fit the model on 10 subreddits with lowest accuracy. The model achieved a 63.58% training accuracy and 47.40% testing accuracy. The confusion matrix of the model prediction on the test set is shown in figure 6.

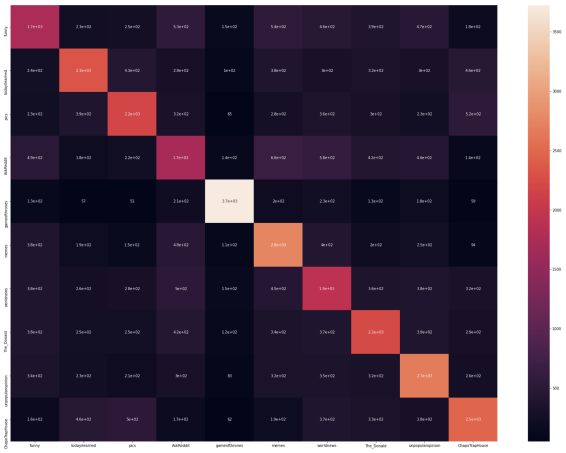


Figure 6: RNN with LSTM 10 lowest accuracy subreddit classification confusion matrix

5.2 CNN performance

For the CNN model, it first consists of an embedding layer in which we will find the embeddings of the top 5000 words into a 32 dimensional embedding and the input we can take in is defined as the maximum length of a review allowed. Then we add the convolutional layer and max-pooling layer. We flatten those matrices into vectors and add dense layers. The last Dense layer is having 40 as a parameter because we are trying to classify posts into 40 subreddits. We set our batch_size to 256 and trained for 20 epochs. After fitting the model on the training data with adjusted hyper-parameters, we evaluate our model on the test data. Our model achieved a training accuracy of 66.14% and a test accuracy of 40.24%. The confusion matrix of the model predictions on the test set is shown in figure 7.

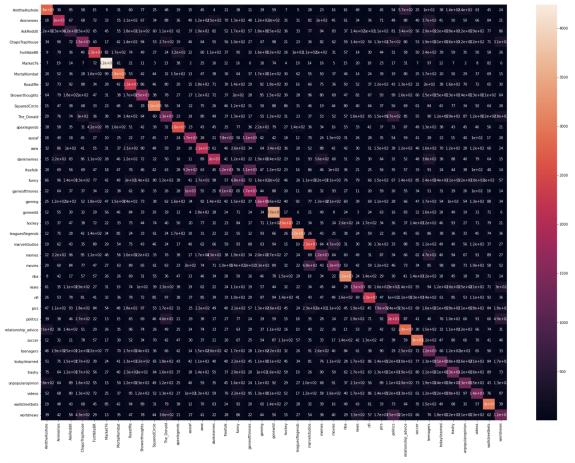


Figure 7: CNN confusion matrix

Similarly, we also performed on 10 highest accuracy and 10 lowest accuracy with the same architect. On 10 subreddit with highest accuracy. The model achieved 95.49% training accuracy and 73.54% testing accuracy. The confusion matrix of the model prediction on the test set is shown in figure 8.

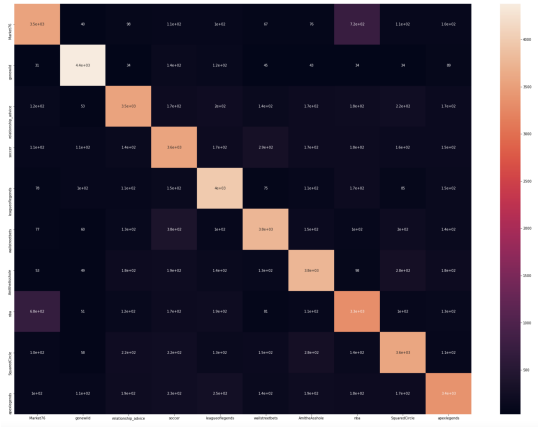


Figure 8: CNN 10 highest accuracy subreddit classification confusion matrix

On 10 subreddits with lowest accuracy. The model achieved 87.99% training accuracy and 41.14% testing accuracy. The confusion matrix of the model prediction on the test set is shown in figure 9

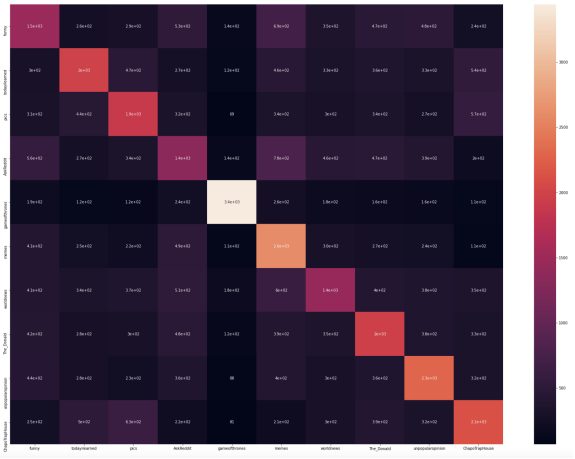


Figure 8: CNN 10 lowest accuracy subreddit classification confusion matrix

6. Conclusion

The performance of our model is disappointing at first look. However, considering the task is to classify one among 40 subreddits the performance is reasonably well. Especially for subreddit ‘news’ and ‘world news’.The posts under ‘news’ and ‘world news’ are similar but the model could classify them nicely. Moreover, when we try the 10 subreddit classification, the performance becomes much better. Overall speaking, both RNN with LSTM and CNN perform well on this task. In our case, the LSTM model performs better than CNN because the training time we spent on LSTM is also longer. To find which is the better architect for this specific task, we still need to do more research on this.

We would like to conduct follow-up research to enhance performance by trying word2vec and Glove as our Embedding. We could also perform a more robust hyperparameter tuning, we are unable to do that this time due to limitation of time, and the training process of both neural networks is slow without GPU boosting. To have more time on training with more epoch and less batch size would also be helpful.

7. References

[1] Tyler Chase, Rolland He, and William Qiu. “Deep Classification and Generation of Reddit

Post Titles” 2017.url:https://web.stanford.edu/class/archive/cs/cs224n/cs224n.1174/reports/2762088.pdf

[2] Prasanna Chandramouli,Radhakrishnan Moni,and Varun Elango “Predicting reddit post popularity” 2017 url:https://github.com/radkrish91/Predicting-Reddit-Post-Popularity

[3] Jake Widman “What is Reddit” March 29 2021 url:https://www.digitaltrends.com/web/what-is-reddit/

[4] Samuel Magnan “1 million Reddit comments from 40 subreddits” 2020 url:https://www.kaggle.com/smagnan/1-million-reddit-comments-from-40-subreddits

[5] Ilya Sutskever, Oriol Vinyals, and Quoc V Le. “Sequence to sequence learning with neural networks”. In: Advances in neural information processing systems. 2014, pp. 3104–3112.

[6] A. Graves, A. Mohamed and G. Hinton, "Speech recognition with deep recurrent neural networks," *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*, Vancouver, BC, Canada, 2013, pp. 6645-6649, doi: 10.1109/ICASSP.2013.6638947.

[7] Sherstinsky, Alex. (2020). Fundamentals of Recurrent Neural Network (RNN) and Long Short-Term Memory (LSTM) network. Physica D: Nonlinear Phenomena. 404. 132306. 10.1016/j.physd.2019.132306.

[8] Asgari-Chenaghlu, Meysam. (2018). Convolutional Neural Networks for Natural Language Processing.

[9] W. Wang and J. Gang, "Application of Convolutional Neural Network in Natural Language Processing," *2018 International Conference on Information Systems and Computer Aided Education (ICISCAE)*, Changchun, China, 2018, pp. 64-70, doi: 10.1109/ICISCAE.2018.8666928.

[10] X. Zhang, M. H. Chen and Y. Qin, "NLP-QA Framework Based on LSTM-RNN," 2018 2nd International Conference on Data Science and Business Analytics (ICDSBA), Changsha, China, 2018, pp. 307-311, doi: 10.1109/ICDSBA.2018.00065.