CS 510 Project: Classifying Short Text Reddit Posts

Mithila Guha, Colin Murphy & Angela Mastrianni

Abstract

A common problem faced by Reddit users is that they may post in a wrong subreddit or be unsure of the correct subreddit for their post. As a step in addressing this issue, we developed classifiers that would identify the appropriate subreddit for a post. These classifiers were designed for short text posts that are less than 200 characters in length. We explored multiple supervised learning methods in this project and established a Multinomial Naive Bayes Classifier, a Support Vector Machine (SVM) classifier and an Artificial Neural Network to classify short text Reddit posts. When we compared their performances, the neural network classifier performed better than both the Naive Bayes classifier and support vector machine classifier in classifying short text posts. In addition, including authorship data also improved classification performance.

Introduction

Reddit is a popular social media website where users can post in different groups, known as subreddits. These groups maintain a content theme, and it is important for the users to strategically target relevant subreddits to post their contents. A well-designed classifier could be used in a process that would notify Reddit users if there is a better subreddit for their content themes. We are proposing classifiers for short text Reddit posts that are less than 200 characters in length. However, the classification of "short" text is particularly challenging due to the sparse representation of the unique, topical languages being used. To address this problem, we explored different supervised learning methods.

In this project, we have designed a Multinomial Naive Bayes Classifier, a Support Vector Machine (SVM) classifier and an Artificial Neural Network for classifying the short text posts. Additionally, we included authorship data as input to the classifiers as a potential method to improve their performance. We can evaluate the performance of our classifier by comparing the results for particular posts to their actual group on Reddit.

Our project is intended to make a research contribution by proposing improved classifiers that can identify appropriate subreddits for users' short Reddit posts. Often Reddit users are not aware of all the different groups on Reddit. This work could minimize their work and help notify them if there is a better group for their posts before posting them to a particular group.

Related Work

Online social networks (OSNs) collect user-generated data in an unstructured and non-annotated format. However, to facilitate narrowly focused and personalized contents to the users on such OSNs, it is important to be able to organize these unstructured user-generated content into clusters of related posts with similar topics.

A significant amount of research has been done on the clustering methods to generate automatic annotation of topics within these OSNs to better facilitate user preferences within the platform. However, most of the methods fall apart due to the challenges of scalability, high dimensionality, and accuracy in generating meaningful cluster labels. To solve this challenge, the study of Bisht, Paul (2013) demonstrated the Hierarchical, Partitioning and Frequent-item sets based methods to propose dimensionality reduction mechanisms and improve cluster quality measurement.

On the other hand, Li, Shah, Liu, & Nourbakhsh (2017) proposed neural word embeddings on short and noisy social media posts with unique lexical and semantic features. In their paper, they introduced doc2vec, wtd word2vec, TF-IDF and LDA for feature representation. However, they concluded that the doc2vec model with k-means clustering produced the best performance on the latent label hashtags for Twitter data set and the Reddit data sets.

Another active research area is Topic Modelling which is a machine learning method for discovering the abstract "topics" that occur in a collection of documents. Blei, David M. et al (2010) explained topics in probabilistic topic models as a mixture of words, where documents are modeled as mixtures of topics. Some of the frequently used topic models with applications on OSN text data are: Latent Dirichlet Alloca-

tion (LDA)(Blei, Ng, & Jordan (2003)), Dynamic Topic Models etc. which discover topics over time (Alghamdi & Alfalqi (2015)).

Recently, Stephan A. Curiskis, et al. (2019) experimented with using different feature selection and clustering methods to evaluate their performance with topic modelling with Reddit and Twitter data. They used a tf-idf matrix, word2vec neural network model and doc2vec neural network model to create the different feature representations. Their different clustering methods included a k-means clustering algorithm, a k-medoids clustering algorithm, a hierarchical agglomerative clustering algorithm, and a Non-negative Matrix Factorisation (NMF) algorithm. They found that when using the tf-idf model for feature representation, the clustering methods performed badly when posts were less than 200 characters in length. Additionally, Hong and Davison (2010) proposed that aggregating shorter text posts by author could improve topic modelling results as authors frequently write about the same topic.

Input variable selection is also important when training artificial neural networks. May, Dandy & Maier claim that "ANN models are too often developed without due consideration given to the effect that the choice of input variables has on model complexity, learning difficulty, and performance of the subsequently trained ANN" (May, Dandy, and Maier 2011, p. 19). According to them, good input variables are not only highly informative but also provide different information than other input variables. Too many redundant or irrelevant input variables can complicate or slow down training. Additionally, careful input data selection can help with the comprehensibility of the artificial neural network model. Forward selection is a search algorithm that can be used for input variable selection. it starts by training the neural network with the most single relevant input. It then iteratively retrains the neural network, each time adding the next most relevant input variable until the accuracy of the neural network is no longer improved or is diminished by the addition of the new input.

Approach

We are drawing inspiration from the work done on Reddit topic modelling by Stephan A. Curiskis, et al. (2019). However, they derived some of their feature representations from word2vec and doc2vec, which used neural networks and then employed different clustering methods for the topic modelling. On the other hand, we will employ the common tf-idf for the feature representation of the text and use neural networks to classify the posts into different topics. We have also evaluated the performance of using authorship data as part of the input classifcation models.

Data Collection

We collected our data by scraping Reddit posts using the PRAW API (Bryce 2012). We collected posts that were less than 200 characters in length from five subreddits about universities in Pennslyvania: r/temple, r/drexel, r/upenn, r/villanova and r/cmu. As these subreddits are all focused on universities, their posts will have similar content discussion. We also collected posts that were less than 200 characters in length from five larger subreddits: r/investing, r/nfl, r/help, r/thingsmykidsaid and r/AskNYC. Since these subreddits focus on different topics, their posts will likely have few similarities in content. The size of our data was limited by the number of short length posts in these subreddits. For each author of a post, we also collected all of the subreddits that they have posted in before their current post. The number of posts collected from each subreddit is shown in tables 1 and 2.

Subreddit	Number of Posts
CMU	332
Drexel	367
Temple	365
UPenn	394
Villanova	117

Table 1: Number of Posts in University Group Subreddits

Subreddit	Number of Posts
Investing	217
AskNYC	251
ThingsMyKidSaid	397
Nfl	78
Help	398

Table 2: Number of Posts in Non-University Group Subreddits

To split our data into training and testing data, we used stratified random sampling to achieve sets of data where the labels are well proportioned, as used by Mathur and Foody (2008) in their work with multi-class SVM classification. We also used tf-idf text vectorization, as prior work showed that tf-idf performed better than word count vectorization (Kibriya et al. 2004).

Binary Classification

With the assumption that each of the Reddit communities (i.e. university subreddits and non-university subreddits) can be most easily classified by our model when they are represented by the corpus of the entire community, we have designed a binary classification neural network. A simple binary classification label vector for the dataset was created to classify the post as a "university" or "non-university" subreddit.

A relatively simple artificial neural network was implemented (MLP with a single hidden layer) with the tf-idf vectorization of the training dataset being used. The simplicity of this model's output provides proof of

concept that the short text provided by these posts can still be used to distinguish between the two communities.

Multi-Classification

For the multi-classification of posts into their subreddits, we classified the data in the university subreddits and the data in the non-university subreddits separately. Our hypothesis was that it would be harder to classify the posts in the university subreddit group using just text features since these posts would be more likely to contain similar content across the different subreddits (questions about finals, professors, etc). In both cases, the labels were the name of the subreddit: temple, drexel, cmu, upenn and villanova in the university group and investing, nfl, help, thingsmykidsaid and AskNYC in the non-university group.

Three different supervised learning methods were used to classify the posts: a naive bayes classifier, a support vector machine (SVM) classifier and an artificial neural network. We used a multinomial naive bayes classifier as it has been shown to have better performance than multivariate naive bayes classifiers in text classification problems (Kibriya et al. 2004). We also choose to use a support vector machine classifier as they have been shown to have a higher classification accuracy than naive bayes classifiers (Kibriya et al. 2004). Additionally, we also used neural networks to classify the reddit posts.

Subreddit	% Of Posts By Repeat Author
CMU	46.7%
Drexel	60.8%
Temple	63.0%
UPenn	66.5%
Villanova	10.3%
Investing	31.3%
AskNYC	42.6%
ThingsMyKidSaid	20.7%
NFL	70.23%
Help	11.3%

Table 3: Percentage of Posts in Each Subreddit Whose Author Had Posted in the Subreddit Before

To represent authorship, we looked at the posting history of the user of each post. We added five additional boolean inputs to represent each subreddit. If a user had posted in that subreddit before the current post, we set the input to true. Otherwise, that input was false. Table three shows the percentage of posts in each subreddit whose author had previously posted in that subreddit before writing their current post. In both the university and non-university groups, there were authors of posts who had posted in multiple of the subreddits in that group before.

Experimental Approach

Experimental Setup

Various architectures for feedforward neural network were explored to improve the general capacity of each model. Additionally, systematic grid search was performed to further tune the model's hyperparameters and minimize overfitting. Further discussion of model optimization and choice of model functions can be found in the appendix.

Naïve Bayes and SVM models were implemented as baseline comparison to that of the neural network and were carried out with the available implementations of sklearn.

Performance Evaluation

All models described in this paper were evaluated by 5-fold stratified cross validation to provide a representative performance metric. Further verification of the mode was performed by a verification test set unseen by the model.

Findings & Discussion

Binary Classification

The binary classification model offers a moderate separation of the two Reddit communities - 83%. The performance of the model to distinguish between these may be attributed to its ability to identify the presence of a university subreddit - this biases is evident in the confusion matrix of Fig. 1. as the model has a greater recall metric for university subreddits.

Investigating features of the 15% of mislabelled posts allows for a better understanding of the chosen input data's impact on the model. Of the original post text scraped from Reddit, each included a title of which was not included in the main corpus used for training. It was noticed that many of the posts lacked context or cohesive narrative. When viewing the original post, without filtered features, we found that the post text was sometimes a continuation of its title - this is a common aspect of social forums, such as Reddit, where a "stream of consciousness" method is used, such that the title of the thread and the body text are of a continuous thought or sentence. One example that we saw when investigating the misclassified posts was a post whose main content was "I am not ashamed whatsoever". The content of the post does not provide much insight into the appropriate subreddit. However, the title, "My first written words were 'Super Poops'", provides a better indication that it belongs to the 'thingsmykidsaid' group.

Additional sources of error in such classification schemes arise from the validity of the ground truth data provided to the supervised learning technique. The assumption being made by the training set to the model is that the training set consists of correctly labeled inputs; however, it can be understood that the posts themselves may be "mislabeled" and contain discussions of topics that are not relevant to the subreddit.

Evaluation	Accuracy
Stratified kfolds training	83% (±1%)
Final Model Verification	85%

Table 4: Binary Classification Neural Network Results

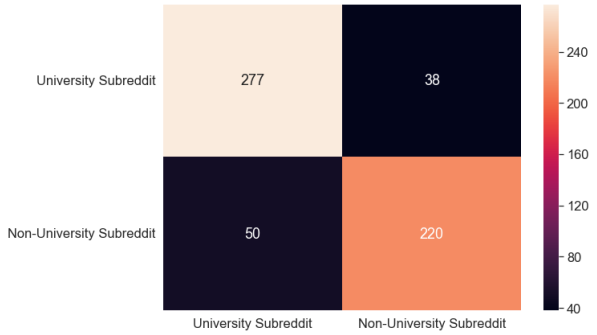

Figure 1: Binary Classification Neural Network Confusion

Subreddits	Method	Accuracy
University	Naive Bayes	46% (±1%)
University	SVM	44% (±1%)
University	ANN	47% (±1%)
Non-University	Naive Bayes	68% (±1%)
Non-University	SVM	73% (±1%)
Non-University	ANN	76% (±1%)

Table 5: Multi Classification Results Without Authorship Information

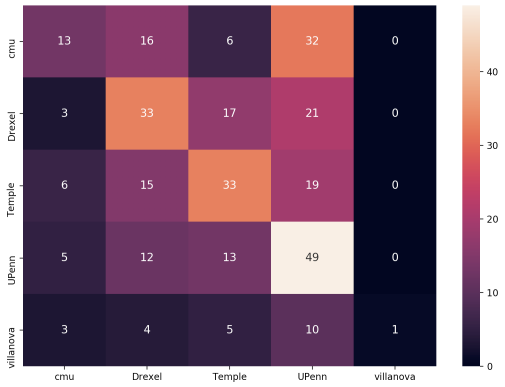

Figure 2: University Subreddits- Naive Bayes Confusion Matrix Without Authorship

Multi-Classification

The first table in this section shows the results of the three different classification methods with just the tf-idf representation of the text as input. As shown by the results, the neural network had the highest accuracy in both groups, although the accuracies were fairly similar across the three classifiers. The first three figures in this section show the various confusion matrices for the different classification methods. As shown by the matrix, the Naive Bayes classifier only classified one of the Villanova posts correctly while the SVM and neural network classifiers were able to correctly classify some more posts in this group. There were fewer posts from the Villanova subreddit in the training and testing data which could explain why the Naive Bayes classifier had a difficult time correctly classifying these posts. As expected, we see better performance when classifying posts in the non-university subreddit group as compared to the university subreddit group. This is aligned with our hypothesis as the non-university groups focus on completely different topics and will have different vocabularies between them while the posts in the university group subreddits will mostly focus on topics related to college and share a similar vocabulary.

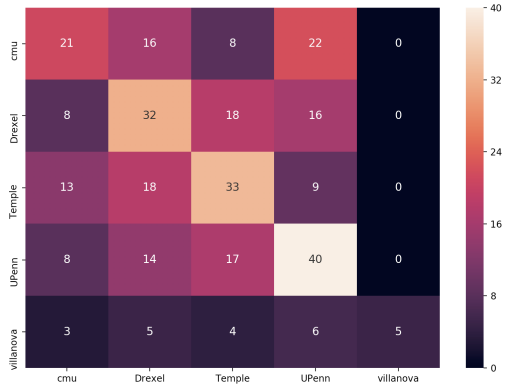

Figure 3: University Subreddits- SVM Confusion Matrix Without Authorship

The last table in this section shows the results of the three different classification methods with the tf-idf representation of the text and authorship information as inputs. Both the results table and the confusion matrices illustrate that adding authorship as a feature improved the classification of the posts for all three methods. In both groups of subreddits, the artificial neural network had the highest accuracy. The multi-classifier faced some of the same data issues that were mentioned

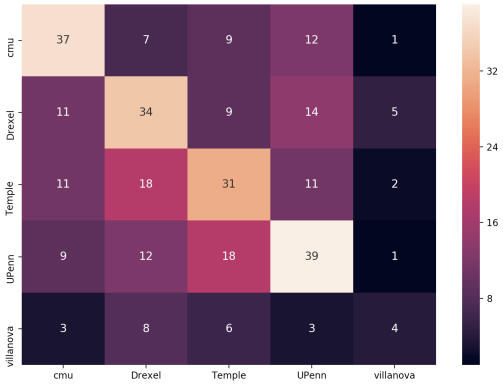

Figure 4: University Subreddits- Neural Network Confusion Matrix Without Authorship

in the binary-classification section, as the ground truth labels for training and testing were based on the current subreddits of the posts, even if that was not the correct subreddit for the post.

Subreddits	Method	Accuracy
University	Naive Bayes	74% (±1%)
University	SVM	73% (±1%)
University	ANN	75% (±1%)
Non-University	Naive Bayes	78% (±4%)
Non-University	SVM	80% (±2%)
Non-University	ANN	83% (±2%)

Table 6: Multi Classification Results With Authorship Information

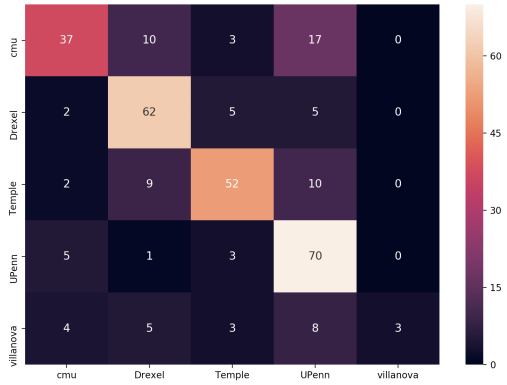

Figure 5: University Subreddits- Naive Bayes Confusion Matrix With Authorship

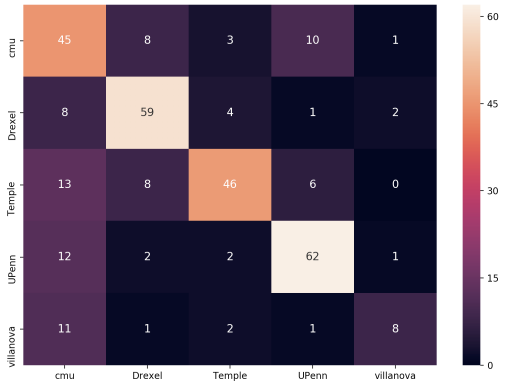

Figure 6: University Subreddits- SVM Confusion Matrix With Authorship

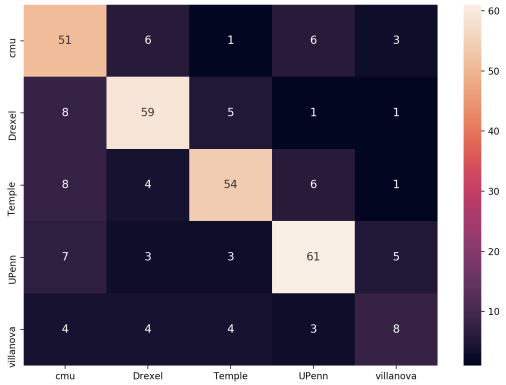

Figure 7: University Subreddits- Neural Network Confusion Matrix With Authorship

Conclusion

In this work, we used artificial neural networks to classify Reddit posts that were less than 200 characters in length. To accomplish this, we collected posts from two groups of subreddits. One group was subreddits which focused on universities in Pennsylvania. The second group contained subreddits that focused on completely different topics, such as the NFL and investing. We first performed a binary classification to classify all posts as belonging to either the university or non-university subreddit group. After tuning the parameters and neural network structure, our binary classifier had an accuracy of 85%.

We then created a multi-classifier that would classify posts into a particular subreddit. We ran this multi-classifier twice - once on the university subreddit group and once on the non-university subreddit group. When the input was only features from the post's text, the classifier achieved an accuracy of 47% in the university group and an accuracy of 76% in the non-university group. As expected, the classifier performed better in

the non-university group because the contents of the posts in that group varied as the subreddits all focused on different topics, whereas the subreddits in the university group focused on similar topics. We compared the results of the neural network classifier to the results of a Naive Bayes classifier and a support vector machine classifier, finding that the neural network classifier performed slightly better than the others.

Finally, we added information about the author to the multi-classifier to investigate authorship features effect on the classifier's performance. However, were limited by the amount of information that Reddit provides about its users as we could only obtain the post history of each user. We added an input for each subreddit, which indicated if the author of the post had posted in that subreddit before writing the current post. Adding these features improved the performance of our classifier in both groups. The accuracy in the university group was 75% while the accuracy in the non-university group was 83%. We also compared these results from the neural network to the results of a Naive Bayes classifier and a support vector machine classifier, finding that the neural network classifier performed slightly better than others.

There are several opportunities for future work. Within the realm of social media there tends to be a greater use of abbreviations and slang words of which may be overlooked by the simple "Bag-Of-Words" feature extraction used here. Consideration of these domain specific features may improve the identification of such short texts (Sriram et al. 2010). Furthermore, the multi-classification model we've designed may employ the binary classifier as an input - the current iteration of the multi-classification model is trained on the two Reddit communities being separated corpora.

Future work could also apply our classifiers to a different data set that contains more information about the authors of texts, such as their nationalities or ages, to see if those features improve the performance of the classifier.

Appendix

Model Optimization

The initial models applied simple ReLU activation functions for layers to quickly train the models for a baseline metric.

Further model optimization was performed using grid search technique to explore the various hyperparameters of the neural network. This search focused on the affects of kernel initializers and optimizer functions that might offer marginal increases in the models performance - for our binary classifier, we found that a RMS Propagation optimizer model performed consistently better than that of the Adam optimizer (2% accuracy increase).

Tuning of the appropriate quantity of epochs and batch sizes were performed to minimize over-fitting of the model. During the training, the validation loss was monitored to allow for early stopping of the training as the model began to converge - finding that the binary classifier was capable of reaching a loss minimum with a relatively small epoch and batch size (10 and 32, respectively). However, the multi-classifier performed best with 100 epochs, batch size of 10 and a softmax activation function. Both models applied cross-entropy loss functions. In both models, we experimented with four different activation functions: softmax, softplus, sigmoid and hard sigmoid. In the binary classifier, sigmoid offered the best performance while softmax performed the best in the multi-classifier.

We also evaluated four different kernal types for the Support Vector Machine: linear, sigmoid, rbf and poly. After experimenting with the different parameters for the multi-classification problem, we found the best performance with a sigmoid kernel, a C-value of 100 and a gamma value of 0.01.

References

Alghamdi, R., and Alfalqi, K. 2015. A survey of topic modeling in text mining. *International Journal of Advanced Computer Science and Applications* 6(1).

Bisht, S., and Paul, A. 2013. Article: Document clustering: A review. *International Journal of Computer Applications* 73(11):26–33. Full text available.

Blei, D. M.; Carin, L.; and Dunson, D. B. 2010. Probabilistic topic models. *IEEE Signal Processing Magazine* 27:55–65.

Blei, D. M.; Ng, A. Y.; and Jordan, M. I. 2003. Latent dirichlet allocation. *Journal of Machine Learning Research* 3:993–1022.

Bryce, B. 2012. Praw: The python reddit api wrappe. https://github.com/praw-dev/praw/.

Curiskis, S. A.; Drake, B.; Osborn, T. R.; and Kennedy, P. J. 2019. An evaluation of document clustering and topic modelling in two online social networks: Twitter and reddit. *Information Processing and Management.*

Hong, L., and Davison, B. D. 2010. Empirical study of topic modeling in twitter. In *Proceedings of the first workshop on social media analytics*, 80–88. acm.

Kibriya, A. M.; Frank, E.; Pfahringer, B.; and Holmes, G. 2004. Multinomial naive bayes for text categorization revisited. In *Australasian Joint Conference on Artificial Intelligence*, 488–499. Springer.

Li, Q.; Shah, S.; Liu, X.; and Nourbakhsh, A. 2017. Data sets: Word embeddings learned from tweets and general data.

Mathur, A., and Foody, G. M. 2008. Multiclass and binary svm classification: Implications for training and classification users. *IEEE Geoscience and remote sensing letters* 5(2):241–245.

May, R.; Dandy, G.; and Maier, H. 2011. Review of input variable selection methods for artificial neural networks. *Artificial neural networks-methodological advances and biomedical applications* 10:16004.

Sriram, B.; Fuhry, D.; Demir, E.; Ferhatosmanoglu, H.; and Demirbas, M. 2010. Short text classification in twitter to improve information filtering. In *Proceedings of the 33rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '10, 841–842. New York, NY, USA: ACM.