Text Classification for University Reddit Communities

Garrison Hess, Xinyuan Lu, and Gaurav Nakum

December 2, 2019

1 Introduction

Universities throughout the world form communities on Reddit. In these communities, known as subreddits, students discuss various aspects of university life. These aspects often include community, culture, academics, and more. Each university subreddit has its own unique characteristics. These characteristics comprise pieces of what makes each student body unique. To explore these distinct characteristics, we collected 78,736 posts and 997,250 comments from the 20 largest university subreddits. Using the comments of each post, we predict the source university. Hence, predictions rely solely on the communication within the university community. Through this inquiry, we seek to understand to what extent Reddit comment patterns can be used to predict source universities.

2 Literature Review

Several studies have investigated various aspects of content and communities on Reddit. Community models were leveraged to help predict re-post popularity in McAuley, Leskovec, and Lakkaraju's 2012 paper. This paper used 132,307 image-based posts dating back to 2008. Gjurkovic and Snajder (2018) analyzed Reddit user behavior in the context of personality classification. This paper used the MBTI9k dataset, which is a corpus of reddit comments and posts labeled with personality types. Gjurkovic and Snajder's work relates relates to ours because each community has characteristics that can be understood as an aggregate personality.

Multiple studies have also explored subreddit classification using text. Tam's work, Classifying Reddit Comments by Subreddit, approached the problem from a deep learning sequence classification perspective. This paper used data made available by Jason Baumgartner's pushshift.io, and filtered the data to include precisely 20 distinct subreddits. Tam concluded that a key limiting factor to performance was training time. In addition, Tam suggested that using sentiment via the Stanford Tree Bank could have been helpful.

Giel, NeCamp, and Kader (2014) used a total of 12,000 posts from 12 distinct subreddits, collected in August 2013. Of all the previous work, this is the most similar to ours. They used Naive Bayes and Logistic Regression, and generated features using bags of words, TF-IDF, and Latent Dirichlet Allocation. We were not aware of this work when first developing our project, but have found their work to be a useful reference. A few key distinct aspects of our work include: collecting significantly more data, feature engineering with Word2Vec, and conducting the analysis in the context of university subreddits. Finally, Gutman and Nam (2015) performed subreddit prediction for 5 subreddits, using a Kaggle dataset from May 2015. For feature engineering, they use Bag of Words, Word2Vec, and Latent Semantic Analysis. In addition, they Naïve Bayes, Logistic Regression, AdaBoost, and Support Vector Machines as their classifiers. Each of these papers provide valuable points of comparison for our work.

Despite the similarities between our work and the previous subreddit classification papers, separating 20 universities is intrinsically different than separating unrelated subreddits. The language within university subreddits is highly similar, when compared to the differences between a Baseball subreddit and a Game of Thrones subreddit. Our results certainly reflect the difficulty of the task. Whereas Giel, NeCamp, and Kader observe a logistic regression baseline accuracy of 67%, our baseline accuracy is closer to 30%. This is a function of both the similarity of university sub-

reddits, and the fact that we predict over 20 classes. Finally, Giel et al. observe that their best model is a Bag of Words, with the comment count length for each post. We observe similar results, except that comment lengths were useless in our task. This is because in significantly different communities, comment lengths vary widely. However in university communities, comment lengths are highly similar.

3 Predictive Modeling

Our predictive modeling process includes exploratory data analysis, feature engineering, and model selection using various performance measures.

3.1 Dataset

We used the Reddit API to collect 78,736 posts and 997,250 comments, from the 20 largest university reddit communities. Each of these subreddits have over 15,000 members, which ensures a reasonable level of university representation. We used the Public Reddit API Wrapper (PRAW) to access the data. We collected the maximum amount of data possible through the public API.

There are several groupings of posts in the Reddit API: hot, gilded, controversial, new, and top. In addition, several of these groupings can be accessed for time periods including: day, month, year, all-time. The limitation was imposed by a 1000 post limit on any given API call. Hence, we performed API calls for all possible combinations of post groupings, and time periods. This required deduplication of repeated posts. Accordingly, our posts date back to the beginning of each university community. However, we are able to get more new posts than old, so there is a small bias in this capacity. We have no reason to expect this small bias to affect modeling in any significant way. In fact, the recency of the data is in some way a positive, because it reflects more recent trends.

3.2 Exploratory Data Analysis

We explored the data using various plots and summary statistics throughout the process. We include some of the summary statistics and visualizations here. These figures provide evidence for our feature selection, which is described below.

University Subreddit	Subs	Posts	Comments	Comments per Post
UCSD	30,412	3789	40,652	10
Aggies	25,219	3902	48,185	12
ASU	18,890	3564	33,559	9
Berkeley	30,189	3981	58,559	14
gatech	23,365	3899	51,946	13
NYU	31,719	3240	19,311	5
OSU	26,735	3882	40,722	10
Purdue	23,586	3746	41,530	11
RIT	15,002	3797	42,039	11
Rutgers	23,664	4224	41,057	9
UBC	29,687	3789	73,433	19
UCF	38,888	4049	50,845	12
UCLA	21,142	3923	35,227	8
UIUC	38,077	4229	62,908	14
UMD	20,466	3896	35,795	9
UofM	16,304	3741	37,790	10
UofT	38,681	4357	88,949	20
UTAustin	24,922	3904	38,563	9
UWaterloo	46,270	4860	114,174	23
VirginiaTech	17,626	3675	42,496	11

In the above table, we present various statistics for the university subreddits. We can see the size of the data we are working with clearly here. The fewest comments for a given subreddit is 19,311 (NYU), and the greatest is 114,174 (UWaterloo).

When looking at the "Comments per Post" column, a clear trend emerges among Canadian schools. That is, they comment at a much higher rate than U.S. schools. This is pertinent because for each additional comment within a post, we have additional data to use in prediction. Conversely, NYU is the least talkative school - with 5 comments per post.

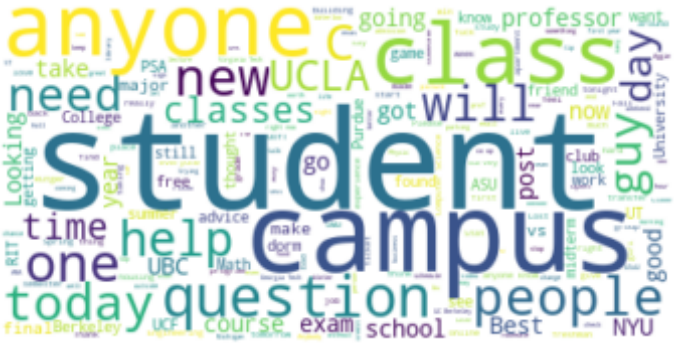

Figure 1: Overall Word Cloud

Figure 1 presents the word cloud for all the comments across all the university subreddits. It matches expectations of a general college community word cloud. It is of note that NYU and UCLA appear in the general word cloud. Apparently they are mentioned more than the other schools in our comment data.

Figure 2: /r/UCSD Word Cloud

The UCSD word cloud is where things get interesting. We see clear patterns emerge, as compared to the general university word cloud. That is, "Geisel", "UCSD", and "CSE" are all big topics of discussion. For anyone who visits the UCSD subreddit, this certainly matches intuition. More importantly, this provides us with concrete evidence that there is at least some distinctness in the UCSD comments, as compared to the rest of the universities. However, the words "student", "anyone", and "campus" remain at the front in both visualizations. This speaks to the similarities between the schools, and our prediction classes, which makes our task significantly more difficult.

3.3 Feature Selection

From the data we collected, we received all post and comment metadata. This included time of post, upvotes, ratio of upvotes to downvotes, author identifier, and various other data. For the purpose of this analysis, we limit ourselves to the body of comments for each post. As discussed above, we are interested in the way text communication differentiates each university community. We considered using the title data, but excluded it because it lacks the communication aspect that comments have. In addition, our word clouds show that UCSD has somewhat distinct comments compared to the overall pool. This provides support for the idea that we can, to some extent, predict the university from the comments.

3.4 Feature Engineering

Since we are strictly using comments as our features, we employ several feature engineering techniques to the comment data. These techniques are canonical methods for working with text data, and they include: Bag of Words, Term Frequency - Inverse Document Frequency (TF-IDF), Word2Vec, and Latent Dirichlet Allocation.

3.4.1 Bag of Words

We use bag of words in a typical way, tokenizing comments and loading them into document vectors. To avoid overfitting and excessive computational intensity, we use the top 1000 words from all the comments.

3.4.2 TF-IDF

We apply the Term Frequency-Inverse Document Frequency (TF-IDF) transformation to the comment data, such that the weights of each word will better reflect their rarity. This is essentially a bag of words which accounts better for rarity of terms.

3.4.3 Word2Vec

Applying Word2Vec was a particularly interesting endeavor. We trained our own model on the comment corpus, and also used Google's pre-trained model. Interestingly, Google's general model significantly outperformed our specific local model. When testing various word similarities, we got solid and reliable results. For example, "UCSD" and "UCLA" returns a similarity of .62, whereas "UCSD" and "Test" returns a similarity of .05. In addition, in the model trained strictly on our document corpus, "UCSD" and "UCLA" returns .79, whereas "UCSD" and "Test" returns .38. This difference is highly important, because it shows the locally trained model's weakness in generalization. Whereas it sees UCSD and UCLA as highly similar, the difference between a good similarity and a bad similarity is highly obscured. In Google's model, the truly similar comparison is over 12x as similar as the unrelated comparison.

Conversely, in our locally trained model the truly similar comparison is only 2x as similar as the unrelated comparison.

3.4.4 Latent Dirichlet Allocation

The Latent Dirichlet Allocation (LDA) is a generative model for topic-modeling text data. We apply it to the comments, taking the top 100 most likely topics for each document. When we observe the dominant topics in the full document corpus, clear patterns emerge. Specifically, we observe that the LDA finds topics such as: majors, class performance, finances and tuition, social life, dating life, sports, and admission. These topics are central to university life, and show us that the LDA is effectively extracting the comment data topics.

3.4.5 Train-Validation-Test Split

We allocate 80% of our data to training, 10% to validation, and 10% to a hold-out test set.

3.4.6 Performance Metrics

We use various performance metrics to evaluate our multi-class classifier. Specifically, we use the accuracy, precision, recall, and F1-score. Since this is a multi-class problem, we compute precision, recall, and F1 as the weighted averages across each of the classes. In addition, we visualize our performance using a confusion matrix.

3.4.7 Relevant Classification Models

Our prediction task is a somewhat high-dimensional multi-class classification problem. This means that we can use any typical multi-class classifier, so long as it does not fall to the curse of dimensionality. Some of these classifiers include: Logistic Regression, SVM, Naive Bayes, K-Nearest Neighbors, Multi-layer Perceptron, Decision Trees, Random Forest, AdaBoost, and Gradient Boosted Decision Trees (XGBoost). We avoided KNN and SVM due to the size and dimensionality of our data. Ultimately, we selected Logistic Regression, because we are primarily interested in feature engineering for the comment data.

4 Results

4.1 Training Performance Metrics by Feature Representation

The following table shows the training performance metrics by model. A logistic regression was used as the classifier for each of these models.

	Acc.	Precision	Recall	F1
BoW	0.493	0.541	0.493	0.502
Word2Vec	0.337	0.3486	0.337	0.335
BoW+W2V	0.544	0.569	0.544	0.550
LDA+BoW	0.235	0.243	0.235	0.229
LDA+W2V	0.354	0.367	0.354	0.352
LDA+TF-IDF	0.165	0.171	0.165	0.158

4.2 Test Performance Metrics by Feature Representation

The following table shows the test performance metrics by model. A logistic regression was used as the classifier for each of these models. As we can see from the train performance to test performance, there is no significant overfitting issues with any of the models.

	Acc.	Precision	Recall	F1
BoW	0.407	0.446	0.407	0.414
Word2Vec	0.319	0.328	0.319	0.316
BoW+W2V	0.429	0.446	0.429	0.432
LDA+BoW	0.231	0.236	0.231	0.226
LDA+W2V	0.348	0.361	0.348	0.347
LDA+TF-IDF	0.157	0.160	0.157	0.149

4.3 Feature Representations

The models presented in the performance tables were selected by testing various combinations on the validation set, and showing which performed best. In addition, the performance tables show strong variation between models using different feature representations. The bag of words model performs impressively. On the other hand, the LDA models perform poorly. This was especially surprising given the highly intuitive outputs

it was providing. However, it may make some sense in the context of topics being too general to differentiate between specific schools. The LDA with TF-IDF model performs especially poorly, which could be explained by our lack of hyperparameter tuning. Finally, the Word2Vec model performs well on its own, which explains the combined performance of it with bag of words.

4.4 Model Selection

Throughout each of the test performance metrics, the bag of words representation dominates. However, the best model uses bag of words in conjunction with Word2Vec. As mentioned previously, the bag of words model uses the top 1000 most used words. This simple bag of words model would not be a bad choice, as it is less complex than the model including Word2Vec. However, Google's Word2Vec is an highly generalized model, due to the enormous amount of text on which it has been trained. On a related note, we optimized our Word2Vec features by leaving the capitalization untouched. This is because Google's model has been so thoroughly trained that it effectively leverages capitalization patterns. Therefore, we select our Bag of Words + Word2Vec model, which beats all our other models in accuracy, precision, recall, and F1-score.

4.5 Regularization

The below figure shows the results of different L2 regularization parameters on our Bag of Words + Word2Vec model. We observe in the plot that a parameter of 10 establishes a good tradeoff.

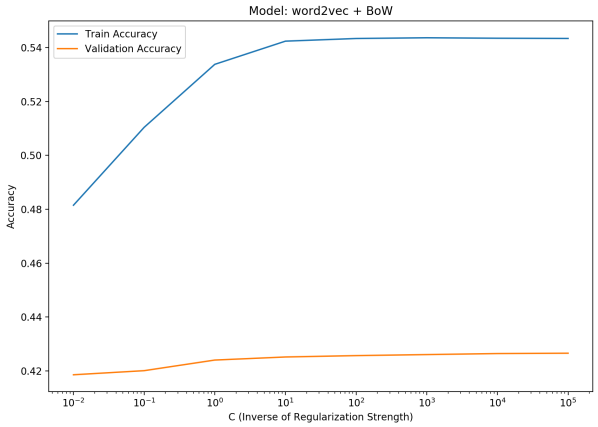

Figure 3: BoW + Word2Vec Regularization

Next, we present our confusion matrix for the Bag of Words + Word2Vec model.

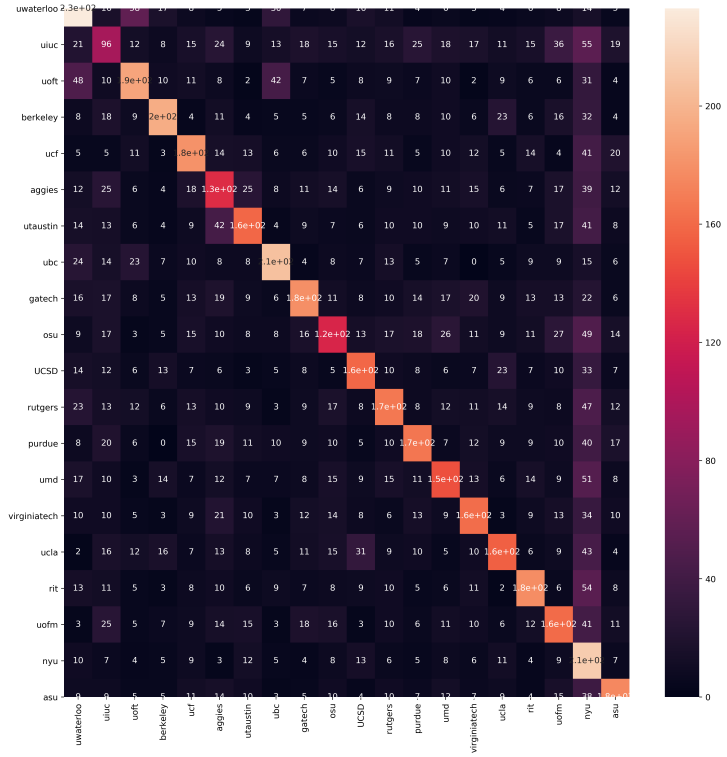

Figure 4: University Prediction - Confusion Matrix

We can see from the confusion matrix that the classifier errors are not especially biased by class. However, there is one class, NYU, which performs especially poorly. This makes complete sense in the context of our exploratory data analysis. Previously we showed that NYU students post far fewer comments per post. Therefore, there is far less text data to use in each prediction.

4.6 Concluding Remarks

We are reasonably satisfied with the performance we have achieved with these models. Word2Vec and Bag of Words are clearly well suited to this task. On the other hand, the Latent Dirichlet Allocation showed promise in its highly intuitive outputs but was not especially predictive. Our logistic regression model successfully showed the best feature representations, but there are several more sophisticated classifiers we would have liked to explore. For example, a random forest classifier or neural network would have better captured nonlinear patterns in the relationship in universities and their discussions. Ultimately, we found similar results to Giel, NeCamp, and Kader (2014), in the strength of our Bag of Words model. However, we have shown that Word2Vec consistently provides additional accuracy when combined with Bag of Words. In summation, we have shown the possibilities and difficulties of classifying Reddit comments by university using text classification methods.

References

[1] David M. Blei, Andrew Y. Ng, and Michael I. Jordan. (2003) *Latent Dirichlet Allocation*, Journal of Machine Learning Research 3 993-1022.
https://ai.stanford.edu/~ang/papers/nips01-lda.pdf

[2] T. Mikolov, K. Chen, G. Corrado, J. Dean. Efficient Estimation of Word Representations in Vector Space (2013).
https://arxiv.org/abs/1301.3781

[3] Julian McAuley, Jure Leskovec, H. Lakkaraju. What's in a name? Understanding the Interplay between Titles, Content, and Communities in Social Media (2012).
http://cseweb.ucsd.edu/~jmcauley/pdfs/icwsm13.pdf

[4] M. Gjurkovic, J. Snajder. Reddit: A Gold Mine for Personality Prediction (2018).
https://www.aclweb.org/anthology/W18-1112.pdf

[5] J. Gutman, R. Nam. Text Classification of Reddit Posts (2015).
https://jgutman.github.io/assets/SNLP_writeup_gutman_nam.pdf

[6] Giel, J. NeCamp, H.Kader. Subreddit Text Classification (2014).
https://pdfs.semanticscholar.org/b583/7babf18bb348ccbc566da67d9dcc120af3bb.pdf

[7] J. Tam. Classifying Reddit Comments by Subreddit.
https://web.stanford.edu/class/archive/cs/cs224n/cs224n.1174/reports/2735436.pdf