Using Natural Language Processing to Classify Reddit Posts

Jordan Denish
Statistics and Computer Science '21
jfd6twz@virginia.edu

Malcolm Mashig
Statistics and Computer Science '21
mjm6jy@virginia.edu

Abstract

This is the project proposal for the CS 4501 Natural Language Processing Final Project, with Professor Yangfeng Ji, written by students Jordan Denish and Malcolm Mashig. We are proposing our project to be classifying Reddit posts to determine which Subreddit webpage they belong to using Natural Language Processing.

1 Problem Definition

The problem we are trying to solve in this project is classifying Reddit posts to the Subreddit that the post belongs to using the text of the post as the feature set. We would like to incorporate several popular Subreddit pages as observations in our data set, especially in sports, politics, and culture. We initially wanted to do a project with Twitter data, but found it difficult to identify a project with clear labels that could be used in a model. We were inspired with using Reddit data when we found a similar project from a few years ago on https://github.com/templecm4y/project-reddit-nlp that classified the sports-related Subreddit pages of Reddit posts using a Naive Bayes Classifier and Support Vector Machines. We intend to perform a similar classification task by processing the text and using a neural network classifier instead. More details on our plan for performing this task is provided in the *Technical Plan* section below.

2 Justification

We are eager to explore the nuances of Subreddit classification and we believe we will come away with very interesting findings after conducting our research. Reddit posts, like posts on twitter, are current, uncensored, and convey the opinions and thoughts of people from all around the world. Unlike other forms of text, Reddit posts might be grammatically incorrect, might use slang, and/or might be flawed in some other way. That said, analyzing Reddit posts and trying to define them by their content becomes a very tricky NLP problem. Language used in the real world is not always perfect, so a classification model – one we hope to build – that can deal with imperfection, colloquialism, and trending terms and still make the correct decisions, is very powerful, practical, and also generalizable, which makes it very interesting to us.

Such a model, in Reddit's case, could remove the need for the creation of Subreddits, because it could automatically categorize a post. This concept of uncensored text generalizes to all informal text communication; the content of which, when analyzed by a model, can be automatically shared with the group of people that it was meant for. As an example, consider a recommendation engine on a site such as YouTube. This type of model could take text from a video description, classify what type of video it is, and then recommend it for people who watch those types of videos. This classification task could very well apply to other domains as well, like for Twitter or Tik Tok. The Reddit platform, however, serves as the perfect source of labeled text communication, because of the Subreddit feature.

The idea and capability of a recommendation engine are fascinating to us, as we are always impressed with how accurate and personalized YouTube or Twitter's recommendations are. With this project, we hope to scratch the surface of how the natural language processing associated with recommendation engines work with a relatively simple classification task.

Another interesting characteristic of Reddit posts is that they typically make numerous references to current events, so a classifier must be cognizant of trending words, either to ignore them for posts in the future or heavily weigh them for the time-being. For example, when we pulled data from the NBA

Subreddit, almost all of the posts mentioned the names of players that played on that night. If an unlabeled post from that night contained one of those names, we could be fairly certain that it belonged in the NBA Subreddit. However, if we are trying predict the Subreddit class of an unlabeled post from later in the year, the players mentioned in prior posts might not even be playing anymore. Therefore, while for sports Subreddits, we expect player names to be incredibly important, we need our model to extract the significance of words related to the topic that are not necessarily time-sensitive. That is just one example of an interesting nuance that we hope to uncover as we build a Subreddit classifier.

3 Technical Plan

Our technical plan for performing this classification task for our final project will be as follows.

As mentioned in the above section, we have already identified Reddit as our source for live data. We plan on scraping the Reddit data from several different Subreddit pages on a daily or weekly basis over the course of the next several weeks. For example, we are already considering scraping the Subreddits for MLB, NBA, and NFL sports leagues, Subreddits for politics and news, and Subreddits for popular culture such as movies, entertainment, and music. Each Reddit page has approximately 25 resulting posts, and by scraping hundreds of pages of Reddit posts routinely, we expect to gather at least 1,000 pages of Reddit posts from the Subreddit classes that we will be trying to identify. Therefore, we expect to have over 25,000 Reddit posts from each Subreddit classification, the label that we will try to predict. With at least 25,000 posts from several different classification possibilities, we expect to have a minimum of 150,000 text observations that will be spread across our training, validation, and test sets.

Second, we plan on parsing all of these text observations by processing the text. One of the decisions we will need to make is our word representations. We will plan on trying different word representations that have studied in this course, such as a Bag of Words representation, but also stronger and more complex methods that learn word embeddings. We plan on trying out these methods and measuring how the performance of our model compares across these word representations. Hopefully, by implementing multiple strategies and measuring their performance, we will gain further insight into which situations these methods are best suited for and why they perform stronger or weaker relative to each other.

The next step in this process is to identify the classifier that we are going to use for this task and implement it. Similar with the text processing, we plan on trying out multiple methods that have and will be taught in this course, comparing performance and learning deeper connections about how these methods work in different situations. More specifically, we plan on utilizing methods we already know such as multinomial logistic regression, but also more complex methods that we are excited to implement in the class of neural network classifiers, such as convolutional neural networks and recurrent neural networks. We want to spend a lot of time experimenting with the hidden layers of these neural networks to understand as much as we can about how the classifiers and each layer works, to gain experience modeling with deep learning. Finally, after implementing these methods, we will tune these methods to find the best performing methods on our validation sets. By working with the tuning parameters, we will gain a deeper understanding about the intricacies of each model. Finally, we will test the performance of our classifier on unseen data.

4 Experiments

We were very excited that we found a Reddit scraper online that made it extremely simple to scrape Reddit posts, and their Subreddit classification values. The scraper was easy to use, and we ran the scraper overnight for several hours to load in over 60,000 Reddit posts from six different categories (NBA, NFL, Politics, Jokes, Music, and Movies). With a total of over 360,000 posts, we exceeded the number of posts we expected to scrape and believed that this was large enough of a sample to build a strong model, while also providing a large testing set to evaluate performance.

Once we scraped the data, the next steps were to preprocess the text data for model fitting and clean up the data. We chose to preprocess Reddit posts that did not have text by replacing these empty posts with their title. There were thousands of posts without text because the post was an image or video, but we believed that replacing the empty text with the title captured enough of the post to include this in the classification. The title was of-

ten times only a handful of words, which might make classification slightly more difficult. We also needed to tokenize and preprocess that text, so it could be vectorized into a numerical matrix that could be fed into machine learning models. We removed punctuation from the text data and used a CountVectorizer to vectorize the text. We chose not to use any preset word similarities or model word embeddings in a recurrent neural network with our classification because we are not modeling sequential data and believe that our problem is simple enough that simply vectorizing the text is all that was necessary.

After preprocessing the text, we began fitting initial models and moving to more complex and powerful models. We started this project with a baseline accuracy of approximately 0.713 from a simple logistic regression multiclassification model without any tuning or regularization, and this model was able to classify the easiest examples from our Reddit categories. Our goal was to continue to improve upon this baseline accuracy using improved preprocessing and modeling techniques that we learned in this course. After tuning our logistic regression model and adding further models that utilized random forest and boosting methods that were also tuned, we ultimately determined that the best model based on cross validation had an accuracy 0.834. This validation accuracy came from our tuned L2 regularized logistic regression classifier, which we were surprised to see was our best model. However, we discuss in the sections below why we believe this to be the case. We believe that this is substantial improvement, especially since most examples that our baseline model misclassified were likely difficult and unclear. The largest jump in improvement of the model's validation set accuracy came from tuning the regularization parameter C in our logistic regression and increasing our sample size substantially to increase the model's power.

Overall, we believe that these results are decent, especially given that many of the Reddit texts we are classifying only have 2-8 words (especially the posts with pictures or videos), although we did expect to achieve a higher accuracy – probably around 0.90 at least for this project. We discuss a few reasons for this below, as well as potential ways we could improve our model's performance in the future. Below is a confusion matrix of our model's performance on all six classes.

Jokes	Music	Movies	NBA	NFL	Politics
163	6	0	2	2	4
28	97	2	2	2	1
19	6	67	6	1	1
11	3	2	137	6	2
16	4	0	4	153	1
25	2	1	0	1	161

The confusion matrix helps to highlight the Subreddits for which our classifier struggled with and those that it performed well with. For instance, a considerable number of misclassifications were made for the Jokes Subreddit. This makes intuitive sense because jokes often involve a variety of topics unlike many of our categories that are generally focused on one topic. A joke can be about pretty much any topic, so it makes sense that the words that appear in the text are less consistent in this category. Another note is that fairly few misclassifications were made for the Politics Subreddit, which also makes sense, given that the topic of politics is relatively narrow in terms of the Reddit posts.

A few unexpected issues we dealt with were simple issues with the Reddit data itself. One issue we needed to deal with was the length of time it took to fit different models. Since we scraped so much Reddit data and wanted to use all of it in order to maximize the size of our training set, model fitting and especially tuning took an extremely long time. While the size of our data set made our model much stronger compared to the baseline models, we could not fairly compare all of our different machine learning models if we were using different amounts of data for each one. Ideally, we would have put our tuning framework for the text preprocessing and model fitting together in a massive Grid Search algorithm, but this would have taken an impractical amount of time to fit a sizable number of combinations. The size of our data set and the number of parameters in our models limited our ability to tune as many combinations as we would have wanted. The size of our data also made it difficult to fit and tune neural networks quickly, which was one of our initial goals. We were able to fit neural networks (both simple and convolutional) on much smaller subsets of the data, but the neural nets and tree-based models did not perform as well with less data, as could be expected. The neural networks also tended to overfit to our training data as it was difficult to efficiently tune due the length of the model fitting process. We believe these models

were overfitting, especially for the neural networks, as the tuning process had a training accuracy of approximately 0.956 and a test accuracy of 0.644.

In an attempt to explore some ways the model was making decisions, we analyzed the weights for words in the vocab to see how they contributed to classifications of one Subreddit over another. Often, very rare words had extreme weights and so to look past this, we multiplied the weight of each word by its frequency to examine those with the most influence. We called this measure our "adjusted weights" which factored in the frequency of the words within the Reddit posts. Fortunately, the most influential words made intuitive sense. For instance, words like "watched", "scene", and "netflix" were influential for the Movies Subreddit, while words like "donald", "trump", and "election" were influential for Politics – this being no surprise for posts scraped about a month before the presidential election.

We also explored words that were potentially confusing in that they were influential for multiple Subreddits, and found some that were interesting. Our goal with this analysis was to observe words that our model tended to confuse between multiple sub-Reddits. Some confusing words for NBA and NFL, for example, were "games", "highlight", and "MVP" which are all general sports terms. In the future, we would like to explore these differences further to see if we could look at the potential misclassifications that may have occurred due to these "confusing" words. All of this demonstrates that our classifier often makes decisions in interpretable ways.

5 Conclusion

Moving forward, to improve our model, we would first consider collecting more data, either to add more potential Subreddits in an effort to complicate the model (whereby it is more powerful and applicable) or to add more data for our current Subreddits at different periods of time, in an effort to prevent biased decisions. For instance, the names of sports players or political officials tend to be influential, but these figures will change frequently with the passage of time and we will want our model to be able to adjust appropriately. Another potentially interesting expansion on our model would be to incorporate a convolutional neural network capable of classifying Subreddit categories for pictures as well – which are also included in some Reddit posts. We have experience training CNNs from another machine learning course and this would allow us to analyze pictures (when available) alongside text, rather than just relying on the brief title in some cases, which likely would improve our accuracy. We are not exactly sure how we would be able to combine the results from the image analysis and the text classification, but we are excited about the potential of that opportunity. Our image CNNs took an extremely long time to fit and tune, so this extension of our project would definitely take some time.

We really enjoyed working with real text data during this project and there are many applications that this could have. While Reddit has sub-Reddit categories, they could use the weights from our classification to find extremely similar sub-Reddit's on their platform. This could be used as a type of recommendation engine for Reddit user to find new sub-Reddit's that they would enjoy. Additionally other social media platforms and NLP platforms could use this Reddit data as labeled training data for text categorization to train their text generation and classification models. Labeled data is somewhat rare in NLP and Reddit provides millions of easily accessible text data for the public, which could be valuable for text classifiers.

To recap, we also learned how complicated working with text data can be. We achieved a relatively high baseline accuracy using simple text preprocessing and logistic regression, and with stronger hyperparameters and preprocessing, improved our performance by a solid margin. This is with a simple classification task, but when considering that Reddit would want to be classify Subreddit posts using hundreds or thousands of labels instead of our six, with an even larger vocabulary size, this project would get much more complicated very quickly.

Finally, while neural networks are extremely powerful models, not all tasks require deep learning to achieve high performance. We initially planned and expected that we would need to implement convolution or recurrent neural nets to achieve the best model performance, but this was not the case. Classical machine learning is still crucial to data science and model building and we will continue to keep this in mind as we work on further machine learning projects.