Predicting User Engagment Within Subreddit Communities

Seth Singson-Robbins

Fordham Univesrity

[ssingsonrobbins@fordham.edu](mailto:ssingsonrobbins@fordham.edu)

**Abstract—This paper investigates the predictive capabilities of engagement of Reddit posts designing a model that could predict the engagement of posts based on their upvotes, comments, and reshares. This requires a great deal of Natural Language Processing and utilizes word embeddings, Neural Networks, and other Machine Learning models to predict how much engagement a reddit post will receive. The results of the final model utilized resulted in a 27.5% improvement in the average mean error of the dataset. This investigates additional analysis of how the model performed and potential use cases of similar models.**

1. Introduction

According to Hubspot, Reddit is the world’s largest online community forum Founded in 2005 and owned by the mass global media company Conde Nast, it is the 5th most visited website with over 430 million monthly active users.(1)

Reddit’s ‘What is Reddit 101’ FAQ section in the ‘What are communities or “subreddits”’ question says that Reddit is a ‘large community made up of thousands of smaller communities’.(2) These are forum channels that are run by individual users of the site that moderate the subreddit based on rules that they determine to ensure that the subreddit has the desired types of conversation and engagement to facilitate the needs of the community. Any type of subreddit could be made allowing for a decentralization of the Reddit website.

Within each subreddit, users anywhere on reddit can submit posts usually in the form of questions or informational messages that they submit to the forum as posts with a title and text. In the posts users can engage by either upvoting or downvoting the post showing that they approve or want to encourage an increased engagement of the post. They can also comment on the post leaving their own insights and continue the discussion of the post itself. Within each comment users can continue to upvote, downvote, and even comment on each of the comments that were submitted leaving to a full discussion and subdiscussions within a post creating community engagement. They lastly can reshare the post on other parts of the site or elsewhere on the internet. These actions lead to engagement on the site but also highlight the likelihood that a post and discussion is either relevant or interesting to that subreddit community.

This paper will be looking at posts of several subreddits on Reddit.com to determine if Natural Language Processing and additional Machine Learning models can help predict the how much engagement these posts will get. These engagement metrics the model will utilize for prediction will be the upvote score (upvotes – downvotes), the number of comments, and the number of reposts that a post gets from other users within the subreddit community. This will have use cases such as determining which subreddit would be the most appropriate for a post based on how much engagement it will be predicted to receive.

1. Methodology

This section looks at the additional work done to determine and run the model. This includes how the data was collected, some dataset exploration, preprocessing on the data, training versus test data, tokenization, metric normalization, embeddings used, and what models including hypermeters were tested for final evaluation.

1. *Data Collection*

Data was collected via API connections with the Reddit website. Historically the website pushshift.io would scrape the API to pull information posts. Although this is not available at this time, the user Maksym Shkliarevskyi utilized it to pull posts from several data science related subreddits.(3) A total of 545,427 posts between March 2008 and May 2022 were pulled from the following 19 different subreddits: r/analytics, r/deeplearning, r/datascience, r/datasets, r/kaggle, r/learnmachinelearning, r/MachineLearning, r/statistics, r/artificial, r/AskStatistics, r/computerscience, r/computervision, r/dataanalysis, r/dataengineering, r/DataScienceJobs, r/datascienceproject, r/data, r/MLQuestions, r/rstats. This is a combination of subreddits relating to statistics, programming, and jobs in the industry. The important features of the data include the date of the post, the subreddit the post was posted on, and the text of both the title and the body of text that was posted. Metric features which will be utilized for metric analysis and tuning is the score, which is the number of upvotes the post got minus the number of downvotes, the number of comments, and the number of reshares of the post.

1. *Data and Text Preprocessing*

Given the nature of the data, there were several modifications that needed to be done to the data to be ready for analysis using the pandas (4) and re packages. (5)

The posts themselves was in a format not easily translatable to machine learning models containing formulas, tables, and programming language in the text. To ensure the models can take these texts and to minimize noise, several text types including numbers, new lines, long spaces, and any null values were removed from the dataset. All contractions were split up so that the actual words are all shown. Some of the text linked to pictures or videos so those were all removed and changed to just ‘http’ to signify that there was a picture or video in that spot. Only words were kept and anything that is considered a stop word by the NLTK corpus (6) was removed from the word list. If a post was empty (which occurred for about 50% of posts in the dataset), the string was set to empty instead of null to ensure it fits into the tokenization and modeling. The text was then split into each specific word left and placed into an array which is required for most models to read the data into model features.

The subreddit, title, and post of the text were concatenated to ensure that the full information of the post was included for the engagement analysis. The subreddit was included to ensure that the size of a community does not have an unintended prediction effect in the text for early modeling.

For each of the engagement metrics, all posts with no comments or reshares are null if they did not have any interaction of those metrics. These were set to zero to ensure they fit into the models.

1. *Data Exploration*

It is worth noting a few insights about the problem text and its relationship with the problem value. Each of the subreddits have a varying average engagement for each of the engagement metrics. Figures 1-3 show the average amount of engagement for each metric. These do not seem to be correlated showing that the type of engagement that users utilize vary greatly by subreddit.

Figure 1. Average Score by Subreddit

A graph with different colored bars

Description automatically generated

Figure 2. Average Number of Comments by Subreddit

A graph of different colored bars

Description automatically generated

Figure 3. Average Number of Reshares by Subreddit

A graph with different colored bars

Description automatically generated

The title of the text has a normal distribution on the average word length averaging around 7 letters per word as shown in Figure 4. The number of words in each post skews to the right showing a long tail of very long titles but most posts averaging around 6 words as shown in Figure 5.

Figure 4. Average Word Length Frequency of Post Titles

A graph of a number of words

Description automatically generated

Figure 5. Word Count Frequency of Post Titles

A graph of a bar graph

Description automatically generated

The body of the post was empty for over 50% of posts. No bodies in posts could be for a variety of reasons such as all the information being in the title or the moderator removing the post. This showed some insights into the performance of the model. Figures 6-8 shows the average performance for posts that did and did not have any text in the body of the post. Posts with text in the body on average had a higher upvote score but had a lower average number of comments. Posts with no text in the body also did not have any reshares.

Figure 6. Body in Text – Average Score

A graph with blue rectangular bars

Description automatically generated

Figure 7. Body in Text – Number of Comments

A graph with blue rectangular bars

Description automatically generated

Figure 8. Body in Text – Number of Reshares

A graph with a blue rectangle

Description automatically generated

Looking at the full text, the word count skews greatly to the right with most posts being until 15 words as shown in Figure 9. The average word length is like the title with an average length of 7 words and close to a normal distribution as shown in Figure 10.

Figure 9. Full Text Word Count Frequency

A graph of a column

Description automatically generated

Figure 10. Full Text Average Word Length

A graph of a number of text

Description automatically generated

1. *Training and Evaluation*

Several models and methods to set up the data were necessary to ensure that the optimal model was utilized for the results. To help with the evaluation of the numerous models, 20% of the dataset was separated from the training data to help with the final evaluation of all the models. Of the remain training data, an additional 20% was set aside to be used for training validation to ensure that the model and hyperparameters chosen had the best performance before being utilized for the final test data and results.

1. *Tokenization*

For most models that utilizes natural language processing, tokenization is required to map and organize the words of the text which will later be utilized to transform the text into a quantitative representations via embeddings. This allows for the models to better understand the text as actual features and utilize mathematical models to classify and predict the data. Tensorflow has the Tokenizer() class (7) that turns every word into a number and has a mapping available to reference the word.

Once all the words in the posts have been tokenized, the text themselves need to be mapped out into a vector. Each word in the subreddit name, title, and post body are all mapped out into a vector and fit onto the feature training data. Due to text being of varying lengths, padding was also required for each section to ensure that the array lists were all the same size with each text being as many items as the longest post. They were then concatenated into a final array for each post.

*H. Embeddings*

Text embeddings are required to create a numerical structure that represents either the word or the text that the machine learning models can understand. Four different methods were used with validation data being used to pick the best embedding model.

Term frequency-inverse document frequency is a method that looks at how unique a word in the text is in relation to other texts in a dataset. This helps identify what makes a document unique and what words are important for separating different documents in different classes.

GloVe is an embedding model created at Stanford University (8) and creates a vector representation of a word based on the frequency of words in the dataset. Like TdIdf, the model looks at the frequency of words in a text compared to other texts in the dataset. The GloVe model being used creates a 50-item vector for each word which creates a 2D representation of the text for each example in the dataset.

Word2Vec is a word embedding model created at Google and uses the co-occurrence of words within the text of each example to create an 2D array which will be a of every word having a 100-item representation. Gensim (9) will be utilized for this embedding.

Bidirectional Encoder Representations from Transformers, also known as BERT, is a neural network also created by Google that utilizes transformers and bidirectional encoding (hence the name) to create a numerical representation of the data. The models that utilized BERT were pulled from Tensorflow Hub (10) and are called Small-Bert having 4 layers and result in a 128-item numerical vector for each of the text in the dataset. Given that BERT analyzes the sentence instead of looking at the relationship of each word, the actual text is utilized for embedding instead of the created word lists like the other embedding models.

1. *Normalization*

The engagement data had a wide array of values that were not normally distributed and differed between the different metrics. To see the metrics apples to apples and minimize noise due to the varying lengths, the values were normalized where the average value is set to 0 and each value is set to the number of standard deviations away from the mean as the new value. The new values were named score\_norm, num\_comments\_norm, and num\_crossposts\_norm.

An engagement score was an aggregate of these three scores by adding up the three normalized values then renormalizing the engagement score allowing to minimize noise due to fluctuations in the range of values.

1. *Models and Hyperparameters*

For the actual machine learning models used, there were 5 distinct models used to find the best performing model. Each model has its pros and cons, and the performance of the models can help reveal insights into the relationships of the data with the value classification. Models used four embeddings listed in the previous section when applicable.

Three of the models used were neural networks powered by TensorFlow. (11) The first model is a general neural network that has hidden layers and dropout to help minimize overfitting. The second model is Recurrent Neural Networks (RNN) that models the data using Long Short-Term Memory (LSTM) and works for models that have a sequential patten, something common in Natural Language Processing tasks. Given the sequential nature of the neural network, TfIdf was not applied as an embedding to this model. The third Neural Network was Convolutional Neural Networks (CNN) which looks at the neighbors around each word in the text to help group and find patterns in the text. This typically is utilized for image recognition but can also be useful in Natural Language Processing tasks when the nearby neighbors have a high influence on the classification. All three neural networks were tested for performance with the same hyperparameters: number of epochs, number of hidden layers, number of items in each hidden layer including if equal or different by layer, and value of the dropout.

The other two models tested were utilizing SkLearn’s machine learning package. (12) The first was Linear Regression works at minimizing the error by applying weights to each feature then adding up the values to a final value. If this model has the best performance, it would mean that each value has a linear relationship with the final score. This model does not have any hyperparameters as it purely works to minimizing the squared error. The other model utilized was a decision tree regressor which is a model that recursively partitioning the data into subsets based on the features. The algorithm selects the best feature to split the data, aiming to minimize the variance of the target variable within the actual values. This model works when the relationship between the variables is not linear but not as complicated as neural networks. The hyperparameters used were the max depth, the minimum number of samples that need to go down each split, and the method to determine the number of features allowed for making a split.

1. Results

The results of the model have been evaluated by looking at each model’s training performance on the validation set with the best hyperparameters utilized to look at the results of the model. The main metric utilized was the average mean error, which is the average difference between the engagement score and what was predicted. A secondary metric looks at the mean squared error, which is the like the average mean error but squares the error before looking at the average. Since the data was normalized around the average, the baseline assumes that all the values would be predicted at 0 without any models with these baselines for the engagement score and each separate metric shown in Figure 11.

Figure 11. Baseline Performance by Metric



Figure 12 below shows the percent improvement of each embedding and model on the validation data versus the baseline. The RNN model utilizing Word2Vec for its embedding had the strongest performance with the optimal parameters achieving an accuracy of 27.5% on the test data on the engagement data. This would be logical as Word2Vec models each word in terms of its meaning but does not do values based on the full sentences like BERT since it is a combination of three texts and has a lot of numerical data. RNN is logical too as it does look at the relationship of the order of the words, so it has information of sentences without the full transformers like BERT.

However, this varies when looking at each metric with a Simple Neural Network with TfIdf embedding having the best upvote performance, Simple Neural Network utilizing Word2Vec on the number of comments performance, and

Figure 12. Model Performance versus Baseline

A black background with yellow rectangles and numbers

Description automatically generated

When looking at the performance of the secondary metric, the mean squared error, as shown in Figure 13, the performance does not look as good with a performance improvement of 0.8% on the mean squared error. This most likely means that the model is not as good at predicting the performance of outliers which will have a much larger error on the model.

Figure 13. Model Mean Squared Error Versus Baseline

A black background with yellow rectangles and numbers

Description automatically generated

1. Additional Analysis

With the test model there are several insights that one could ask of the data. Given the low accuracy, these should be taken as not guaranteed to be the case and stronger models could reveal conflicting results.

When looking at the average mean error by month and year of the post as shown in Figure 14, the results have better performance towards the beginning (around 2008) and the end (around 2022) of the date range. The performance was the worst around 2018.

Figure 14. Mean Average Error by Time Period

A graph showing a number of error

Description automatically generated

The performance by subreddit as shown at Figure 15 shows that the performance did vary and subreddits related to jobs in the industry having the best performance (the best subreddits were Data Science Project and Data Science Jobs) while the worst performance was related to technical subreddits (Artificial and Data Science).

Figure 15. Mean Average Error by Subreddit

A graph of colorful bars

Description automatically generated

1. Related Work

There are many papers that have been trying to undertake the task of engagement modeling with varying levels of success which will be taken under consideration for how the paper will also do its modeling. In the paper ‘Classification of Posts on Reddit’ by Naik, Sachin, and Kuri of UCSD, a classification model was created to predict the number of upvotes looking at several features, including the text sentiment (11). The paper ‘CSE 255 Assignment 2: Upvotes Prediction for Reddit Submissions’ by Wadhwa, Garg, and Gupta, also from UCSD, also predicted the number of upvotes using features of the posts like the number of words and the number of resubmissions (12). The paper ‘The Impact of Crowds on News Engagement: A Reddit Case Study’ by Horne and Adah of Rensselaer Polytechnic Institute, looked at many features of the posts like sentiment scores, the number of insightful words, and number of quotation marks to predict both the number of upvotes and the number of comments (13). None of these papers used neural network models and only one paper used an NLP Embedding model (only GloVe) without using more recent NLP models for comparison. The paper ‘A Tale of Two Subreddits: Measuring the Impacts of Quarantines on Political Engagement on Reddit’ by Shen and Rose of the Language Institute looked to see how the effects of quarantine during the COVID pandemic affected the political changes and engagement on different subreddits showing that time can be a factor in the quality and engagement of posts (14).

1. Conclusion

Given the low performance of the model and lack of correlations during the exploration of the dataset, it comes into question if the posts are a good indicator for user engagement on the reddit site. There is a lot of noise that comes from if a user will interact with a post including the time of day and the type of users that are in each of the subreddit communities. This could vary greatly and a better understand could come from qualitative studies as an enrichment to the models where interviewers can directly ask users what goes into their decisions to like or comment on posts, they see in their reddit feeds.

If a better model is created, this could have many use cases such as creating a model to let users know the best subreddit they should add their posts to for engagement. It could also help understanding what types of messages are important to the communities that are within each subreddit.

1. References

(1) P. Bump, “24 Reddit Stats and Facts to Know in 2022,” blog.hubspot.com, Oct. 18, 2022. <https://blog.hubspot.com/marketing/reddit-stats>

(2) “What are communities or ‘subreddits’?,” *Reddit*, Apr. 2023. https://support.reddithelp.com/hc/en-us/articles/204533569-What-are-communities-or-subreddits-

(3) “Reddit Data Science Posts (500k+),” www.kaggle.com. https://www.kaggle.com/datasets/maksymshkliarevskyi/reddit-data-science-posts

(4) Pandas. (2018). Python Data Analysis Library — pandas: Python Data Analysis Library. Pydata.org. https://pandas.pydata.org/

(5) Python. (2009). re — Regular expression operations — Python 3.7.2 documentation. Python.org. <https://docs.python.org/3/library/re.html>

(6) nltk.corpus package — NLTK 3.5 documentation. (2012, December 12). Www.nltk.org. <https://www.nltk.org/api/nltk.corpus.html>

(7) tf.keras.preprocessing.text.Tokenizer | TensorFlow Core v2.3.0. (n.d.). TensorFlow. <https://www.tensorflow.org/api_docs/python/tf/keras/preprocessing/text/Tokenizer>

(8) Pennington, J. (2014b). GloVe: Global Vectors for Word Representation. Stanford.edu. https://nlp.stanford.edu/projects/glove/

(9) Gensim: topic modelling for humans. (n.d.). Radimrehurek.com. https://radimrehurek.com/gensim/

(10) TensorFlow Hub. (n.d.). Tfhub.dev. Retrieved December 13, 2022, from <https://tfhub.dev/google/collections/bert/1?utm_source=www.tensorflow.org&utm_medium=referral>

(11) P. Naik, S. A S, and V. Kuri, “Classification of posts on Reddit \* Sachin A S,” University of California in San Diego. Available: https://cseweb.ucsd.edu/classes/wi15/cse255-a/reports/fa15/021.pdf

(12) R. Wadhwa, V. Garg, and K. Gupta, “CSE 255 Assignment 2 : Upvotes Prediction for Reddit Submissions,” University of California in San Diego. Available: <http://jmcauley.ucsd.edu/cse258/projects/fa15/014.pdf>

(13) B. Horne and S. Adali, “The Impact of Crowds On News Engagement: A Reddit Case Study,” *Proceedings of the International AAAI Conference on Web and Social Media*, vol. 11, no. 1, pp. 751–758, May 2017, doi: https://doi.org/10.1609/icwsm.v11i1.14977.

(14) Q. Shen and C. P. Rosé, “A Tale of Two Subreddits: Measuring the Impacts of Quarantines on Political Engagement on Reddit,” *Proceedings of the International AAAI Conference on Web and Social Media*, vol. 16, pp. 932–943, May 2022, doi: https://doi.org/10.1609/icwsm.v16i1.19347.