RedditRank Proposal Document

Murtaza Latif - 1004307065
Andrew Wang - 1003259305

Total Word Count: 1172
Penalty: 0%

Introduction

On Reddit, content is prioritized by user voting through "upvotes" and "downvotes" and organized in subsites called "subreddits". The goal of our project is to predict the final score of a given post in the subreddits "AskReddit" and "Pics". Neural networks are well-suited for this task given the breadth of inputs and the labelling (votes) intrinsic to posts. Neural networks also provide enough adaptability to process both image and text data.

Since users often upvote what they agree with, the ability to predict post popularity can translate directly into insights for underlying prejudices and affinities of the voting population. Beyond gaining this demographic information, a predictor can be used widely by industry and users alike to improve advertising or craft better content by providing a means for self-validation.

Background and Related Work

One related project that exists is *Comment Karma Predictor on Internet Forums* [1]. This project utilized binary classification to determine comment popularity, and linear regression to predict comment score. Like us, they are collecting data about online posts and predicting how well users of that forum would respond (via voting) to the comments.

A thesis paper titled, *Popularity Prediction of Reddit Texts* [2], is also relevant. The paper outlines the challenges of popularity classification in a massive forum where both post topics and user cultures are dynamic. The study uses various approaches to tackle the problem and reports the results of the models used in a specific set of subforums.

Illustration / Figure

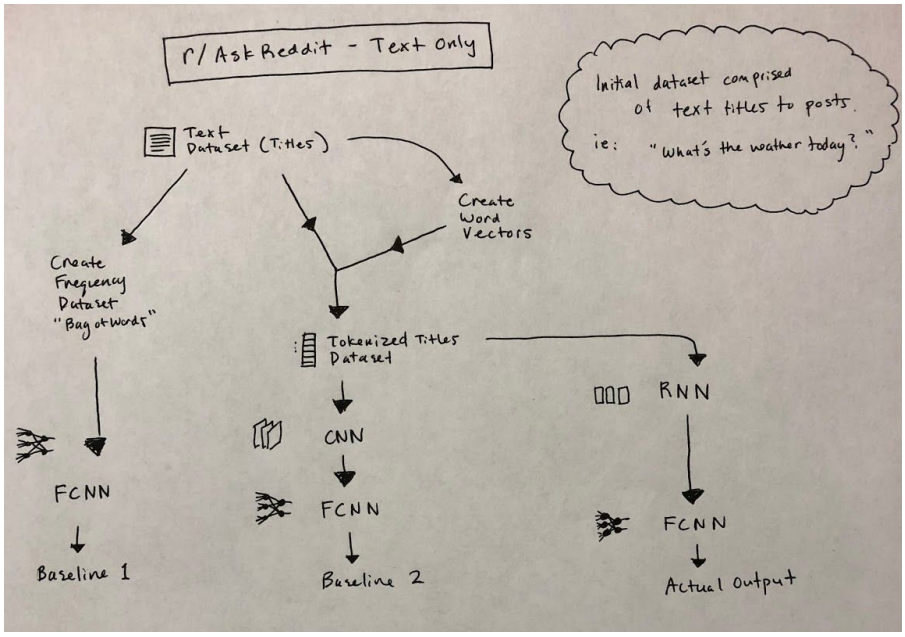

Figure 1: *Architecture for predicting "r/AskReddit" post scores using post titles*

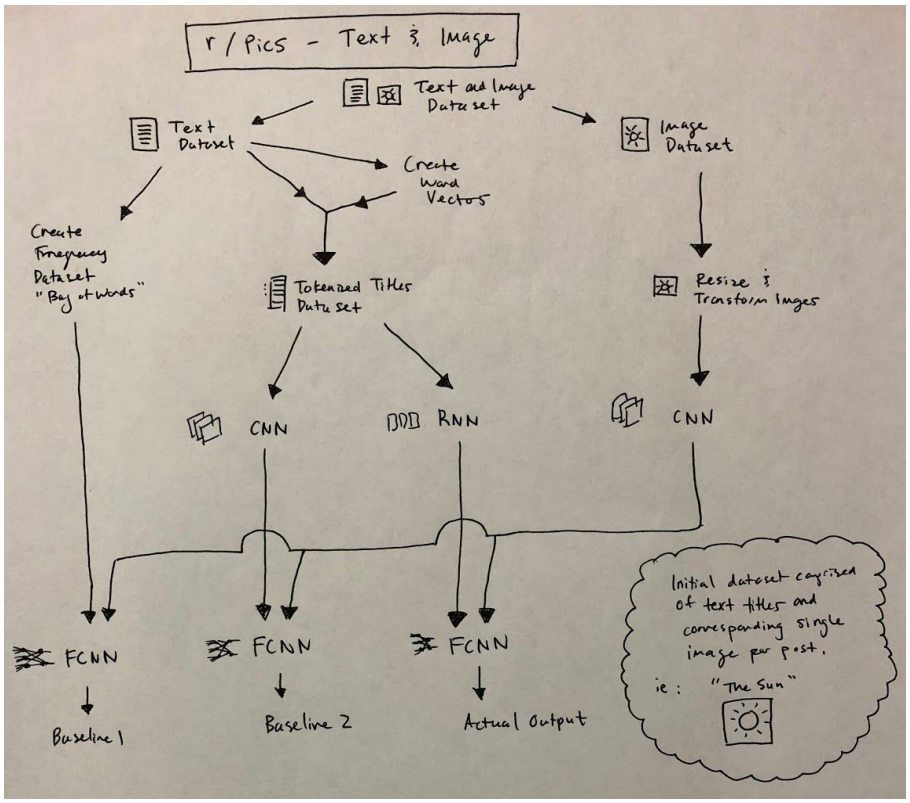

Figure 2: *Architecture for predicting "r/Pics" post scores using post titles and images*

Data Source, Labelling and Processing

There are multiple publicly available datasets [3][4] with pertinent data. Due to our project's scope, the majority of the corpus will be comprised of data that is collected from scraping the Reddit website using a Python package called PRAW [5]. This will prevent the dataset from consuming more space than necessary.

The network will predict the score of a post after 2 days (Reddit's algorithm biases against older posts [6]). Hence, each 2-day or older post will use its score as its label. Practically, this can be achieved by filtering out posts that are younger than 2 days.

Architecture

Given a post in "r/AskReddit" or "r/Pics", we expect the post's title as a text input for both and the post's content as an image input for the latter. To process the titles into tokenized form, we will use the GloVe pre-trained word vectors in an embedding layer. These vectors will pass through a recurrent neural network (RNN) using GRUs.

If the post is from "r/AskReddit", we simply pass the RNN output through a fully connected neural network (FCNN) to output a score prediction. If the post is from "r/Pics" we incorporate the image as well. We take each image, resize it, normalize it, then pass it through a convolutional neural net (CNN). The output from both the CNN and aforementioned RNN are then both passed into a FCNN.

Baseline Model

The project will use two baseline models. The first will take each post title and convert it into a word frequency vector ("bag of words"). These vectors will then be passed through a FCNN to output a predicted score. The second baseline will begin with the tokenized dataset outlined above. Instead of using an RNN, it will run the data through a CNN and then an FCNN.

As above, for posts from "r/Pics", we will take the output from each image through a CNN and concatenate it with the output from the text processing before running it through a FCNN. For the first baseline, this amounts to concatenating each word frequency vector with its CNN-processed image. For the second, it means concatenating each CNN-processed title with its respective CNN-processed image. The below table summarizes the underlying architecture for the various models.

Table 1: *Summary of architectures where the data is transmitted through the systems listed from left to right.*

r/AskReddit	**Model**	Tokenize	RNN	FCNN
	Baseline 1	Create Frequency Vector		FCNN
	Baseline 2	Tokenize	CNN	FCNN
r/Pics	**Model**	Tokenize Text	RNN	FCNN
		Preprocess Images	CNN	
	Baseline 1	Create Frequency Vector		FCNN
		Preprocess Images	CNN	
	Baseline 2	Tokenize Text	CNN	FCNN
		Preprocess Images	CNN	

Ethical Framework

We preliminarily identify the stakeholders as those who use Reddit ("redditors"), advertisers, the company itself, and ourselves (as creators). Under the assumption that our model is accurate and used widely, we can consider many ethical implications.

Foremost, advertisers could use the model to learn what content is popular on Reddit, thus better tailoring ads. Validation of prototype ads can also be done through the model. Second, redditors benefit in terms of justice (equality), as all users can equally improve their grasp on what a popular post is through trial and error using the predictor.

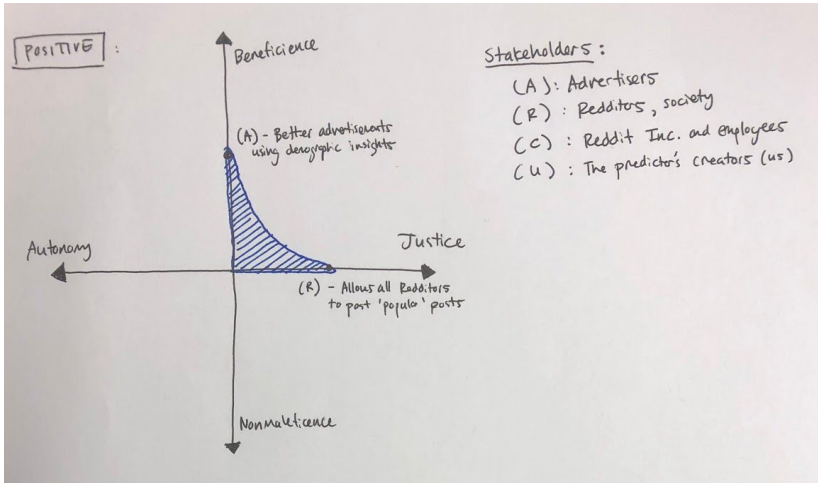

Figure 3: *Positive reflexive principlism graph*

Conversely, redditors will only want to post content with high predicted scores. This potentially decreases the amount of content submitted; this is a downside for both redditors and Reddit Inc., since less content correlates to less traffic.

Since Reddit is a dynamic platform, data acquisition and retraining would occur frequently. This is a large computing cost for the model's owners. If the model algorithm is not open source, biases (unconscious or conscious) may be introduced, skewing prediction results (e.g. skewed towards certain political ideologies).

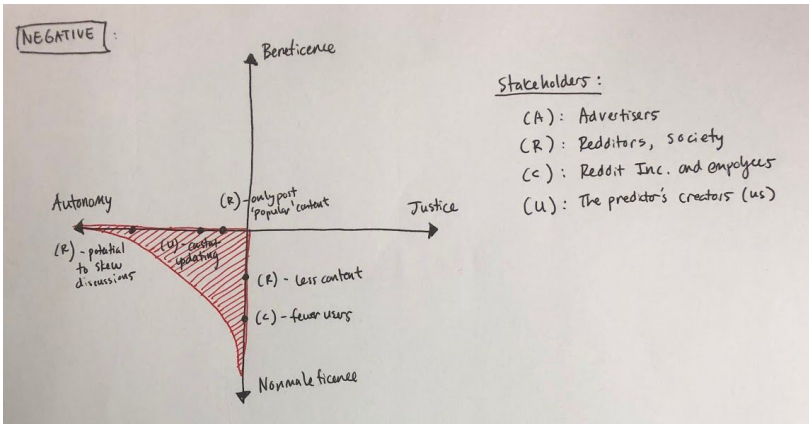

Figure 4: *Negative reflexive principlism graph*

Project Plan

We plan to meet often and use online communications secondarily. The initial plan is to meet semi-weekly with adjustments according to unexpected obstacles. Below is a schedule that divides tasks into two major responsibilities corresponding to each partner.

Table 2: *Task, deadline and responsibility breakdown plan*

Project Task	Responsibility Breakdown	Deadline
Data Collection	• Complete web scraping system • Filtering, labelling and splitting	10-31-2019
Baseline Creation	• Build first baseline model • Build second baseline model	11-07-2019
Compiling Progress Report	• Collect figures, data and statistics for the report • Writing the report (will be further split upon receiving the document outline)	11-12-2019
Main Model	• Complete construction of model architecture • Use Git to split and merge portions of the model	11-23-2019

Testing and Debugging	• Perform tests to identify issues in the model • Make changes to the model to improve accuracy	11-30-2019
Final Report & Presentation	• Collect figures, data and statistics to use during the presentation and report • Creating the report and presentation content	12-02-2019

Risk Register

The most likely major risk is that the model may take too long to train. The size of each example can be large (up to 149 words for a title), and there is a potential risk that the model will converge too slowly. This issue may be addressed by finding alternate methods of computing (using GPU or Cloud Computing) and by optimizing hyperparameters towards training speed (smaller batch sizes or layer sizes).

Another major risk is a lack of words in the word vector dictionary. Slang terms are commonly used on Reddit, but they cannot be converted using GloVe. We can expect this to happen with a moderate likelihood given the nature of the subreddits we are investigating. We do not expect slang to appear often in questions nor picture submissions titles. A solution to this risk is filtering out data that uses these words and collecting more data to fill in the gaps. However, this solution may narrow the scope of the classifier to view posts with words that are recognized by GloVe.

References

[1] D. Lamberson, L. Martel, and S. Zheng, “Hacking the Hivemind: Predicting Comment Karma on Internet Forums,” 2014. [Online]. Available: http://cs229.stanford.edu/proj2014/Daria%20Lamberson,Leo%20Martel,%20Simon%20Zheng,Hacking%20the%20Hivemind.pdf

[2] T. Rohlin, “Popularity Prediction of Reddit Texts,” n.d.. [Online]. Available: https://scholarworks.sjsu.edu/cgi/viewcontent.cgi?article=8251&context=etd_theses.

[3] Directory Contents. [Online]. Available: https://files.pushshift.io/reddit/submissions/.

[4] Linanqiu, “linanqiu/reddit-dataset,” GitHub, 22-Feb-2018. [Online]. Available: https://github.com/linanqiu/reddit-dataset.

[5] G. Tanner, “Scraping Reddit data,” Medium, 12-Feb-2019. [Online]. Available: https://towardsdatascience.com/scraping-reddit-data-1c0af3040768.

[6] A. Salihefendic, “How Reddit ranking algorithms work,” Medium, 19-Mar-2016. [Online]. Available: https://medium.com/hacking-and-gonzo/how-reddit-ranking-algorithms-work-ef111e33d0d9.