

Reddit Karma Farmer

Shuyang Wang

shuyangw@umass.edu

Oliver Pang

opang@umass.edu

Quyen Tran

quyentran@umass.edu

Justin Lee

justinlee@umass.edu

1 Introduction

Reddit.com is an extremely popular website that is meant for news aggregation and general topic discussion. One of the most important aspects of Reddit, is the ability to give imaginary internet points called "karma" to comments, and posts that one likes. What we want to create is a neural network that will generate a comment in a specific subreddit that is likely to receive karma. As a group, we all like to browse Reddit, and thought it would be interesting to create this program and see if its comments could actually generate karma from actual users. We feel that it is important to see if bots/programs could actually cause users to upvote their posts and trick them into thinking they are real. There is also current concern over particular parties buying high influence accounts to influence public opinion from within Reddit. This study can help verify the plausibility of using automated techniques to acquire those accounts.

Reddit has its own sub-communities called "subreddits" which users can browse and subscribe to according to their needs and interests. For example, those who enjoy basketball can subscribe to the subreddit "r/nba", with the "r/" prefix characters denoting it as a subreddit. This implies that only comments pertaining to the topic of the subreddit will be considered by other users, while off-topic comments will be voted down or "downvoted".

Typically, more successful comments will have a high rating relative to the number of subscribers of the subreddit because a great number of users approve of or share the same opinion as the original comment. Comments may even be "gilded", meaning that other users have anonymously sponsored a comment with money out-of-pocket. We would like to analyze these metrics and generate our own comments that exhibit the behaviors seen

in highly rate comments and we would like to generate our comments in accordance to each subreddit to maximize the potential gain of karma.

2 Related work

There has been significant amount of work done with regards to the generation of text that involve LSTM Recurrent Neural Networks. We have found various articles and papers regarding the generation of text and comments with relation to given context.

The first paper deals with sequential data prediction, which involves the problem of predicting succeeding words given some context (Mikolov et al., 2010). This paper forms the basis of text generation using RNNs upon which we will build. The intent of the paper is to provide a superior solution to the previous feedforward NN and backoff models. To achieve this, the team used a model they denote as RNN LM, where the core of the model consists of a Elman network with a hidden layer size dependent on the training data. They also allow the model to be dynamic, in that it trains even during the testing phase. By stacking these upon each other they were able to outperform their targets.

As this paper simply provides an early application of RNNs to text generation, many more advanced approaches have since outperformed the model.

One such approach, called the Gated-Feedback RNN (GFRNN) (Chung et al., 2015), utilizes global gating units per pair of layers to control signals from upper to lower recurrent layers. Specifically, this structure distinguishes itself from traditional RNNs by allowing information from upper recurrent layers to flow back into lower stacked recurrent layers, capturing long-term and short-term dependencies.

3 Your approach

One approach that we're going to explore is utilizing LSTM RNNs to generate the text that serves as the body of the comment that we wish to compute. We chose LSTMs to allow for increased context capabilities on top of countering the vanishing gradient problem encountered while training rnns.

We will train our neural network on the dataset we've defined in a later section of this proposal. After pre-processing by extracting the specific data we need, we intend to encode the word tokens acquired as integers in embedding vectors. The words of a high karma comment must also be higher weighted, so we may weigh words based on the karma of the sentence, or duplicate training data based on karma. Another is to simply train on comments with a karma above a certain point. We will be exploring the different approaches to determine which one is more feasible and/or accurate for our purposes.

We can later compare our results to that of other text, or more specifically, comment generation models that also utilize neural networks or Markov chain models. We may even post these comments to the live website to see how much attention, positive or negative, we can garner.

What baseline algorithms will you use?: As our final metric will be based upon how much karma the bot is able to farm, we have decided on using a Markov Chain model trained on the same data as a baseline algorithm. This approach has proved effective for generating semi-sensical sentences, and is even employed in the subreddit "r/subredditsimulator". Our hope is that by using the deeper nature of RNNs and the context memory of LSTMs, we generate comments that take into account more than the last word in the sentence, and thus are more sensical overall.

3.1 Milestones & Schedule

1. Take in Reddit comment data, and preprocess it (1 week)
2. Build baseline neural network structure and markov chain model (2 weeks)
3. Write progress report. (due Nov. 16)
4. Train and test neural network (2 weeks)
5. Analyze the output of the network, debug and work out potential problems (2 weeks)

6. Work on final report and presentation (1 weeks)

4 Data

Reddit comment datasets are readily available online for public use. The data is presented in JSON format with fields defining comment body, author, time created, edited, id values, etc... Each comment given by Reddit's API comes with an amount of metadata. Data that needs to be extracted from that metadata contain the following:

- Body - The message of the comment itself.
- Upvotes - Number of times the comment was voted with positive reaction.
- Downvotes - Number of times the comment was voted with negative reaction.
- Score - Number of upvotes subtracted by the number of downvotes.
- Gilded - Number of times a comment has been gilded. Gilded comments are specially recognized by users.

The data that we will be using initially is a ~5GB subset of comments taken from 2015. If we do require additional data, we can utilize a ~1TB dataset of many more comments.

Any additional data can be retrieved through Reddit services and formatted accordingly. Because our dataset consists of Reddit comments, it is the most reliable source to train a neural network on readily available text to generate our own Reddit comments.

5 Tools

Since our work primarily focuses on handling data and managing our neural network, we will writing almost all, if not all of our codebase in Python 3.6. More specifically, we plan to make heavy usage of machine learning libraries such as Tensorflow and possibly also PyTorch and Keras to implement the specifics of our model.

References

- Chung, J., Gülçehre, Ç., Cho, K., and Bengio, Y. (2015). Gated feedback recurrent neural networks. *CoRR*, abs/1502.02367.
- Mikolov, T., Karafit, M., Burget, L., Cernock, J., and Khudanpur, S. (2010). Recurrent neural network based language model. 2:1045–1048.