

---

# A Look Into the World of Reddit with Neural Networks

---

**Jason Ting**

Institute of Computational and Mathematical Engineering

Stanford University

Stanford, CA 94305

jmting@stanford.edu

## Abstract

Creating, placing, and characterizing social media comments and reactions is a challenging problem. This is particularly true for reddit.com, a highly trafficked social media website with thousands of posts per day. Each post has an associated comment thread, and users of Reddit can vote the comments up or down, generating a net score, or "Karma," for each comment. Users aspire to collect this "Karma," and these comments build the community on Reddit. When reading the content of the comment thread, however, it is often unclear why some comments succeed and receive high Karma while other comments lose Karma. This project explores using Recursive Neural Networks to give better insight into Reddit communities through the sentiment analysis of Reddit comments and Recurrent Neural Networks on Reddit comments to characterize the voting patterns of Reddit users and determine the Karma strength of Reddit comments through identifying comments with positive and negative Karma.

## Introduction

Reddit is an entertainment, social networking, and news website where registered community members can submit content, such as text posts or direct links to websites. Content entries are organized by topics of interest called subreddits. Users can then vote submissions up or down to determine their position on the site's page based on the net positive votes that the post accumulated called Karma. Furthermore users can make comments on these submissions and vote on these comments to also determine their position on the site's page for Karma. The website has gained massive popularity over the years, receiving over 7.5 billion total page views and 170 million unique visitors a month. Moreover, Reddit users vote over 24 million times on the website each day [1].

The enormous popularity of Reddit makes it a fascinating website for social analysis. With such a large number of users writing comments and responses to posts along with voting on content, there are plenty of insightful patterns of social interactions characterized through the posted comments and the Karma associated with each post [2]. Since these users upvote on content they believe is good and downvote content they believe is bad, comments with positive Karma can be associated with positive sentiment and comments with negative Karma can be associated with negative sentiment to the users in their respective subreddit. The primary motivation of this project is to gain better insight on what brings people together in online communities through sentiment analysis of subreddit comments through using Recursive Neural Networks (RecNN) and how users perceive sentiment in the context of their own community through Recurrent Neural Networks (RNN) to classify what comments have net positive or negative Karma in their respective community.

## Background/Related Work

While using neural networks has not been applied to analyzing Reddit comments before, there has been past work using Reddit's data to analyze the website. Lakkaraju [3] sought to evaluate Reddit submissions based on the title of the submission. This paper studied how the title, submission times, and community choices of image submissions affect the success of the content by investigating resubmitted images on Reddit. The language model used include modeling good and bad words, LDA, parts of speech tagging, length of title, sentiment analysis, and Weighted Jaccard similarity. Using this language model they were able to account for how the success of each submission was influenced by its title, and how well that title was targeted to the community. Although using language models is an important factor to the success of the post, the quality of the content, submission time, and the community contributed greatly to the success of the post, especially for the posts that accumulated a lot of Karma.

## Experiment

### Data

The Reddit API and code are open sourced so Reddit's comments can be readily scraped. Initially the comment found on Reddit was going to be scraped continuously over the period of a week, but the dataset took fresh comments off r/new and had no voting data associated with the post. Furthermore there is a rate limiting issues with Reddit, where the website would only permit users to grab data every two seconds. To resolve these issues, I ran a MapReduce [4] job that uses multiple computers to fetch data from Reddit simultaneously along with utilizing the praw library [5] to interact with the Reddit API. Using this method the dataset consists of approximately 1 million comments from Reddit taken from the top 200 posts of 10 popular subreddits. Comments longer than a set threshold were filtered out. Some time series analysis of the scraped comments are shown in Figure 1.

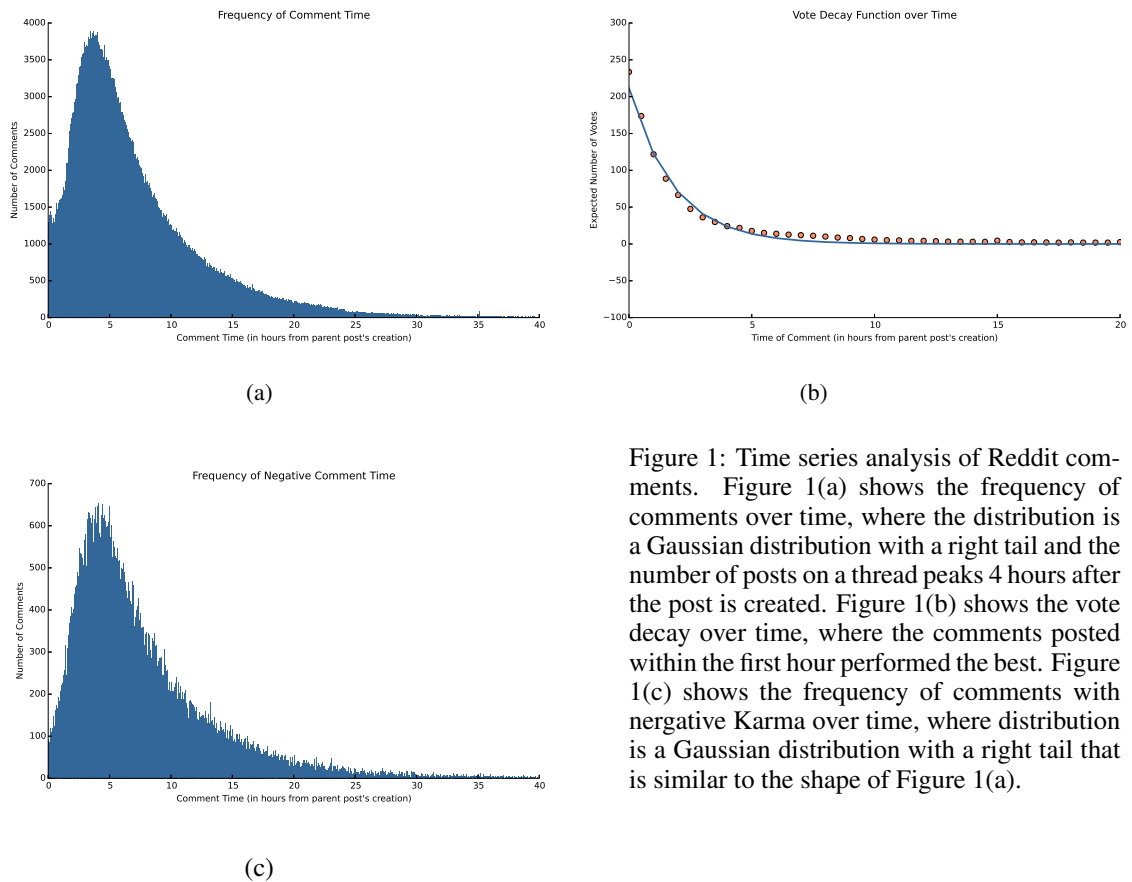


Figure 1: Time series analysis of Reddit comments. Figure 1(a) shows the frequency of comments over time, where the distribution is a Gaussian distribution with a right tail and the number of posts on a thread peaks 4 hours after the post is created. Figure 1(b) shows the vote decay over time, where the comments posted within the first hour performed the best. Figure 1(c) shows the frequency of comments with negative Karma over time, where distribution is a Gaussian distribution with a right tail that is similar to the shape of Figure 1(a).

Based on Figure 1(a) and Figure(b), Reddit users most frequently comment soon after the post is created, whose comments posted early after the post was created in general does the best. Just as Lakkaraju [3] concluded that submission time significantly contributes to the success of a post, comment post time significantly contributes to the success of the comment, making it difficult to build a model that relies entirely on the comment content to predict the total number of Karma a post will receive. However Figure 1(c) shows that the distribution of the posts that receive negative Karma is similar to the distribution shown in Figure 1(a), therefore controlling the time component along with other confounding factors, thus permitting sentiment driven models to classify comments based on positive and negative Karma.

### Evaluation Metric

The evaluation metric used for this project is the accuracy of what comments are classified correctly along with a confusion matrix of what comments are misclassified. Cross Validation is used to train the model and where I assigned 70% of the data to the training set, 15% of the data to the validation set, and 15% of the data to the test set. Each model trains on each subreddit separately and the model results are the average performance in the training and test set of the 10 scraped subreddits.

### Approach

These neural network models should capture conventional sentiment of the comments and sentiment as perceived by the users in their respective subreddit. For the former RecNNs have the capacity to learn from data with a nested hierarchy data structure and capture sentiment of a phrase [6], so RecNNs is used for this task.

For determining sentiment by the subreddit’s context through the positive and negative Karma labels, RNNs are used for this task. Since each comment is associated with the label positive/negative for the full comment, RNNs must handle data with learning long-term dependencies and account for the vanishing gradient problem, where gradient signal gets so small that learning either becomes very slow or stops working altogether. Gated Recurrent Unit and Long-Short Term Memory are models that accounts for these problems and are best suited for the task. RNNs are used from the Theano library [7] and uses a Cross Entropy loss function.

### Baseline Model

To demonstrate the effectiveness of RNNs, a baseline model is used to classify the Reddit comments. For feature selection I used the bag of words and the supervised learning models Gaussian Naive Bayes and Support Vector Machines (SVM).

### Recurrent Neural Network

RecNNs compute compositional vector representations for phrases of variable length and syntactic type to classify each phrase [6], which can be seen in Figure 2. Sentiment are classified into 5 classes, where the posterior distribution of the labels are defined as the following:

$$y^a = softmax(W_s a)$$

The data comes from Stanford’s CoreNLP Sentiment Treebank [8], which I used Stanford Parser to generate parse trees and binarized them using the TreeBinarizer class within leparser, collapsing nodes with single children for nodes without labels.

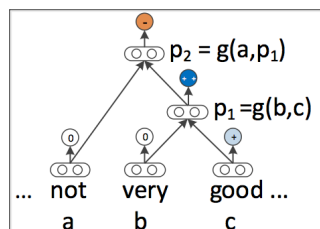


Figure 2: A diagram [6] of Recurrent Neural Network for sentiment. The parent vectors are computed from the bottom up with a compositionality function  $g$  and use node vectors as features for a classifier at that node.

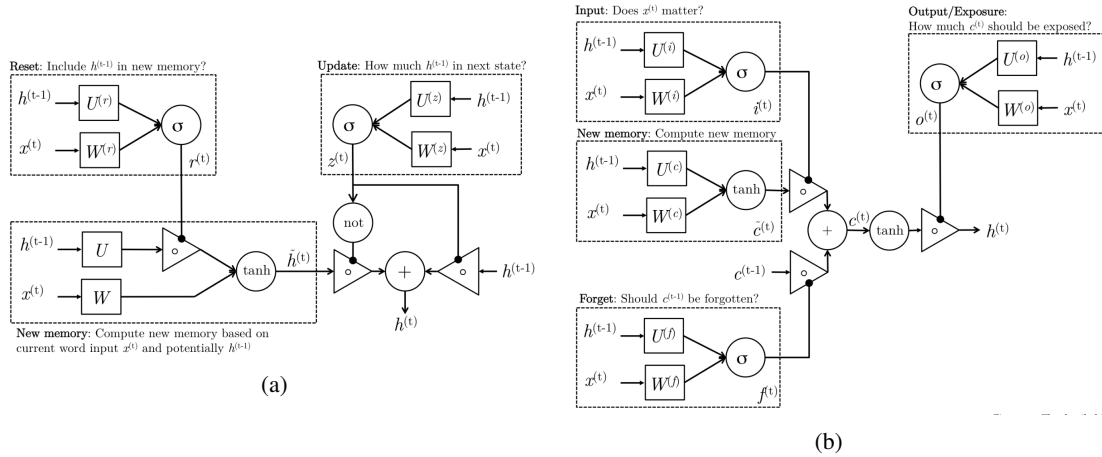


Figure 3: A diagram [9] of the GRU model in (a) and LSTM model in (b). A description of the role of each equation is written above each component.

## Recursive Neural Network

### Gated Recurrent Unit

Gated Recurrent Unit (GRU) have the capacity to allow the gradient to flow at different strengths depending on the inputs and capture long-term dependencies [10]. GRU can be broken down to four fundamental operational stages. Each stage can be described mathematically with the following four equations whose model is described in Figure 2(a):

$$\begin{aligned} \text{Update Gate: } z^{(t)} &= \sigma(W^{(z)}x^{(t)} + U^{(z)}h^{(t-1)}) \\ \text{Reset Gate: } r^{(t)} &= \sigma(W^{(r)}x^{(t)} + U^{(r)}h^{(t-1)}) \\ \text{New Memory: } \tilde{h}^{(t)} &= \tanh(r^{(t)} \circ U h^{(t-1)} + W x^{(t)}) \\ \text{Hidden State: } h^{(t)} &= (1 - z^{(t)}) \circ \tilde{h} + z^{(t)} \circ h^{(t-1)} \end{aligned}$$

### Long-Short Term Memory

Long-Short Term Memory (LSTM) is a complex type of neural network model that differs from GRU. LSTM can be broken down to six fundamental operational stages [11]. Each stage can be described mathematically with the following six equations whose model is described in Figure 2(b):

$$\begin{aligned} \text{Input Gate: } i^{(t)} &= \sigma(W^{(i)}x^{(t)} + U^{(i)}h^{(t-1)}) \\ \text{Forget Gate: } f^{(t)} &= \sigma(W^{(f)}x^{(t)} + U^{(f)}h^{(t-1)}) \\ \text{Output/Exposure Gate: } o^{(t)} &= \sigma(W^{(o)}x^{(t)} + U^{(o)}h^{(t-1)}) \\ \text{New Memory Cell: } \tilde{c}^{(t)} &= \tanh(W^{(c)}x^{(t)} + U^{(c)}h^{(t-1)} + W x^{(t)}) \\ \text{Final memory cell: } f^{(t)} \circ \tilde{c}^{(t-1)} + f i^{(t)} \circ \tilde{c}^{(t)} \\ \text{Final hidden state: } h^{(t)} &= o^{(t)} \circ \tilde{c}^{(t)} \end{aligned}$$

## Results

### RecNN Comment Sentiment

The results are shown in Figure 4. Each model was tested using coarse hyperparameter tuning to find optimal learning rates and regularization. Then I implemented a two-layer fully connected neural network with a softmax classifier on top. I did coarse hyperparameter tuning over learning rate, regularization, number of units in hidden layer, dropout rate, and number of epochs. I also added a depth level index in the hidden layer. The best model was the 2 layer 35 unit hidden layer model, which achieved 81.26% classification accuracy on the training set and 81.23%, which is significantly better than the baseline model, and a confusion matrix of the best model can be seen in figure 5. Despite this accuracy, the confusion matrix showed that the misclassifications tend to

guess comments that are more neutral.

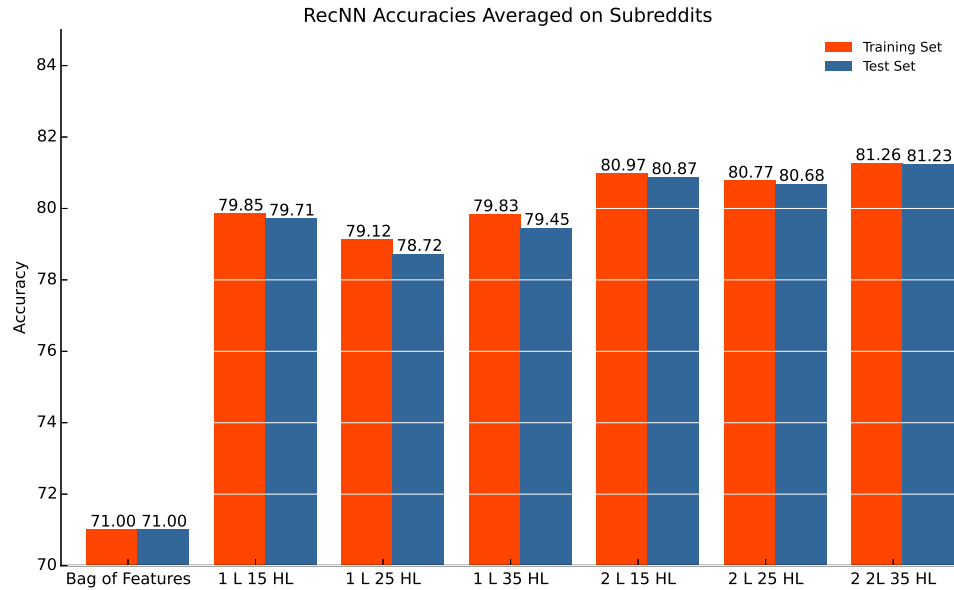


Figure 4: L is Layer and HL is units of hidden layers in the model.

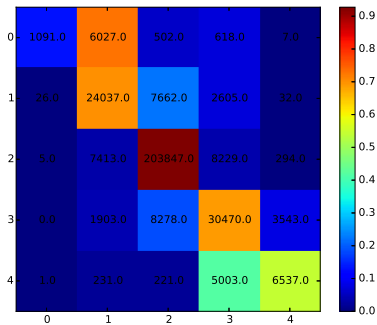


Figure 5: Confusion matrix for highest performing RecNN model Sentiment represented in following order from 0: very negative, negative, neutral, positive, very positive.

Using the best RecNN sentiment model, a distribution of the sentiment for 10 popular subreddit can be seen in Figure 6.

### RNN Comment Karma

The results are shown in Figure 7. Each model was tested using coarse hyperparameter tuning to find optimal learning rates and regularization. I did coarse hyperparameter tuning over learning rate, regularization, number of units in hidden layer, dropout rate, and number of epochs. The net run with tuned parameters showed higher performance than linear classifiers on the hand-built features. The best model was the LSTM model with 128 hidden layer units, which achieved 93.06% classification accuracy on the training set and 92.79% , which is significantly better than the baseline model, and a confusion matrix of the best model can be seen in figure 8. The confusion matrix shows the types of misclassifications are mostly even.

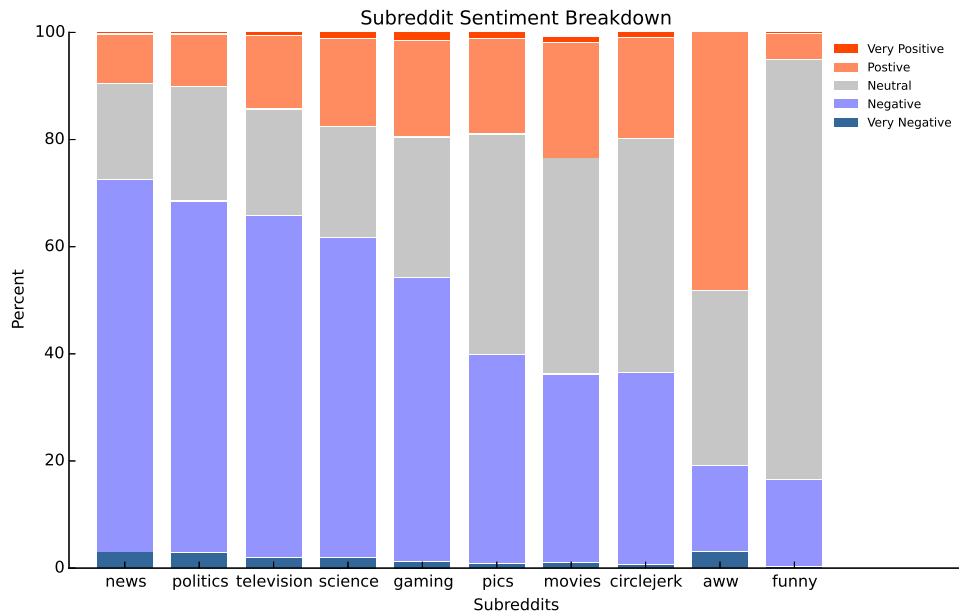


Figure 6: Percentage of comment for each class of sentiment for 10 popular subreddits.

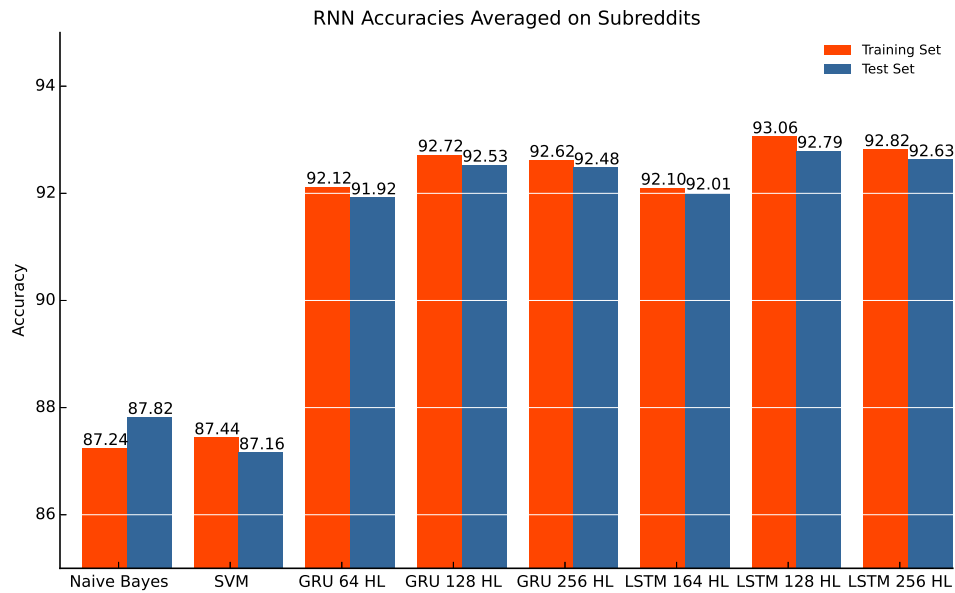


Figure 7: HL is units of hidden layers in the model.

## Conclusion

This project demonstrated that using RecNN for sentiment analysis on Reddit comments is viable and works well, and GRUs and LTSMs are reliable methods for measuring the quality of Reddit comments. The highest performing RNN models were able to classify Karma strength of comments

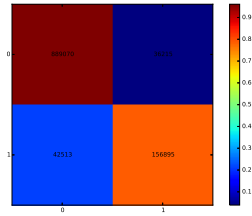


Figure 8: Confusion matrix for highest performing RecNN model Sentiment represented in following order from 0: Postive and Negative.

well despite being re-purposed from classifying sentiment, which indicates that there is some correlation between Karma and perceived sentiment.

The best RecNN model classified the comments sentiment with 81.23% accuracy. The best RNN model classified the comments sentiment with 92.79% accuracy.

There are some issues with the approach that must be mentioned. Because Reddit is an internet forum, so the comments do not necessarily have to be context driven. For example, many users post a comment as a URL for images that serves as their reaction or reference to other users, which provides little to no information on sentiment. As a result, these types of comments are noisy to the model and this approach would not work well in subreddits where users frequently post their comment context indirectly. Furthermore, the dataset consists of popular subreddits, so the model may not be generalized to work on less popular subreddits or subreddits with peculiar comments.

The next step to train the models on less popular subreddits to test how well these models work more generally. Furthermore, the RNN model can be improved and provide further analysis is to incorporate the other features that contribute significantly to the success of a comment into the neural network models, such as time of post, the post title, and content. Through integrating these features the model can become more complex and modular so that it can predict numerical Karma scores accurately.

## References

- [1] Reddit. <https://www.reddit.com/about>, 2015. [Online; accessed May-2015].
- [2] Singer, Philipp, et al. "Evolution of reddit: from the front page of the internet to a self-referential community?." Proceedings of the companion publication of the 23rd international conference on World wide web companion. International World Wide Web Conferences Steering Committee, 2014.
- [3] H. Lakkaraju, J. McAuley, and J. Leskovec. What's in a name? understanding the interplay between titles, content, and communities in social media. In Seventh International AAAI Conference on Weblogs and Social Media, 2013.
- [4] Dean, Jeffrey, and Sanjay Ghemawat. "MapReduce: simplified data processing on large clusters." Communications of the ACM 51.1 (2008): 107-113.
- [5] Praw. <https://praw.readthedocs.org/>, 2015. [Online; accessed May-2015].
- [6] Socher, Richard, et al. "Recursive deep models for semantic compositionality over a sentiment treebank." Proceedings of the conference on empirical methods in natural language processing (EMNLP). Vol. 1631. 2013.
- [7] F. Bastien, P. Lamblin, R. Pascanu, J. Bergstra, I. Goodfellow, A. Bergeron, N. Bouchard, D. Warde-Farley and Y. Bengio. Theano: new features and speed improvements. NIPS 2012 deep learning workshop.
- [8] Manning, Christopher D., Surdeanu, Mihai, Bauer, John, Finkel, Jenny, Bethard, Steven J., and McClosky, David. 2014. The Stanford CoreNLP Natural Language Processing Toolkit. In Proceedings of 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations, pp. 55-60.
- [9] Chung, Junyoung, et al. "Gated feedback recurrent neural networks." arXiv preprint arXiv:1502.02367 (2015).
- [10] CS 224D: Deep Learning for NLP, Lecture Notes: Part IV. <http://cs224d.stanford.edu> [Online; accessed May-2015].
- [11] Hochreiter, Sepp, and Jürgen Schmidhuber. "Long short-term memory." Neural computation 9.8 (1997): 1735-1780.