Deep Classification and Generation of Reddit Post Titles

Tyler Chase tchase56@stanford.edu
Rolland He rhe@stanford.edu
William Qiu willqiu@stanford.edu

Abstract

The online news aggregation website Reddit offers a rich source of user-submitted content. In this paper, we analyze the titles of submissions on Reddit and build contextual models that learn the patterns of posts from different subcommunities, called subreddits. The scope of our project is twofold. First, we use post titles from 10 hand-selected subreddits and build a single-layer LSTM classifier model to predict the subreddit a particular title is from. Additionally, we implement a bot that is able to generate random post titles using LSTMs trained on each individual subreddit. Our classification algorithm performs quite well and achieves an average test accuracy of 85.6%. Our post generator had mixed results, with an average test perplexity of approximately 200 across the subreddits. Qualitative assessment of the generations demonstrate that our model outputs vaguely sensible results on average, with posts from certain subreddits being easier to generate than others. Though there is certainly room for improvement, we believe our novel results provide a good baseline that can be extended upon.

1 Introduction

Reddit is an online social news aggregation and internet forum. With over 540 million monthly visitors, 70 million submissions, and 700 million comments [1], Reddit is a rich dataset for various analyses. The site rewards interesting posts and users who submit them in the form of "karma", given by others who may choose to up-vote them. The site is also sectioned into various subcommunities, called "subreddits", each of which focuses on different topics, in which users post relevant content. To our knowledge, there has not been any work done with applying deep learning to Reddit, so this project presents a novel approach to the task.

For this project, we focus our work on semantic analysis of Reddit post titles, which effectively serve as headlines for submissions. First, we create a classification model that is able to determine the subreddit a particular post title is from. This has various practical applications; for instance, one can create a bot that looks at posts made in various subreddits, and comments a recommendation that the submission be posted to a different subreddit (if more appropriate). Alternatively, a real-time subreddit recommendation system can be created to help users find a subreddit to post to while they are in the process of submitting their posts. Subreddits would benefit from a larger quantities of relevant content, and users would benefit not only from larger amounts of "karma" for their posts, but also by being exposed to communities that are aligned with their interests.

Next, we build a post generation model that is able to randomly generate post titles for a particular subreddit. To achieve this task, we build separate language models to learn the contextual and syntactic structure of posts in different subreddits. The quality of a post title can often make or break the popularity of the submission. This post title generation model could help shed light on the types of wording and post structure that results in popular Reddit content.

[1] http://www.redditblog.com/2015/12/reddit-in-2015.html

2 Background and Related Work

2.1 Word Vectors

Most deep learning language models require some fixed representation of words to train on. Typically, words in the vocabulary are first converted to fixed-dimensional vectors that aim to capture semantic similarities and differences. Current state-of-the-art methods for generating such vectors include word2vec, a context window based model proposed by Mikolov et. al. [1], and GloVe, a global co-occurrence based model proposed by Pennington et. al. GloVe has the advantages of being consistently faster and providing better results [3], so we used this method to generate our word vectors.

The main idea behind GloVe is using global word co-occurrences to solve the following weighted least squares problem:

$$J = \sum_{i,j=1}^{V} f(X_{ij}) \left(w_i^T \tilde{w}_j + b_i + \tilde{b}_j - \log X_{ij} \right)^2 \tag{1}$$

where V is the vocabulary size, X is the co-occurrence matrix, f is the weight function, $W, \tilde{W}$ represent the word vectors for each word, and $b, \tilde{b}$ are bias terms for each word.

2.2 Recurrent LSTM Models

Long Short-Term Memory Models (LSTM), which extend the traditional recurrent neural network architecture, have been a staple method for training language models. Specifically, most previous work has used the sequence-to-sequence approach to train models that are capable of generating textual output, either in the form of novel new phrases or in translation tasks [6]. Specifically the model, when given a sequence of inputs $(x_1, x_2, ..., x_t)$, attempts to predict a sequence of outputs $(y_1, y_2, ..., y_t)$. The outputs, in the case of training to generate a sequence of text, become $(x_2, x_3, ..., x_{t+1})$; here, the sequence is padded with a <start> at x_1 and <end> token at x_{t+1}. Each LSTM cell is composed of the following equations:

$$\begin{aligned}
i_t &= \sigma(W^{(i)} x_t + U^{(i)} h_{t-1}) \\
f_t &= \sigma(W^{(f)} x_t + U^{(f)} h_{t-1}) \\
o_t &= \sigma(W^{(o)} x_t + U^{(o)} h_{t-1}) \\
\widetilde{c_t} &= \tanh(W^{(c)} x_t + U^{(c)} h_{t-1}) \\
c_t &= f_t \circ c_{t-1} + i_t \circ \widetilde{c_t} \\
h_t &= o_t \circ \tanh(c_t)
\end{aligned}$$

One of the main advantages of LSTM models over vanilla RNN models are their ability to persist and discard information over long time sequences via the input gate i_t and the forget gate f_t. A cell graphically showing this equation structure is shown on the left hand side in figure 1. In classification tasks, the outputs of each LSTM cell h_t have a linear transformation applied to them, followed by a softmax function in order to calculate the likelihood of a given outcome category.

3 Methodology

3.1 Dataset

The dataset we use comes from the Reddit Submission Corpus [2], which contains all reddit submissions (both posts and comments) from January 01, 2008 to August 31, 2015. The total number of subbreddits on Reddit exceed 1 million [3], most of which are too small to glean useful insights from; we therefore hand-select 10 popular subreddits to focus our work on. These subreddits are shown in table 1 along with brief descriptions of the kinds of content they contain. In order to generalize

[2] `http://files.pushshift.io/`

[3] `http://redditmetrics.com/history`

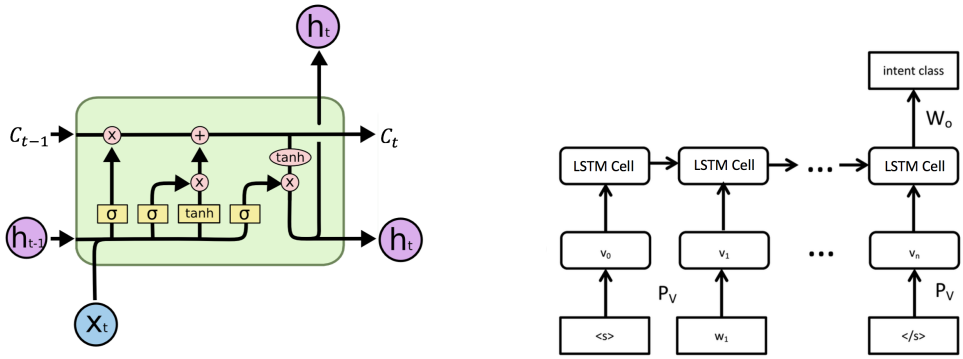

Figure 1: The left hand side shows a graphical representations of the equations representing an LSTM cell. The right hand side shows the structure of an LSTM with a classifier on the end.[2] [4]

Subreddit	Description
r / Askreddit	A place to ask and answer thought-provoking questions
r / LifeProTips	Tips that improve your life in one way or another
r / nottheonion	Real news stories that SOUND like they're satire articles, but aren't
r / news	News primarily relating to the United States
r / science	Latest advances in astronomy, biology, medicine, physics and the social sciences
r / trees	Anything and everything marijuana
r / tifu	Shared stories about moments where we do something ridiculously stupid
r / personalfinance	Personal finance questions and advice
r / mildinginteresting	Mildly interesting stuff
r / interestingasfuck	Very interesting stuff

Table 1: List of the 10 subreddits we used, along with their descriptions; these were used for both our classification and post generation models

the evaluation of model performance, we included both subreddits that are easy to predict as well as subreddits that can be easily confounded with each other. In addition, we only use posts in 2015, which is recent enough to provide a large amount of useful data, but not recent enough such that vote statistics have not stabilized. Moreover, we only choose the top 1,000 posts per month by upvote count for each sureddit, in order to filter out low-quality content. This results in 120,000 post titles in total, or 12,000 from each subreddit. Our final dataset simply contains the text of post titles along with the subreddit each title is from.

3.2 Reddit Post Categorization

In order to predict the subreddit origin of a post title we use an RNN that utilizes LSTM cells as shown in figure 1. This model takes in a sequence of words that compose a post title $(w_1, w_2, ...w_n)$, converts them to embeddings generated from our GloVe model $(v_1, v_2, ...v_n)$, feeds these as inputs to the LSTM cells and generates a subreddit prediction at the end of the series of LSTM cells as shown in figure 1.

For the reddit post generator, we followed previous approaches to language generation by training our data on a basic LSTM model. The general structure of the model is formulated as a sequential labeling task whereby the model attempts to label a word at time t +1, x_{t+1}, from a word at time t, x_t. The model is trained by minimizing the cross entropy cost of predicted and actual words. From multiple testing and implementations, we found that using a LSTM of hidden size of 200 to train on an input sequence length of 2 for 50 epochs performed the best in generating posts that are novel/interesting and comprehensible. We measured the performance of the model by measuring the perplexity of the model on a test set of post titles.

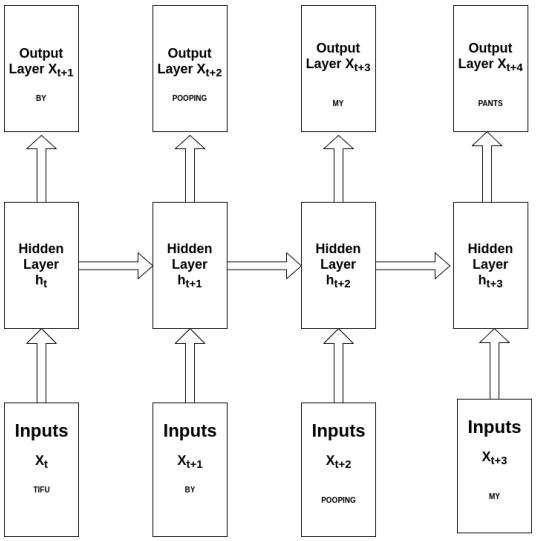

Figure 2: Basic structure of the LSTM RNN Network

At post generation time, we feed-forward a single token to our network to get the vector of probability distribution of succeeding tokens from the trained model. We then sample from the vector m words with the highest probability, weighting the choices by their likelihood of occurring to generate the next word. We continue this iterative process to generate new tokens from previous tokens until we reach an <end> token, at which point the sentence is complete.

For evaluation of the model, we use perplexity, which is a common measure used for assessment the performance of language models [5]. Intuitively, this metric measures of how accurately our model is able to predict a sample sequence of words. However, this doesn't capture the full extent of our objective, which is to generate titles that sound reasonable and pertain to the subreddit topic. Unfortunately, there is no good quantitative metric that captures this qualitative idea well – consequently, human judgement is required to get an idea of how well our model performs. Therefore, we created a rating system (Table 3) to assess the quality of each generated title, and hand-annotated a sample of our generated titles. We also used our classifier to classify a sample of posts generated by our post generator to see how closely the generated posts stick to topic.

4 Experiments

4.1 GloVe Vectors

To train our GloVe vectors, we used a corpus of all post titles from the top 50 subreddits by subscribers over the past year, as well as our subreddits considered in the reddit classification.[4]. This resulted in approximately 9.5 million post titles, from which we trained our vectors. We tokenized the corpus by including contiguous sequences of letters (and dashes/hyphens if they occur inside a word), as well as punctuation. Our total vocabulary size consisted of approximately 850,000 tokens.

We used our own implementation of GloVe to create 200-dimensional embedding vectors, using the same hyperparameters as described in the original paper [3]. This is necessary because Reddit contains many words that are unique to it's subreddits. For example tifu is a word used in almost every post in the tifu subreddit. We use vanilla gradient descent instead of adagrad, due to faster training times, and ran it for 75 iterations. Furthermore, we also perform GPU optimizations with CUDA in order to make our code run faster.

[4] as indexed at `http://redditlist.com/`

Subreddit	Perplexity
AskReddit	216.148
LifeProTips	143.067
nottheonion	210.107
news	209.052
science	199.246
trees	313.366
tifu	120.212
personalfinance	215.410
mildlyinteresting	156.187
interestingasfuck	206.33

Table 2: Test Perplexity by Subreddit

Rating	Description
1	Complete gibberish or indecipherable text
2	Minimal grammatical structure or completely off-topic
3	Some relation to subreddit topic, many grammatical mistakes or inconsistencies, meaning is vaguely decipherable
4	Moderate grammatical mistakes or mild inconsistencies in the meaning of the title
5	Reasonable post in subreddit, on-topic and minimal grammatical mistakes

Table 3: Rating system used for annotating our post generations

4.2 Reddit Post Categorization

For predicting the subreddit origin for a post title we implemented a LSTM of length 20 and depth 1. This model contains a 200 dimensional hidden layers. During training optimization is carried out over 10 epochs with a batch size of 100 posts. The model is trained on 80% of the 120,000 post titles, with 10% of the posts left for optimizing select hyper-parameters, and 10% for final testing.

Hyper-parameters for the dropout rate and the learning rate are optimized as shown in figure 5. We determine the optimal dropout rate to be 0.55 (with initial learning rate of 0.003) by scanning between 0 and 1 in 20 steps. Then we determine the optimal learning rate to be 0.005 by scanning between 0.001 and 0.040 in 20 steps.

4.3 Reddit Post Generator

To evaluate the post generation model, we first examined the test perplexity of the model for each subreddit, the results of which is presented in Table 2. The average perplexity hovers around 200. This number is somewhat misleading because it does not really tell us about how comprehensible newly generated posts would be. Thus, for qualitative assessment of the generator, we attempt to measure how well the post generator performed by letting our post classifier classify 100 randomly generated posts for each given subreddit. Because our classifier performs relatively well on new data, whether or not it can correctly classify our generated posts will serve as a good indicator of post generation success. In particular the classification model may capture tokens and structure characteristic of a particular subreddit. We also hand-annotated a sample of generated posts using the evaluation metric presented in Table 3. From these evaluations, the final model we decided on was trained using a hidden layer of size 200, 0.0003 learning rate, and no drop out, on 90% of the data for each subreddit, using 10% for evaluating test perplexity.

5 Results

5.1 GloVe Embeddings

We can qualitatively evaluate the performance of our embeddings by plotting select words on a 2-D plane. To do this, we perform a singular value decomposition on the embeddings and take the first

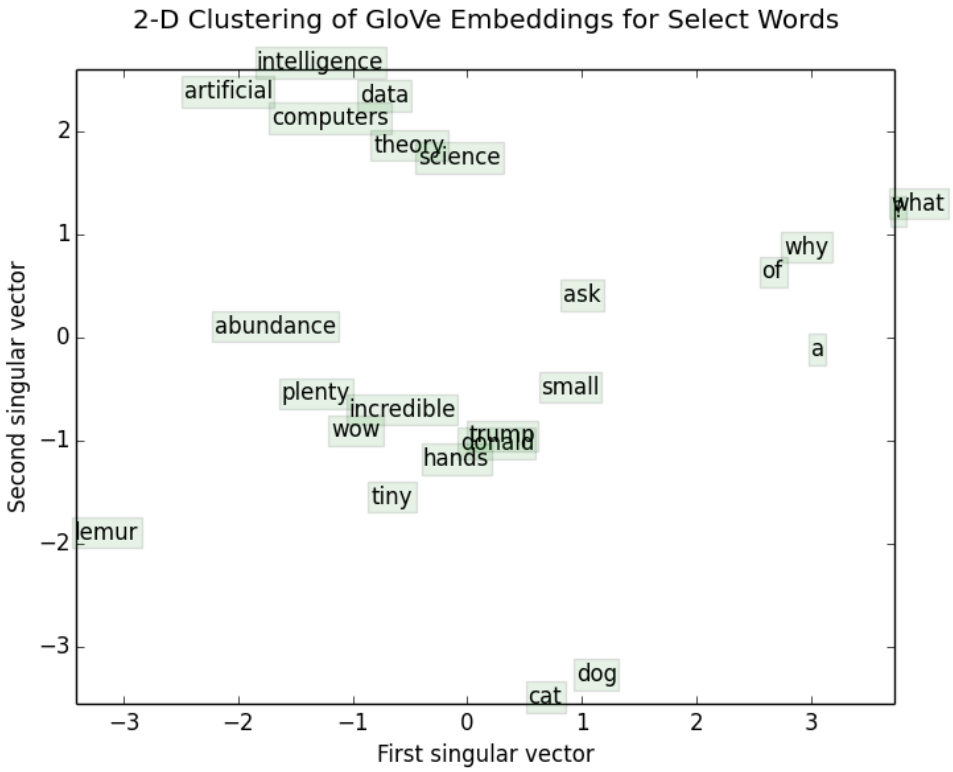

Figure 3: Plot of 2-D representation of embeddings for 24 select words

2 singular vectors as the axes to plot against. Finally, a group of 24 select words were chosen to be plotted – the result is shown in figure 3. Some notable groupings include the words [artificial, intelligence, data, computers, theory, and science], [dog, cat], and [donald, trump, tiny, and hands], which are clusters we would expect. We also examined the nearest neighbors for a few words to further verify the accuracy of our embeddings Table 6 (located in the Appendix).

5.2 Reddit Post Categorization

After training our model on the training data and adjusting our two hyper-parameters of interest (dropout rate and learning rate) on the developement data we then test our categorization model on the test data. The model acheived a training accuracy of 90.9% and a test accuracy of 85.6%. The confusion matrix of the model predictions on the test set can be seen in figure 4.

Some reddits that are predicted very well are r/AskReddit, r/LifeProTips, and r/tifu. This is expected because these subreddits have tokens that unique to their posts. r/AskReddit is mostly composed of questions and often contains the token ”?” at the end of a post. r/LifeProTips often contains the token ”LPT:” at the front of the post. r/tifu often begins with the two tokens ”TIFU” and ”by”. These subreddits serve as a sanity check for the algorithm since conventional machine learning methods could most likely do well in categorizing them.

We have two pairs of subreddits that we anticipated significant confusion for and for these subreddits our algorithm did surprisingly well. The first pair of subreddits is r/nottheonion and r/news. r/nottheonion contains posts about real news stories that sound like they are satire but aren’t, while r/news contains posts with all kinds of news. Our classification algorithm is able to correctly classify r/nottheonion posts 77% of the time, and correctly predict r/news posts 68% of the time. We don’t view this as too worrisome, considering many r/nottheonion post titles could very well be on r/news as well – indeed, a human often would have trouble accurately classifying some of the confounded posts.

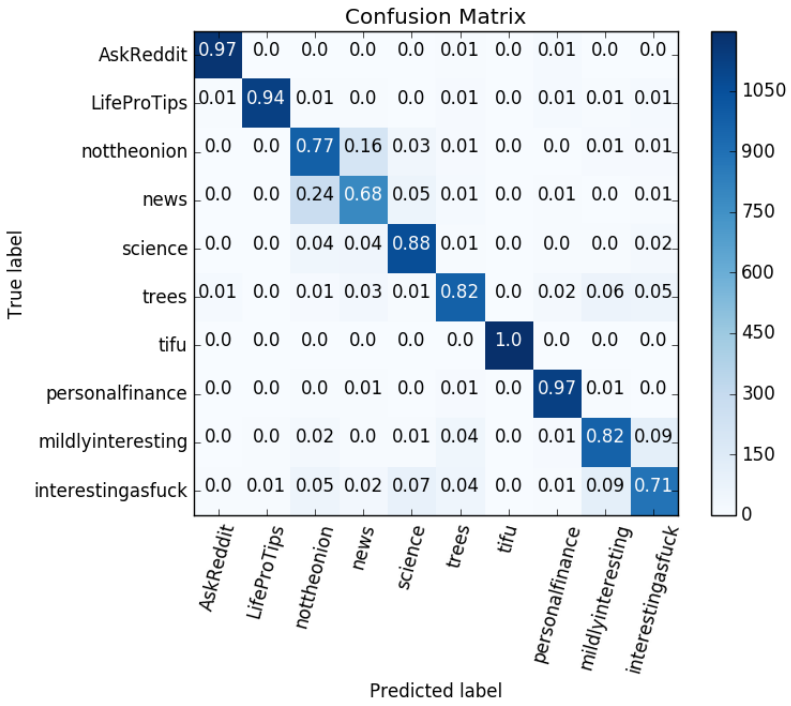

Figure 4: Confusion matrix for our classification model

The second pair of subreddits we anticipated significant confusion for were r/mildlyinteresting and r/interestingasfuck. Our classification algorithm did surprisingly well. It correctly classified posts from r/mildlyinteresting 82% of the time and correctly classified posts from r/interestingasfuck 71% of the time.

5.3 Reddit Post Generation

Overall, the model had an average test perplexity of around 200 across the different subreddits. However, this does not provide a great indicator of how good the posts are qualitatively in terms of comprehensibility. Also, because of the large differences in the grammatical and semantic complexity of posts across subreddits, the model performed drastically different in terms of generating comprehensible posts across them. To make up for this flaw in evaluation, we adopted a novel approach in determining the overall quality of generated posts. Specifically, we first generated 100 posts per subreddit and evaluated them by feeding them into our trained classifier. The classifier was able to categorize the generated post correctly 81.8% of the time. This is only 3.8% less than our test accuracy of the categorization model on actual reddit post titles. This suggests that our post generation algorithm is capturing contextual information with reasonable success. Although, as noted earlier this says little about syntactical or semantic success in generation. Second, we utilized hand annotation and assigned a score of 1-5 in terms of comprehensibility on a sample of generated posts produced by our generator. We averaged the average score for each subreddit across the 3 human coders to generate the final score, which is presented in Table 4.

Table 5 presents a sample of posts generated by our post generator for each subreddit, organized by good and bad. Immediately we see that there is a noticeable difference in the comprehensibility of posts across the subreddits. It is clear that for subreddits where posts tend to follow a rigid structure (/r/tifu or /r/AskReddit), the post generator was able to generate some comprehensible posts. However, for subreddits that have more complex language structures/greater variations in syntactical structures (/r/nottheonion or /r/news), the model performed more poorly. One obvious reason for this is that because the model attempts to predict the next word with only the previous word, for post

Subreddit	Average Rating	Rank
mildlyinteresting	2.40	5
science	2.76	2
interestingasfuck	2.27	7
trees	2.38	6
personalfinance	2.60	3
AskReddit	2.47	4
LifeProTips	2.13	8
nottheonion	1.96	9
news	1.82	10
tifu	3.44	1

Table 4: Average ratings for our annotations on the sampled generations for each subreddit. Rank represents the ordering of subreddits that provided the most reasonable predictions.

titles that have more complex structures, it cannot easily capture or retain context/structure past the first preceding word. In fact we can see that the context quickly shifts after the next word is generated. One possible fix for this problem is to use an n-gram approach whereby we use a sequence of words to predict the next word or next sequence of words, so that more contextual information is retained across multiple words. The quality of these posts also reflect the overall comprehensibility scores from hand annotations.

6 Conclusion

Our classification model performed reasonably well and exceeded our expectations. It is able to learn the patterns of post titles with a simple, rigid structure extremely well; moreover, it also is able to correctly classify a large majority of post titles that don't adhere to a fixed structure. In addition, despite some classification confusion between similar subreddits, the model still manages to classify most post titles.

Our post generator, however, had more mixed results. Being a much more difficult task, subreddits that have clearer syntactical structures typically resulted in better generated posts. However, the results are poorer for subreddits that have more complex structures or have greater variation in sentence construction overall. In the end, our model on average is able to generate vaguely sensible results, though nowhere near good enough to match the quality of titles created by actual people.

Future work should consider the incorporation of additional features such as using n-grams as inputs, as well as using attention mechanisms to account for a larger contextual window. In addition, using more sophisticated state-of-the-art language models such as variational LSTM and CharCNN can help improve performance. Finally, hyperparameter tuning can also be optimized using Bayesian methods, which is significantly better than the grid search method we used.

Acknowledgements

We would like to thank Danqi, our project mentor, for her guidance and help in answering many of our questions. Wed also like to thank Microsoft Azure for providing us with GPU computing time for training and testing our models.

Appendix

Subreddit	Good Posts	Bad Post
mildlyinteresting	i saw an illusion of my thumb my apple looks like a picture.	the sun. 2 p. this . same still had for 15 years.
science	the brain can predict climate. scientists have discovered an exceptionally luminous galaxy around the universe	a drug . it is associated with their pregnancies.
interestingasfuck	how to be very interesting	a tribal ceremony at a toast to 1, and remains untouched to it gets an iphone 6, and it takes it
trees	i'm stoned. my new lighter.	when prosecution man and enjoy i had to my life.
personalfinance	can help me make more, i need advice.	my money. how to get a ton of my life?
AskReddit	what is the world and what is acceptable?	what are it like, but would you get $100k on the final person or your life?
LifeProTips	lpt : how to avoid your heart.	lpt: if you don't want about them back up and they are in them.
nottheonion	man arrested for a day	texas high school, but fails for thinking he told a cabinet, hiding from the energy from the sun
news	police chief has been disciplined in the largest ev.	ohio in u.s.
tifu	tifu by having sex. [nsfw] tifu by having a baby. tifu by going to a war. tifu by almost using reddit.	tifu - nsfw) 20 . (nsfw] slightly slightly nsfw]

Table 5: Sample of posts generated by the LSTM post generator

science	news	fitness	glove
scientific	cnn	gym	gloves
scientist	headlines	workout	compartment
studies	newspaper	bodybuilding	first
physics	updates	exercise	hoodie
research	reporter	weight	t-shirt
technology	latest	routine	pac
psychology	media	workouts	assorted
fiction	fox	lifting	logo
scientists	tv	trainer	striped
engineering	bangladesh	motivation	latex

Table 6: Top 10 nearest neighbors in the embeddings for select words

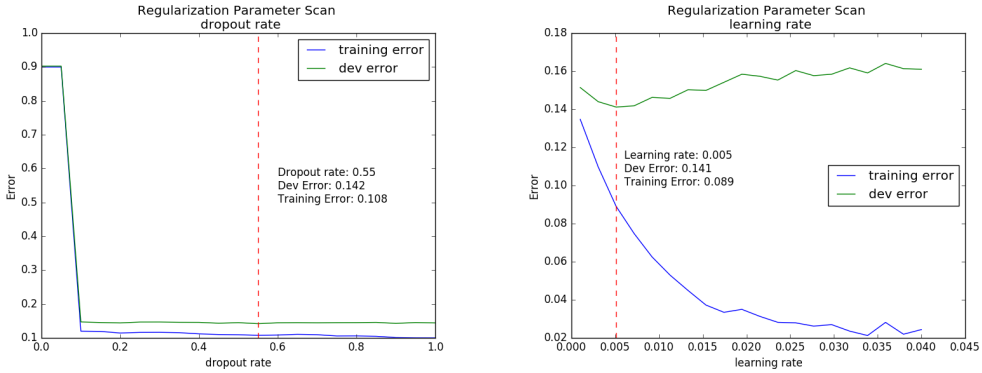

Figure 5: Hyperparameter tuning

References

[1] Tomas Mikolov et al. "Efficient Estimation of Word Representations in Vector Space". In: *CoRR* abs/1301.3781 (2013). URL: http://arxiv.org/abs/1301.3781.

[2] Christopher Olah. *Understanding LSTM Networks*. http://colah.github.io/posts/2015-08-Understanding-LSTMs/. Blog. 2015.

[3] Jeffrey Pennington, Richard Socher, and Christopher D Manning. "Glove: Global Vectors for Word Representation." In: vol. 14. 2014, pp. 1532–1543.

[4] Suman Ravuri and Andreas Stolcke. "A comparative study of recurrent neural network models for lexical domain classification". In: *Acoustics, Speech and Signal Processing (ICASSP), 2016 IEEE International Conference on*. IEEE. 2016, pp. 6075–6079.

[5] R. Rosenfeld. "Two decades of statistical language modeling: where do we go from here?" In: *Proceedings of the IEEE* 88.8 (Aug. 2000), pp. 1270–1278. ISSN: 0018-9219. DOI: 10.1109/5.880083.

[6] Ilya Sutskever, Oriol Vinyals, and Quoc V Le. "Sequence to sequence learning with neural networks". In: *Advances in neural information processing systems*. 2014, pp. 3104–3112.