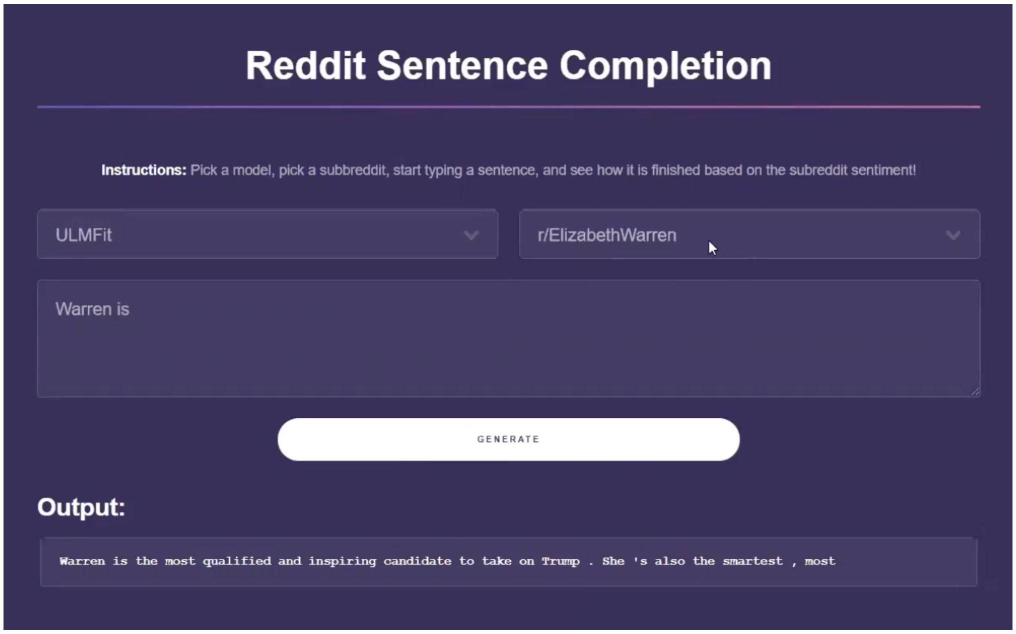
Reddit Auto-Moderation by Evaluating Community Opinion

Course project for CSE 6240: Web Search and Text Mining, Spring 2020

Ayush Baid, Ankur Bhardwaj, Tarun Pasumarthi



Reddit Sentence Completion

Instructions: Pick a model, pick a subbreddit, start typing a sentence, and see how it is finished based on the subreddit sentiment!

ULMFit

r/ElizabethWarren

Warren is

Output:

```
Warren is the most qualified and inspiring candidate to take on Trump . She 's also the smartest , most
```

Figure 1: Our final application. There are two dropdowns which allows the user to select the model (ULMFit) and the subreddit (r/ElizabethWarren) to analyse. The user enters the prompt (Warren is) to generate the model completed output, which they can analyse.

1 ABSTRACT

Platforms like Reddit have become a host of contentious political arguments and sometimes go extreme with radical opinions/threats. It is a big challenge for companies to moderate discussions and keep forums free from hate-mongering and divisive language. Our objective is to create a moderation system which will leverage machine learning with human expertise to take decisions. The machine intelligence system learns the views of a subreddit and auto-completes sentences related to any subject (like "Trump...", "Jews...", etc.). The completions are then evaluated by a human expert to take final decisions. We train our opinion models on comments from 14 subreddits and use them to complete a corpus of 350 phrases. Our proposed use of state-of-the-art NLP techniques using transfer learning learns the context and opinions for different topics and has a high usability compared to baselines, and this method is more insightful and requires no hand-crafted rules like other auto-moderation techniques.

2 INTRODUCTION

People on social media are more divided along political beliefs and ideologies than ever before. Platforms like Reddit, Facebook and Twitter have experienced of influx of radical and extreme views, echo-chambers and negative commentary about the people who do not agree with their views. Moreover, these platforms are becoming sites of contentious political arguments and sometimes go extreme with death threats and radical opinions. Amidst vast growth of social media communities, it is becoming more and more challenging for companies to moderate discussions and keep forums a place for open and honest conversations.

Reddit is organised into more than a million communities called subreddits, where subscriber count reach as high as twenty million. Each subreddit is devoted to a different topic - for example: r/Soccer, r/DonaldTrump, r/News and many more. Some of these subreddits are severely political, polarized and become echo chambers of radical opinions: r/TheDonald, r/OurPresident, r/YangForPresidentHQ.

We introduce a novel moderation technique of a mix of machines and human expertise: a computing system which parses large datasets of comments to learn the opinions and beliefs of a subreddit. The computing system is then used by a human moderator summarize the opinions about polarizing topics.

This problem can be re-interpreted as contextual sentence completion. To build our dataset, we gathered 100k comments from 14 different subreddits that represent various parts of the political spectrum. We then fed the data from each of these subreddits into our two baseline NLP sentence completion models, a fixed window language model and an RNN language model. To evaluate our baseline, we fed 350 politically polarizing phrases into the sentence completion model for each of the 14 subreddits, and then manually computed the usability percentage of the outputted sentences. We then trained our data on the GPT-2 and ULMFit models, which contain millions of parameters and are pre-trained on huge datasets. We computed the usability percentages by using the same method that we implemented on the baseline models.

As our results would show, the GPT-2 model had the best usability percentage at around 50 percent. This was followed by the ULMFit model and RNN baseline, both of which had about 10 percent usability percentage. Lastly the fixed window baseline had the worst usability percentage at about 1 percent.

Furthermore, to provide an intuitive understanding of the performance of each model we developed an interactive web application. Figure 1 provides the screenshot of this application in which the user can see the how a specific model completes an inputted phrase for a selected subreddit.

By incorporating this computing system on a larger scale, moderators could gauge the health of a subreddit as a whole before delving into individual comments, thereby saving countless hours manual effort.

3 RELATED WORK

3.1 Auto-Moderation

Currently, content regulation on Reddit is a socially distributed endeavor in which individual moderators coordinate with one another as well as with automated systems [3]. These automated systems, or automods, filter out individual comments by looking for user generated regex patterns. Not only is this approach reliant on human moderators to come up with these expressions, but it provides a very limited rule based approach that is expensive and not feasible at a massive scale.

Machine Learning based automoderator tools, like Washington Post's ModBot [4], bridge the scalability issue, but still relies on a combination of manually crafted rules and traditional NLP techniques to create a binary classification model which again doesn't provide insights and act as a blackbox tool for moderators.

Furthermore, there is no moderation tool which works on a community as a whole and can condense petabytes of user's posts. During recent years, Reddit has been slow to shut down subreddits even after numerous user complaints.

3.2 Text Generation

Since 2013, word embeddings pre-trained on algorithms like word2vec [6] and Glove [7] have been used for all NLP related tasks to initialize the first layer of a neural network. Though these pre-trained word embeddings have been immensely influential, they have a major limitation: they only incorporate previous knowledge in the first layer of the model—the rest of the network still needs to be trained from scratch which is difficult in sparsity of labeled data. These embeddings also struggle against the large variations of the language as a same view can be expressed in numerous different styles.

In last few years, there have been a lot of advances in NLP with more complex networks, however they haven't been widely used in moderation of social media communities. Networks like GPT-2 [9] demonstrate successful learning of sentence semantics and syntactic concepts to generate text which matches human performance in some cases and we have explored there application for our use case.

4 DATASET DESCRIPTION

To create our dataset, we wanted to aggregate the comments of multiple popular and controversial political subreddits. We tried to mantain the balance in the dataset and the 14 subreddits we chose for this project comes from a spectrum of topics and ideologies, as described in table 1.

Left	Right	Mixed	Apolitical
r/democrat r/ElizabethWarren r/Impeach_Trump r/OurPresident r/SandersForPresident r/The_Mueller	r/Conservative r/libertarian r/The_Donald	r/news r/politics	r/aww r/soccer r/YangforPresidentHQ

Table 1: Subreddits used: categorized by their political leaning. We use the name and description of the subreddit to categorize them

4.1 Source

We got our data from PushShift, which is a Reddit crawler that saves copies of objects (threads and comments) and their meta information. Our primary date range was to fetch 20,000 comments each month from September 2019 to December 2019 to obtain a total of 100,000 comments. This was done to ensure that we have comments from a variety of threads from different time periods. We created numpy arrays for each subreddit and the query month.

4.2 Data Pre-processing Steps and Explanation

We followed the following pre-processing steps for baseline approaches:

- Break comments into sentences using the Punkt tokenizer
- Remove the user and subreddit internal links, common on Reddit
- Remove characters which are not English alphabets
- Convert to lower case
- (Optional) Remove stop words
- (Optional) Add <eos> token (end of sentence token)

We are operating on a limited resource budget and hence we decided to limit the number of comments we use in this project. We started with 50,000 comments and then increased it to 100,000 as we needed more data for our models.

4.3 Raw Data Statistics

We used pandas to run some basic count statistics to generate 3 stats on the dataset. The results are in table 2. Also, we analysed most frequent words as well as most important tf-idf tokens for each subreddit. Result for one of the topics are in Figure 3.

4.4 Data Analysis

We used multiple techniques to perform Exploratory Data Analysis on our dataset in order to drive intution and formulate testable hypothesis. As we used sentences from 100k comments for training

Subreddit	#Sentences	Vocab-Size	Avg tokens/sent.
r/aww	182,296	43,068	8.83
r/Conservative	305,443	54,651	13.92
r/democrat	1,857	4,246	14.33
r/ElizabethWarren	255,458	42,049	16.15
r/Impeach_Trump	95,563	31,274	12.97
r/OurPresident	189,939	36,070	13.08
r/libertarian	359,008	55,875	14.62
r/news	268,829	52,733	14.07
r/politics	283,121	51,272	13.84
r/SandersForPresident	315,289	50,832	13.95
r/The_Donald	269,942	50,958	12.89
r/The_Mueller	320,194	50,337	12.39
r/soccer	207,777	47,368	12.07
r/YangForPresidentHQ	360,995	50,137	14.99

Table 2: Count-based stats on raw data computed in Pandas

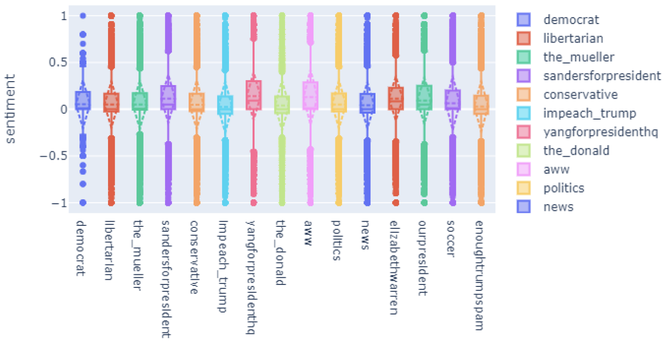


Figure 2: Sentiment Analysis of Subreddits using Textblob

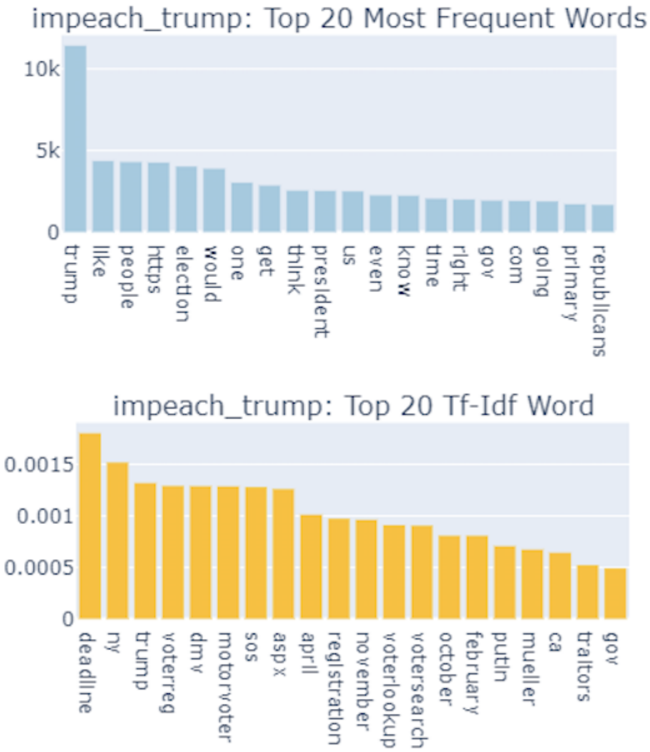


Figure 3: Most common words vs top tf-idf terms for r/Impeach_Trump Subreddit

our baseline models it was important to check if we have similar number of sentences for each corpus. Also, for training the LSTM baseline model, we have taken back propagation window size of 10 i.e. the model learns the weight parameter using hidden state context from last 10 words and hence we made sure that the average number of tokens in each training imput in greater than 10 as can be seen in table 2. Other important feature was the vocabulary size which showed normal distribution centered around mean of 43000 with a lot of overlap among political and non-political topics which helped in generating contrasting outputs using similar tokens.

Using Textblob sentiment analysis, we observed that except for few subreddit datasets all have neutral sentiment with average polarity close to 0. While other subreddit topics like *aww*, *YangforPresidentHQ* showed positive average sentiment score and *Impeach_Trump*, *The_Donald* showed slight negative sentiment. In order to understand the the distribution of common words, we found most frequent words used in all subreddits - for instance *Impeach_Trump* has *trump*, *elections* etc. as common words. Moreover, the tf-idf analysis of the corpus helped us find words responsible for the negative sentiment in few subreddit- for ex. *deadline*, *traitors* were among the most relevant tokens in *Impeach_Trump* subreddit.

5 EXPERIMENTAL SETUP

Data split: We split the comments obtained from each month and each subreddit into a 70-30 split to create train and validation sets.

We designed 350 phrases manually (link) which will be fed to the language models to produce completed outputs. These outputs will help a human/machine moderator to take decision and provide the insight behind why a subreddit is being classified as toxic/harmful.

Training problem and metrics: For the training, we posed the problem as next word prediction and hence used cross-entropy loss across training and validation set for this milestone.

As our problem to feed manual phrases to the language model and obtain a word/sentence output is a new one, there is no inherent ground truth. Conceptually, we are training our language models to complete the sentences for a subreddit and repurpose the model to complete brand new phrases. We hope that the completed sentences will be consistent with the opinion of the subreddit. This is a novel problem formulation and hence we cannot use the common metrics of classification tasks like AUC-curves and accuracy.

To tackle this difficulty, we came up with a new metric where we manually review the completed phrase from the models and tagged it as *useful* or not *useful*. We then get the ratio of useful completions from a model trained for a particular subreddit and that serves as the

usability metric for our project. We use personal experience from using Reddit and visiting the 14 subreddits to judge if a particular phrase completion is consistent with the general discussions and ideology of the sub.

Here are a couple of examples to illustrate our metric:

- For subreddit the_donald, a phrase "Trump administration has something" is categorized as *not useful*
- For subreddit politics, a phrase "Trump should be impeached" is categorized as *useful*
- For subreddit SandersForPresident, a phrase "Sanders should be impeached" is categorized as *not useful* as it is not a real possibility and is also not something which is discussed in a pro-Sanders subreddit.

System Configuration: We ran some aspects of the work like dataset scraping and simple data analytics described till this point on Google Colab, which provides 16GB of RAM for a session. The machine learning models were trained on the cluster made available to Ayush by his advisor. The specs of those machines are variable, but they generally have GPUs with 12-15 GB memory, Intel Xeon CPUs with around 50 cores, and upwards of 350 GB of RAM.

6 BASELINES

We tried two different baseline language models based on neural networks and we will describe the details and the results in this section. Before diving into neural models, we tried building a very simple language model based on n-grams where we computed probability of next word given bi-gram words in the whole corpus i.e. n-grams model with n=2. This model had two main problem while choosing larger values of n - firstly sparsity of exact same words as in the query lead to bad results and secondly there is no understanding of semantics. Thus, to solve these issues we looked into neural based language models.

We use two simple and commonly used language models for this project. The first one is the fixed-window based language model and the second is an RNN-based language model.

6.1 Fixed-Window Based Language Model

This language model was proposed by Bengio et al. [1] in 2003. It has a very simple architecture composed of just fully connected layers. The model has a fixed context window which is embedded into a feature space using the word embeddings layer. All the embedding vectors resulting from the context are concatenated into a single big vector and the fully connected layers of the model have to process this big vector to predict the next word. An example with single layer model is provided in figure 4.

This is a very limited model, as it does not have sufficient complexity to learn about the meaning of different words, the semantic and contextual understanding of the language. It suffers from the sparsity of the training data, and the fact that there is no learning of context within the model. However, it also offers an advantage that it does not need huge volume of training data.

As we have a limited training data compared to deep-learning standards, we limit the context window size to 2. We also use a pretrained word2vec model [5], a 300 dimension embedding space learnt on google news. The pretrained embeddings will provide us a lot of similarity associations between words instead of training it from scratch.

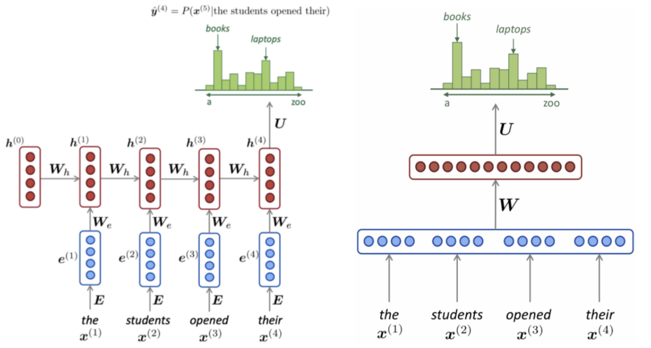


Figure 4: Left: LSTM and Right: Fixed-window based language model

Credits: Stanford 224n class

6.2 RNN based language model

Language has an inherent temporal dimension. As a result, the introduction of RNNs [11] has been highly beneficial to the NLP community. RNNs can be used to complete sentences but they are not restricted by the fixed context size like the previous baseline.

We follow the LSTM based recurrent language model [8] and modify it to better suit our needs. The ‘smaller’ version in their paper has 2 LSTM layers and a hidden space dimension of size 200. They also propose to tie the weights of the encoder and decoder in the network. We replicate this model architecture.

For training, we use the Stochastic Gradient Descent optimizer with gradient clipping. We train the model for each subreddit for 40 epochs and using the back-prop through time (BPTT) value to 10. This choice was made after considering the average number of tokens presented in the data analysis section.

7 PROPOSED METHOD

Our approach is to make text completion language models based on the comments of each subreddit such that we can generate opinions about any subject based on the discussion. As the discussion among users is generally about a variety of topics, the ability of the model to complete a sentence about a particular person/event can be considered a good proxy for the *opinion* about the topic. The flowchart of our approach is presented in figure 5

In recent few years, NLP has seen an explosion in the deep-learning based language models. The current state-of-the-art models have millions of parameters and are very good at understanding semantic structure of the language, context, emotions, and complex constructs like sarcasm. We plan to use two models: ULMFit [2] and GPT-2 [9]. Instead of training from scratch, we follow the recent trend of transfer learning in NLP [10]. The use of transfer learning is a game changer as it allows quick fine-tuning on smaller datasets.

In our case we only have 100k comments for training each subreddit language model which is quite sparse, so in ULMFit approach we initially fine tune just the last 2 layers of language model which has already been pre-trained using wikitext-103 dataset. Further,

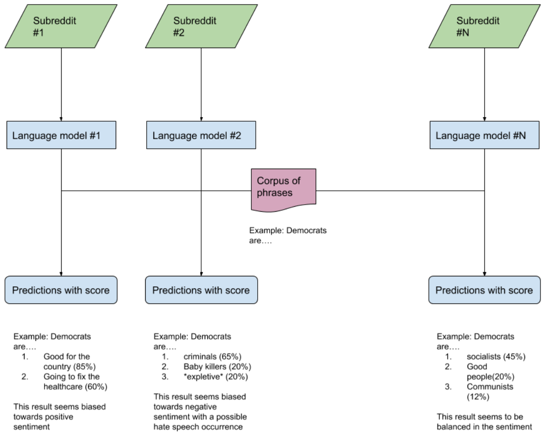


Figure 5: Proposed approach of using Deep-Learning based language models for each subreddit, which will help complete the phrases designed by the moderators. The moderators will then review the sentence completion and take actions based on the summary of generated text

we unfreeze full network and run few epochs to further fine tune wikipedia language model for our own task. GPT-2 is a casual unidirectional transformer network trained on a huge text dataset called WebText, which has data scrapped from the websites and the size is around 40GB. Even the smaller version of this model has 117 million parameters. We use HuggingFace's implementation to fine tune the model for 3 epochs.

In the baseline approach of fixed window language model, the number of parameters in the model scales with the n-gram size and thus the length of the history captured. Moreover, the n-gram history is finite and thus there is a limit on the longest dependencies that can be captured. A big RNN/LSTM based model has a lot of parameters, and training it from scratch requires a lot of training data. Our transfer learning approach allows us to use very complex models with limited unlabelled dataset.

Our choice of models (GPT-2 and ULMFit) have proven to be better at understanding the language semantics and are good at text generation. As these methods have gigantic number of parameters. Hence, we will use transfer learning and use pre-trained models on large datasets and fine-tune the last couple of layers on comments from each subreddit. Our approach of training the model for an intermediate task of sentence completion instead of a classification tasks like sentiment/polarization detection enables us to use unlabelled data and makes our approach highly scalable to millions of comments from all the subreddits without any extra adjustment. This is an improvement over other automoderation techniques.

8 EXPERIMENTS

We train our moderation system with two baselines (fixed window model and rnn model) and our final two NLP models (GPT-2 and ULMFit) and deploy them to complete the corpus of phrases. Sample outputs are presented in table 3. The usability scores for the 4 methods across three subreddits are presented in table 4.

Input	Subreddit	Model	Output
Trump is a	r/politics	FixedWindow* RNN ULMFit GPT-2	supporters traitor Russian puppet evil puppet
Trump is a	r/the_donald	FixedWindow* RNN ULMFit GPT-2	train cuck handling ISIS smarter than the Dems
Biden should	r/politics	FixedWindow* RNN ULMFit GPT-2	Warren should not be the nominee Sanders be in the Senate if he had a shot at the nomination
Biden should	r/the_donald	FixedWindow* RNN ULMFit GPT-2	Trump be the same be the president be on the cutting edge of all of this
UBI	r/politics	FixedWindow RNN ULMFit GPT-2	Trump is a good thing not a job should not be an auto--matic negative incentive
UBI	r/the_donald	FixedWindow RNN ULMFit GPT-2	dr *null* is socialism ! I need an ANTIFA card!

Table 3: Sample phrase completions. FixedWindow* means that we have removed stopwords from the input (is, a, should)**.**

Method	politics	the_donald	our_president
FixedWindow	1.34%	0.68%	0.73%
RNN	13.08%	10.82%	8.70%
ULMFit	8.17%	18.31%	6.48%
GPT-2	**55.35%**	**51.68%**	**33.49%**

Table 4: Usability Metric for baselines

8.1 Result Discussion

The results from the fixed window model were unsatisfactory in terms of syntactic and semantic rules of the generated english text. For ex. trained on r/politics, it predicts the words 'trump' and 'people' for most of the input prompts because of count based design and thus is not useful at all for our intended use case.

The LSTM based model, these models perform very well in understanding the dependency in the sentence structure. However, it struggles to keep the understanding of the subject till the end and makes conflicting predictions for the same phrase. It also repeats the same predictions for different subjects. The examples show that the model identifies *Trump*, *Bernie* in the input query as Person and then predicts next words as person's quality like *liar*, *honest*, *criminal* with proper prepositions in place; however the capability to predict these qualities in specific person's context is still wanting. Few results are also not usable: r/The_Donald is a pro-trump subreddit and our outputs are negative for trump. Similarly, r/Our_President will not have positive opinion of Trump.

In order to make language models learn more details about specific subjects using small datasets as ours, ULMFit and GPT-2 architectures work very well. For instance, ULMFit relates *UBI* (which refers to the Universal Basic Income scheme) with no Job which is an excellent correlation but fails to create a legitimate story while generating text. GPT-2 model on the other hand relates *UBI* with negative incentive and also generates valid text by using words like *should not* and *automatic* in the sentence.

9 CONCLUSION

Deep Learning in NLP has given promising approaches for solving classification, text-generation, question answering problems using transfer learning and this has been of great importance because of sparsity of labeled datasets for most of these problems. The simple objective of modelling the next word given the observed history contains much of the complexity of natural language understanding which we have explored in the project and have seen promising results. We observed that if language models have more context they are able to use knowledge of both language and the world to heavily constraint the distribution over the next word and produce logical phrases about specific subjects.

It could be better for moderators if instead of just predicting the output these models should have the capability of suggesting the confidence level in dependency parsing of the generated sentence. Our approach has a problem that we need to store a model for every subreddit. This means this has poor scalability and cannot be used to deploy for the complete website. There can be some work done under the conditional model learning approach or figuring out model similarity between different subreddits so that the number of models has sub-linear complexity.

10 CONTRIBUTION

In the first phase of the project we chalked out three main tasks which included data exploration and scraping; building baseline model pipelines and finally training validating models across each subreddit. Each of us contributed in all three aspects as we divided the workflow in parallel to finish each task before moving onto the next one. Later Tarun worked on building an application to showcase the capability of generating text using multiple models we trained for different subreddits, while Ayush and Ankur worked on improvising upon the baseline appraoches using GPT-2 and ULMFit architectures.

11 ACKNOWLEDGMENTS

We used the following open-source code base for our project

- RNN based language model: we started with Pytorch example and made minor modifications.
- FastAI's ULMFit code
- HuggingFace's GPT-2 code

REFERENCES

[1] Yoshua Bengio, Réjean Ducharme, Pascal Vincent, and Christian Jauvin. 2003. A neural probabilistic language model. *Journal of machine learning research* 3, Feb (2003), 1137–1155.

[2] Jeremy Howard and Sebastian Ruder. 2018. Universal Language Model Fine-tuning for Text Classification. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. 328–339.

[3] Shagun Jhaver, Iris Birman, Eric Gilbert, and Amy Bruckman. 2019. Human-machine collaboration for content regulation: The case of Reddit Automoderator. *ACM Transactions on Computer-Human Interaction (TOCHI)* 26, 5 (2019), 1–35.

[4] Ling Jiang and Eui-Hong Han. 2019. ModBot: Automatic comments moderation. In *Proceedings of the Computation+ Journalism Symposium*.

[5] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781* (2013).

[6] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*. 3111–3119.

[7] Jeffrey Pennington, Richard Socher, and Christopher D Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*. 1532–1543.

[8] Ofir Press and Lior Wolf. 2016. Using the output embedding to improve language models. *arXiv preprint arXiv:1608.05859* (2016).

[9] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners. *OpenAI Blog* 1, 8 (2019), 9.

[10] Sebastian Ruder, Matthew E Peters, Swabha Swayamdipta, and Thomas Wolf. 2019. Transfer learning in natural language processing. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Tutorials*. 15–18.

[11] David E Rumelhart, Geoffrey E Hinton, and Ronald J Williams. 1986. Learning representations by back-propagating errors. *nature* 323, 6088 (1986), 533–536.