HinReddit: Understanding Hate and Conflict in Reddit Communities

Yu-Chun Chen yuc330@ucsd.edu
Chengyu Cheng chc401@ucsd.edu
Yanyu Tao yat017@ucsd.edu
Shuibenyang Yuan shy166@ucsd.edu
Halıcıoğlu Data Science Institute, University of California San Diego

ABSTRACT

In many Online Social Networks, negative behaviors such as hateful comments occur frequently and have greatly affected user experience as well as online communication quality. In our project, we investigate contents from Reddit and aim at classifying hateful post from regular posts. Instead of using NLP techniques, we use graph embedding methods. Specifically, we construct a heterogeneous information network (HIN) that captures several relationships among users and posts. We believe the HIN model is better than the commonly used NLP methods because, other than text-centric posts, it also enables us to predict memes or other image-centric posts. HinReddit presents methodologies over graph techniques of Node2vec, Metapath2vec, and DGI. One finding of our study is that users' posting and replying habits are necessary features, but more user-to-user relations will be needed to further fulfill HinReddit sufficiency. For comparison purpose, we also perform controversial/hateful subreddit identification based on the ground truth labels given by Reddit, and another important finding is that hate speech is more usefully modeled by communities than a single post.

I. INTRODUCTION

As countless social platforms are developed and become accessible nowadays, more and more people get used to posting opinions on various topics online. Even though people hold their rights to give free speech, some people consistently abuse their rights in delivering hateful speech. In order to prevent that from happening, our project plan to investigate contents from Reddit, which is a popular social network that focuses on aggregating American social news, rating web content and website discussion, that carries rich potential information of contents and their authors [1].

Our goal is to classify hateful posts from the normal ones. Being able to identify hateful posts not only enables platforms to improve user experiences, but also helps to maintain a positive online environment. In the beginning of this paper, we would like to stress that the boundary of 'hate' is vague and there is no correct nor consolidated definition of 'hatefulness,' our classification of hateful posts depends only on a unified definition within our team, which we divide into categories of severe toxic, toxic, threat, insult, and identity hate [2]. We all agree that other people's recognition of 'hate' may be but not limited to these four categories, and our labeling method allows full freedom of other definition of 'hatefulness.'

Our hypothesis is that hateful speech tends to appear in groups, and we believe that people who interact with a certain hateful post tend to participate in a discussion under other hateful posts and that's also why we will be using graph embedding methods. Specifically, we will create a heterogeneous information network to capture the relationships among Reddit posts, which is then used as our features.

1.1 Dataset

Our project includes following datasets:

Main dataset used for project analysis

This is a dataset we obtain from Reddit through API. We use the API called PushShift to obtain Reddit post information [3], including post text, title, and user IDs who reply to either the post itself or any of the reply below the post and the comments that it provided. We use PushShift because it offers a specific API to obtain the flattened list of repliers' IDs and takes considerably less time than doing the same with PRAW [4]. After a brief EDA on the most popular 124 subreddits, we select 47 subreddits in which 37 are quarantined and 10 are normal. A subreddit is quarantined if Reddit decides its content is too offensive for average reddit users, and thus we expected to obtain more hateful posts from these subreddits.

Our data represents a population of posts that is different from the actual population as we focus on quarantined subreddits, and thus our represented population would be more 'hateful'. However, it suits our purpose as it is obtained from the actual platform and contains more recent posts, and thus we obtain real-world and up-to-date perspective when training. Moreover, as we include more quarantined subreddits, we obtain more hateful posts and offset some problems that imbalanced data brings.

Our raw data includes three kinds of files: 1) csv files that contain the basic information of each post, 2) json files that

contain the post ID along with all of the comment ID belongs to it, and 3) csv files that contain the information of each comment.

Kaggle Toxic Comment Classification Dataset used for data labeling

This is a dataset provided on Kaggle [2], including information of hundreds of thousands of Wikipedia comments along with multiple negative labels. We mainly use this dataset to train a multi-layered NLP classifier model through Keras to label our reddit post data before we use it for HIN learning.

1.2 Related Work

HIN Based Problem

Detecting hateful posts on Reddit is similar to our domain problem of detecting Android malware both conceptually and technically [5]. Despite using different platforms, these two case studies both aim at identifying the malicious units from the benign units, and the goals are to produce a healthier and more positive environment to users. As we did in our replication using graph embedding techniques, here in our study, we will also pay attention to the connections as well as the communities of our object and construct heterogeneous information network (HIN) upon those connections that enables further training and classifications.

Specifically, in our HIN graph, we will have Reddit post nodes equivalent to App nodes in the replication project and user-interaction nodes equivalent to API nodes in the replication. While Hindroid investigates more of the relationships among API calls, for instance, having three out of four matrices developing different interactions of APIs, and thus focuses less on relationships among Apps themselves, we plan to add to our HIN the relationship among Reddit post nodes themselves to further diversify our network graph.

Social Network Based Problem

Studies regarding the detection of hateful speech, content, and user in Online Social Networks have been manifold. In the report Characterizing and Detecting Hateful Users on Twitter [6], the authors present an approach to characterize and detect hate on Twitter at a user-level granularity. Their methodology consists of obtaining a generic sample of Twitter's retweet graph, finding potential hateful users who employed words in a lexicon of hate-related words and running a diffusion process to sample more hateful users who are closely related in the neighborhood to those potential ones. However, there are still limitations to their approach. Their characterization has behavioral considerations of users only on Twitter, which lacks generality to be applied to other Online Social Networks platforms. Also, with ethical concerns, instead of labeling hate on a user-level, we believe that detecting hate on a content-level will be more impartial.

II. METHODOLOGY

2.1 Labeling

Since the original data obtained from Reddit is not labeled, we will be using a RNN and bidirectional layers, through python library Keras, as well as pre-trained word representation vectors from GloVe [7], to label the Reddit posts before we use it for our project main analysis.

By following a tutorial of using Keras and the pretrained word vectors [8], we will train a multi-label bidirectional layer RNN model with Kaggle labeled dataset of Wikipedia comments detailed in Datasets. We will save this model in directory interim. This multi-label model then can be used to calculate each Reddit post or comment a score between 0 to 1 for each of the label toxic, severe_toxic, threat, insult, identity_hate.

The labeling process for a single post is explained as follows: We first obtain scores for the post itself if it has textual content. Then, we also obtain scores for all of its comments to compute an average of all five labels. We then compute total scores for the post by adding scores of post content and its comments. We then compute the max of all five total scores, and if the max value is greater than the threshold, we classify the post as 'hateful'. In our project, we set the threshold to 0.5. If the post is removed it will be labeled as deleted and the NA post will also be labeled as NA.

In this way, we can also label those posts which are missing textual content by making use of its comment data. Moreover, with this labeling process, we are defining hateful posts so that they not only include those that demonstrate hatefulness in its content, but also those that stir up negative discussions in comments and replies.

2.2 Graph Extraction

We have developed two kinds of network embeddings over the course in the hope to try out if different embeddings can have an impact on model results. Both kinds of embeddings include two types of nodes: post nodes and user nodes.

Graph 1

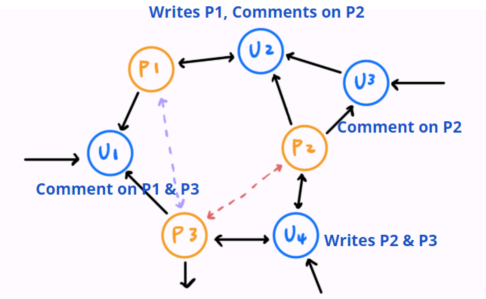

Figure 1: First Graph Embedding

Three relationships are represented in Graph 1 as shown in *Figure 1* above: authorship, involvement, and reply. The matrices are A, P, and U respectively.

A matrix represents authorship: there're arrows pointing to post nodes from a user node for posts that are written by the user. The P matrix represents involvement: there're arrows pointing to user nodes from a post node for users who either writes or comment below the post. Finally, the U matrix represents reply: there're arrows pointing to user node A from another user node B if A has replied to B under any post. This can be represented further below:

A: $A_{ij} = 1$ if user i writes post j
P: $P_{ij} = 1$ if user j either writes or comments below post i
U: $U_{ij} = 1$ if user i is replied by user j

The advantage of this graph embedding is that posts nodes are very close if they are written by the same users. On the other hand, even if posts are written by different users, their nodes are still connected through involving users in common, and the closeness of the posts in our graph is based on the similarity between the group of users that interact with each of them.

Graph 2

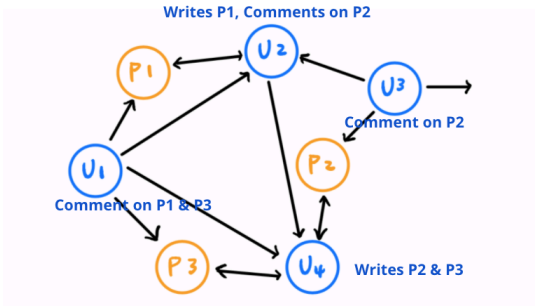

Figure 5: Second Graph Embedding

Our Graph 2 also embed similar relationships to Graph 1 but with a more intuitive flow. The matrices are also represented as A, P, and U. However, these relationships are defined slightly different from those in Graph 1.

The A matrix in Graph 2 represents the same authorship relationship but in an opposite direction: whenever a user writes the post, the post node will point to the user. The P matrix also represents involvement similarly but with an opposite direction: whenever author writes the post, the author will point to the post node, and whenever a user replies under a post, the user node will also point to the post node. The U matrix is the same as in Graph 1: author will point to user when the author is replied by the user, and user A will point to user B when user B is replied by user A. This can be represented further below:

A: $A_{ij} = 1$ if post i is written by user j
P: $P_{ij} = 1$ if user i either writes or comments below post j
U: $U_{ij} = 1$ if user i is replied by user j

The difference between these two graphs is that the second one adds weights to users of first level replies who directly comment below the main post by adding extra arrows. For this specific embedding, post nodes are still close if they are written by the same users. On the other hand, they are also close because they have many first-level repliers in common.

This graph embedding then produces the following network of posts. *Figure 4* presents post network classified by whether it is hateful, while *Figure 5* below presents post network classified by subreddits.

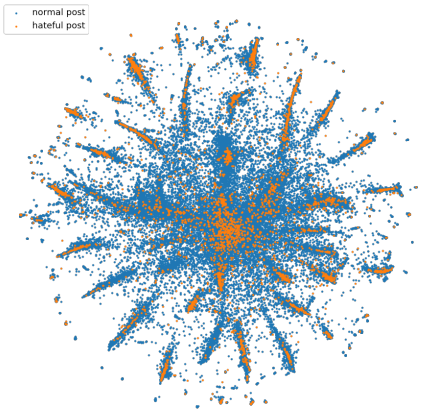

Figure 4: Normal/Hateful Post Network

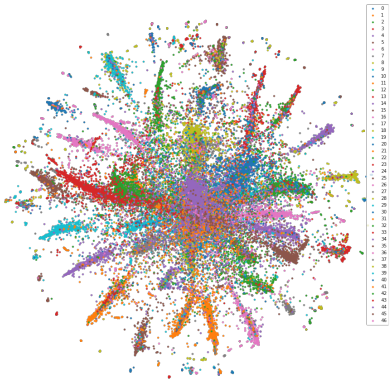

Figure 5: Post Network by Subreddits

2.3 Classification Modeling

Metrics

Since we are performing binary classification with imbalanced dataset, True Positive and True Negative play more crucial roles in our classification model. Because graph techniques will be significantly influenced by traditional balancing data technique like over-sampling and under-sampling, we will evaluate our model with following metrics: Recall and Precision, to catch more potential hateful posts. We also present AUC as an indication of how well our model is distinguishing between hateful and benign post.

Techniques

Hinreddit presents three methodologies over following graph techniques: Node2vec [9], Metapath2vec [10], and DGI [11]. We use these three semi-supervised learning techniques to get the representational learning on the graph. Typically, we use node2vec, metapath2ec, and deep graph infomax (DGI) since they are well-explained on features and popularly addressed in different papers on graph neural networks. We expect to use our embeddings to cluster the post nodes into two different communities: hateful posts and normal posts.

Node2vec and Metapath2vec contain only graph information, while DGI has the power to include other features about the nodes. Node2vec and metapath2vec are similar: both of them can automatically exploit the neighborhood structure through sampled paths on the graph by random walk. Metapath2vec demands more on memory and speed since it also catches information about different metapaths. DGI does not rely on random walk: it rather replies on maximizing mutual information between patch representations and corresponding high-level summaries of graphs, which is the post features in our case.

The details of each algorithm are explained below:

Node2vec: This is an algorithmic framework for learning continuous feature representations for nodes in networks. The algorithm is able to capture the relationships in the neighboring nodes by using the biased random walk. It can be adjusted by p and q values, which are chances to go forward and backward respectively.

Metapath2vec: This model applies random walks on the formalized metapath and used it to construct the heterogeneous neighborhood of a node. It is then put into a heterogeneous skip-gram model to perform node embeddings. Our chosen metapath is: ('post', 'commented by', 'user'), ('user', 'replied by', 'user'), ('user', 'wrote', 'post'), or PUA for Graph 1 and P^TU^TA for Graph 2.

Deep Graph Infomax: DGI is an unsupervised graph neural network that can learn high level node representations by applying mutual information maximization. It also enables us to include baseline features directly in the model.

III. EXPERIMENTAL RESULTS

3.1 Baseline Model Result

Estimator	Precision	Recall	AUC	ACC
Logistic Regression	0.1346	0.7355	0.7630	0.7883
RandomForest	0.1422	0.3900	0.6429	0.8744
GradientBoosting	0.6258	0.1163	0.5566	0.9596

Table 1: Baseline Model Result

3.2 HinReddit Results with Graph 1

Our first graph embedding produces the following network of posts. *Figure 2* and *3* are network graphs processed with Node2vec, while *Figure 4* and *5* are network graphs processed with Metapath2vec. *Figure 2/4* present post network classified by whether it is hateful, while *Figure 3/5* present post network classified by subreddits.

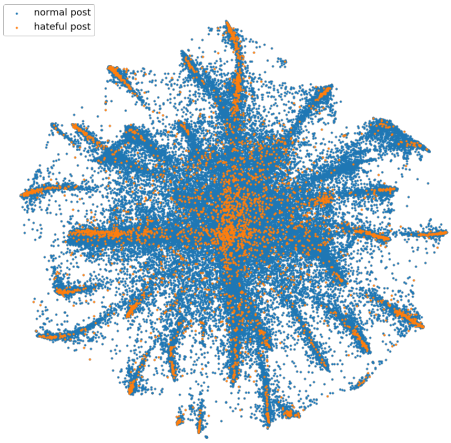

Figure 2: Normal/Hateful Post Network with Node2vec

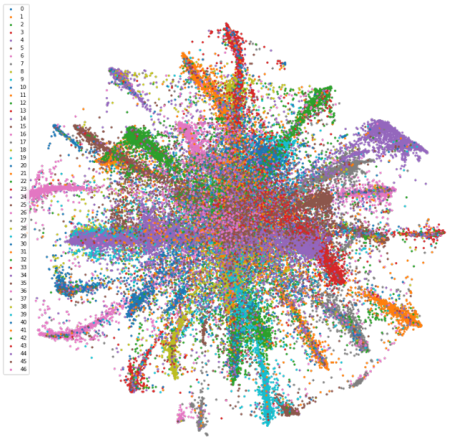

Figure 3: Post Network by Subreddits with Node2vec

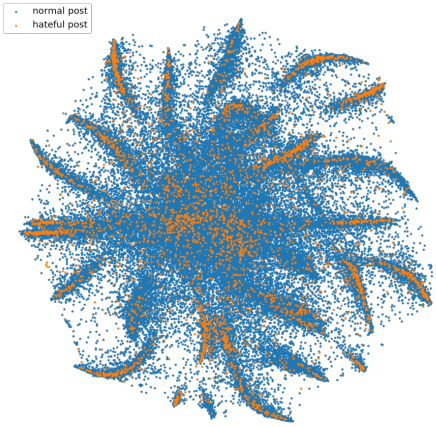

Figure 4: Normal/Hateful Post Network with Metapath2vec

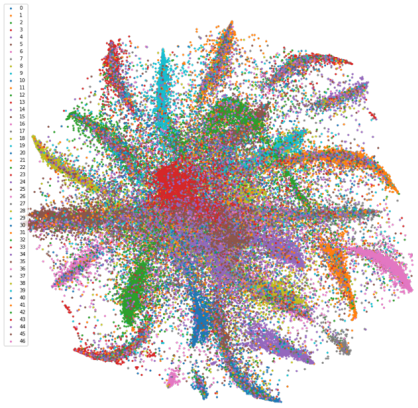

Figure 5: Post Network by Subreddits with Metapath2vec

The model results are then presented in table below.

Task	Algorithm	Precision	Recall	AUC	ACC
controversial community detection	Node2vec	0.9362	0.8225	0.8161	0.8196
controversial community detection	Metapath2vec	0.9079	0.7893	0.7587	0.7754
subreddit community detection	Node2vec	NA	NA	NA	0.8052
subreddit community detection	Metapath2vec	NA	NA	NA	0.7027
hateful post detection	Node2vec	0.0569	0.5436	0.5737	0.6013
hateful post detection	Metapath2vec	0.0559	0.5413	0.5694	0.5952
hateful post detection	DGI	0.1379	0.7172	0.7593	0.7979

Table 2: Model Results with First Embedding

3.3 HinReddit Results with Graph 2

Our second graph embedding produces the following network of posts. *Figure 6* and *7* are network graphs processed with Node2vec, while *Figure 8* and *9* are network graphs processed with Metapath2vec. *Figure 6*/*8* present post network classified by whether it is hateful, while *Figure 7*/*9* present post network classified by subreddits.

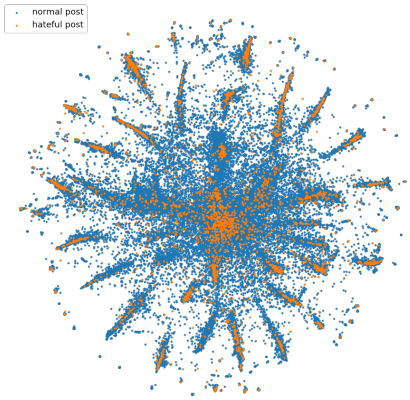

Figure 6: Normal/Hateful Post Network with Node2vec

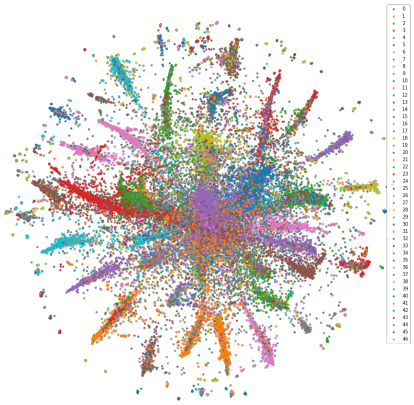

Figure 7: Post Network by Subreddits with Node2vec

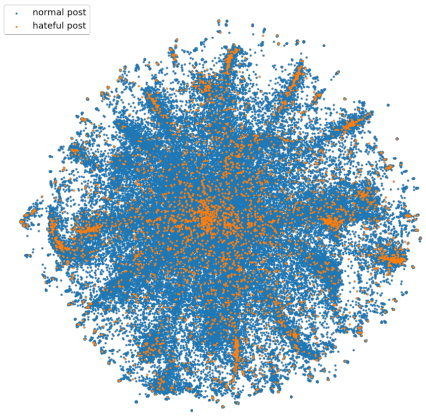

Figure 8: Normal/Hateful Post Network with Metapath2vec

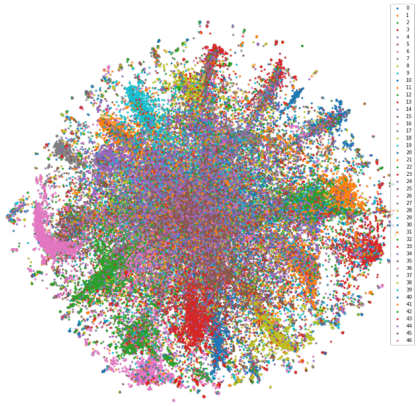

Figure 9: Post Network by Subreddits with Metapath2vec

The model results are then presented in table below.

Task	Algorithm	Precision	Recall	AUC	ACC
controversial community detection	Node2vec	0.9497	0.8772	0.8597	0.8692
controversial community detection	Metapath2vec	0.9052	0.7771	0.7503	0.7649
subreddit community detection	Node2vec	NA	NA	NA	0.8364
subreddit community detection	Metapath2vec	NA	NA	NA	0.6409
hateful post detection	Node2vec	0.0644	0.6250	0.6126	0.6012
hateful post detection	Metapath2vec	0.0535	0.5298	0.5586	0.5849
hateful post detection	DGI	0.0772	0.7172	0.6561	0.6531

Table 3: Model Results with Second Embedding

3.4 HinReddit Results with Baseline Features

Estimator	Algorithm	Precision	Recall	AUC	ACC
Logistic Regression	Node2vec	0.1306	0.7275	0.7565	0.7830
Logistic Regression	Metapath2vec	0.1284	0.7206	0.7520	0.7808
Logistic Regression	DGI	0.1379	0.7172	0.7593	0.7979
RandomForest	Node2vec	0.6143	0.0490	0.5238	0.9584
RandomForest	Metapath2vec	0.6032	0.0433	0.5210	0.9582
RandomForest	DGI	0.2939	0.1505	0.5673	0.9487
GradientBoosting	Node2vec	0.6395	0.1072	0.5523	0.9596
GradientBoosting	Metapath2vec	0.6081	0.1026	0.5498	0.9591
GradientBoosting	DGI	0.5611	0.1151	0.5556	0.9587

Table 4: Results of First Embedding and Baseline Features

Estimator	Algorithm	Precision	Recall	AUC	ACC
Logistic Regression	Node2vec	0.1313	0.7320	0.7588	0.7833
Logistic Regression	Metapath2vec	0.1314	0.7355	0.7601	0.7827
Logistic Regression	DGI	0.1303	0.7628	0.7686	0.7740
RandomForest	Node2vec	0.6125	0.0559	0.5272	0.9585
RandomForest	Metapath2vec	0.5692	0.0422	0.5204	0.9580
RandomForest	DGI	0.2146	0.1984	0.5831	0.9352
GradientBoosting	Node2vec	0.5935	0.1049	0.5509	0.9590
GradientBoosting	Metapath2vec	0.6174	0.1049	0.5510	0.9592
GradientBoosting	DGI	0.6062	0.1334	0.5648	0.9596

Table 5: Results of Second Embedding and Baseline Features

IV. DISCUSSION

According to the graphs of post networks, we see no clear clusters in network of normal/hateful posts with whichever graph embedding and algorithm we use. On the other hand, clusters have clearer boundaries in graphs of network of posts by subreddits.

As seen above in the tables, we have obtained fairly low precisions and recalls with our current user-post embeddings and models when classifying on a post level. The results can be understood together with our Exploratory Data Analysis in section *A.2*. The data has shown that only 7% of users have ever engaged in hateful posts, and among them almost half of users have themselves write posts/comments that are labeled as hateful. Moreover, for users who have engaged in hateful posts, only around 28% of their posted speeches are labeled as hateful. These numbers suggest users have a small chance of creating their own hateful posts/comments although they have engaged in any of the hateful posts. Furthermore, even if they have created any, it is not a consistent behavior. This then demonstrates that our initial hypothesis might not be accurate as mere relationships of users' reply behavior and authorship cannot provide much useful information in identifying hateful posts, which is then confirmed by our model results.

Due to the fact that our graph representation only embeds authorship and reply behavior, and as Node2vec and Deep Graph Infomax both are greedy in the training process, the

models cannot clearly distinguish between hateful and benign posts, which is shown by the AUC values that are only slightly higher than, or even lower than, 0.5 and our baseline models that make use of post-related features.

Nevertheless, our graph embeddings perform well on community detection and distinguish between normal/quarantined subreddits better. This is reasonable because 85% of users are only active in one subreddit. If we have constructed our graphs correctly, they would capture community network well even with only posting/replying behavior.

Our result has limited applicability depending on data sources. Possible data sources include other online social platforms such as Twitter, Facebook, LinkedIn, and Instagram. However, platforms have similar overall structure but differ in detailed construction and user habits. Our results mostly apply to platforms that have similar nested replying system such as Twitter, and possibly community categorization similar to the subreddit feature in Reddit.

V. FUTURE WORK

First possible improvement can be done in the labeling process. To label our Reddit data, we can neither define 'hate' using ground truth nor find a definition with clear boundary, thus we can only train an NLP classifier with labeled Wikipedia comments as well as pretrained Wikipedia vocabularies. The labels of these Wikipedia comments are also vague and clear definition are not provided. It would be better if we can obtain labeled social platform data, such as labeled tweets to be trained with pretrained Twitter vocabularies provided by Glove.

Another possible improvement, which we originally would like to implement, is to include more user-to-user relations in our graph representations. Some examples include subreddit subscription lists and friend connection. We currently have a hard time including these relationships because user information features are still in development in the API we use, pushShift. Although Reddit's own API, PRAW, offers related features, it unfortunately employs a different system of user ID from what PushShift uses, and we are unable to connect them to obtain user information. This can be done as soon as PushShift successfully develops user information features.

Finally, we can also improve our algorithm to make use of the timeline data we obtain along with posts and comments. By adding a time feature, we can construct sequence nodes and feed into Recurrent Neural Network models, such as Long Short-Term Memory networks.

VI. ACKNOWLEDGEMENT

We sincerely thank Prof. Aaron Fraenkel (University of California, San Diego) and our TA Shivam Lakhotia (University of California, San Diego) who provided expertise that greatly helped us with this Data Science Senior Capstone throughout Winter 2020 and Spring 2020.

VII. REFERENCE

[1] Reddit https://www.reddit.com
[2] Toxic Comment Classification Challenge https://www.kaggle.com/c/jigsaw-toxic-comment-classification-challenge/data
[3] PushShift Reddit API Documentation https://github.com/pushshift/api
[4] PRAW: The Python Reddit API Wrapper https://praw.readthedocs.io/en/latest/
[5] Hou, S., Ye, Y., Song, Y. and Abdulhayoglu, M., 2017. Hindroid: An Intelligent Android Malware Detection System Based On Structured Heterogeneous Information Network. [online] Available at: <https://www.cse.ust.hk/~yqsong/papers/2017-KDD-HINDROID.pdf> [Accessed February 2020].
[6] Ribeiro, M., Calais, P., Santos, Y., Almeida, V. and Meira Jr, W., 2018. Characterizing And Detecting Hateful Users On Twitter. [online] arXiv.org. Available at: <https://arxiv.org/abs/1803.08977>
[7] GloVe: Global Vectors for Word Representation https://nlp.stanford.edu/projects/glove/
[8] Multi-Label text classification in TensorFlow Keras https://androidkt.com/multi-label-text-classification-in-tensorflow-keras/
[9] Node2vec https://snap.stanford.edu/node2vec/
[10] Metapath2vec https://ericdongyx.github.io/papers/KDD17-dong-chawla-swami-metapath2vec.pdf
[11] Deep Graph Infomax https://arxiv.org/abs/1809.10341
[12] List of Quarantined Subreddits https://www.reddit.com/r/GoldTesting/comments/3fxs3q/list_of_quarantined_subreddits/

APPENDIX

A.1 Label Statistics

As you may know, Reddit has already banned lots of subreddit that contained explicit or controversial materials. Thus in order to discover more hateful speech, we researched online and find out a list [12] contained both banned and quarantined subreddits. Quarantined subreddits are subs that host no advertisement, and Reddit doesn't generate any revenue off toxic content. People can still access those subs, but there will be a prompt warns telling people about the content on the sub. We have selected around 37 quarantined subreddit along with 10 normal subreddits.

By using the data ingestion pipeline, we have successfully extracted 5,000 posts from each of the 47 subreddits which is 235,000 posts in total. Some basic statistics about the comments are shown in the tables below.

Proportion of Each Label

We then look at the labels at a higher level without grouping them into different subreddits. The figure below shows the distribution of the labels among posts.

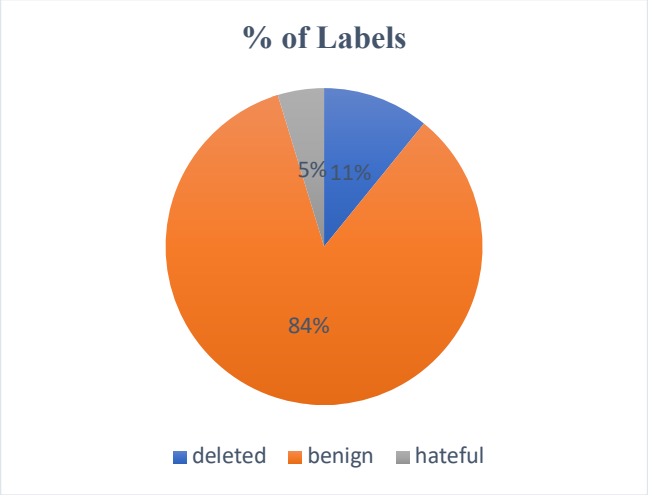

Figure A.1: Label Distribution

Number of Comments

Another feature could be the number of comments under each post. The average length of comment for posts labeled hateful is relatively smaller than that for posts labeled as benign.

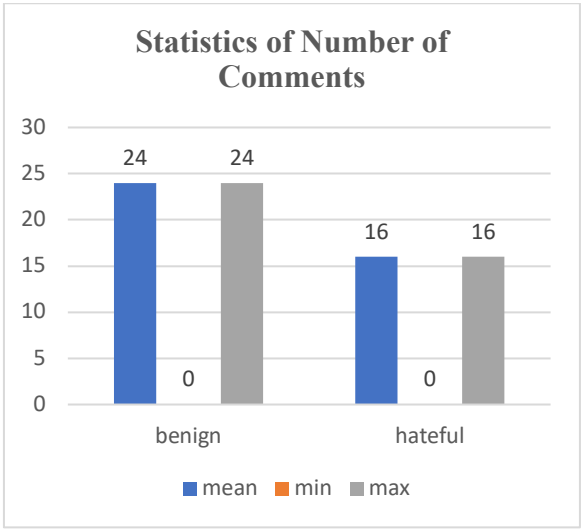

Figure A.2: Number of Comments

Length of Text Content

Dig deeper into the content of the posts for different labeling groups, we investigate on the length of the content. From the table below, it shows that even though the min and max of the length of content in each group is around the same, the average length of content for posts that are labeled hateful is more than double of the average length of content for posts that are labeled benign. Thus, we can add this as one of our features.

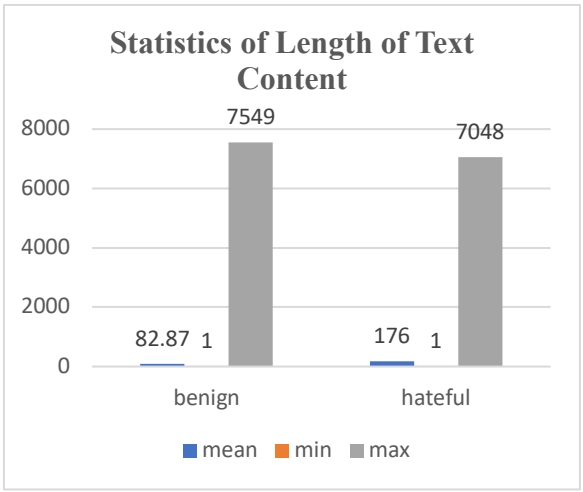

Figure A.3: Length of Text

Upvote Score

Moreover, we also find difference in score for the two groups, the mean score of benign posts are generally higher than those of hateful posts.

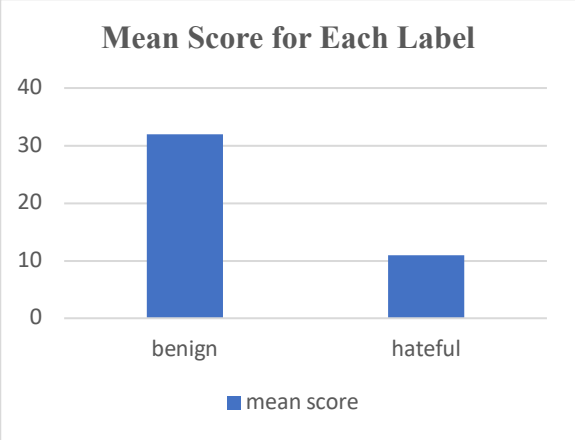

Figure A.4: Mean Upvote Score

Vocabulary

Moreover, in order to evaluate the quality of the label, we have also done some textual analysis. We find out the top 30 words in posts after removing stop words for each of the groups. However, we have also removed about 20 words that appeared in both groups. Those should be the common words that appeared in the conversation and thus is not helpful as a feature for our classification.

malign_word	count	benign_word	count
fuck	5,835	amp	13,155
nigger	4,078	work	12,508
fucking	3,233	feel	12,388
shit	2,907	right	12,208
place	2,713	gt	11,256
sex	2,200	things	11,244
started	1,840	new	11,002
ass	1,800	need	10,629

Vocabulary

Table A.1: Sensitive Vocabulary

A.2 Interaction b/w Posts & Users

We have 483,173 unique users in our data. 7% of users in our data have been involved with hateful posts. Among them, 44.26% of users have themselves create posts/comments labeled as hateful.

Percentage of users only post once	Proportion of users post only in 1 subreddit
44.95%	85.24%

Interaction b/w Posts & Users

Table A.2: User Posting Habits

We can observe that nearly half of the users post only once and are not active authors on Reddit. Most of them are only involved within one subreddit, thus their behavioral movements are representative of that subreddit.

In addition to general user, we also investigate hateful post users' behaviors specifically.

Proportion of users who only post once among all users engaged in hateful post	Proportion of users who post only in 1 subreddit among all users engaged in hateful post
21.01%	70.03%

Post-User Engagement

Table A.3: User Posting Habits for those who have Engaged in at least One Hateful Post

We can observe that users who engage in hateful post are more active authors compare to general users.

Some users engage in both benign and hateful posts, and we found that among users who have engaged in hateful posts, 14% of authored posts and comments are classified as hateful.