ANALYSIS OF TOXICITY IN SOCIAL MEDIA

Major Project-II End-Semester Report

Submitted in partial fulfillment of the requirements for the degree of

BACHELOR OF TECHNOLOGY

in

COMPUTER ENGINEERING

by

ANEESH AITHAL (15CO107)
ADITHYA S KAMATH (15CO103)
HARSHITH KUMAR (15CO120)

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

NATIONAL INSTITUTE OF TECHNOLOGY KARNATAKA,

SURATHKAL, MANGALORE - 575025

August, 2019

DECLARATION

We hereby declare that the Major Project - II End-Semester Report entitled **ANALYSIS OF TOXICITY IN SOCIAL MEDIA** which is being submitted to the **National Institute of Technology Karnataka, Surathkal** in partial fulfillment of the requirements for the award of the Degree of **BACHELOR OF TECHNOLOGY** in **Computer Engineering** is a *bonafide report of the work carried out by us.* The material contained in this report has not been submitted to any University or Institution for the award of any degree.

Aneesh Aithal (15CO107)
Department of Computer Science and Engineering

Adithya S Kamath (15CO103)
Department of Computer Science and Engineering

Harshith Kumar (15CO120)
Department of Computer Science and Engineering

Place: NITK, Surathkal.
Date: 25.04.2019

CERTIFICATE

This is to *certify* that the Major Project - II End-Semester Report entitled **ANALYSIS OF TOXICITY IN SOCIAL MEDIA** submitted by:

(1) **ANEESH AITHAL** (Register Number: 15CO107),
(2) **ADITHYA S KAMATH** (Register Number: 15CO103),
(3) **HARSHITH KUMAR** (Register Number: 15CO120)

as the record of the work carried out by them, is *accepted as the Major Project-II End-Sem Report submission* in partial fulfilment of the requirements for the award of degree of **Bachelor of Technology**.

Dr. Annappa
Guide

Chairman - DUGC

Acknowledgment

We would like to thank Professor Annappa for his support and encouragement for this project and his valuable inputs. We would also like to extend our gratitude to the Computer Science and Engineering department of National Institute of Technology, Karnataka for providing us with the opportunity to work on this project.

Place: Surathkal
Date: 25.04.2019

Aneesh Aithal
Adithya S Kamath
Harshith Kumar

Abstract

Social Media has facilitated a new level of interactions that have led to a lot of interesting discussions. However, discussing things one cares about online, can be challenging. The menace of abuse, bullying and harassment online means that many people become hesitant in expressing themselves and often give up on seeking different opinions. Platforms struggle to effectively facilitate conversations, leading many communities to limit or completely shut down user comments or into embracing the echo chamber. Instead of resorting to blanket censorship methods, we believe social media platforms can effectively deal with this problem by analysing levels of toxic behaviour in their various sub-communities and then dealing with these problem sources individually. We focus our analysis on Reddit, a popular social news aggregation, web content rating and discussion website.

We conducted several experiments on Reddit comment data using a model we created based on a Wikipedia dataset. From these experiments, we were able to create a list of the most toxic subreddits, calculated the effect of moderation on toxic behaviour and the correlation between comment scores and toxicity. We also tested a model that uses pseudo-labelling to learn from new Reddit comments.

Keywords: Toxic behaviour, Reddit, moderators on social media, bidirectional gated recurrent unit, fastText

Contents

Chapter 1

Introduction

The term ‘toxicity’, when used in the context of conversations, refers to the degree of offensiveness that is implied on the reader. Most of the online platforms that encourage conversations contain a small community of users who tend to have a more offensive behavior. These communities cause unnecessary conflicts, which invoke the action of ‘moderators’ who are given the task of filtering out the offensive conversation. In worse cases, such communities could be too large for the moderators to deal with, which would ultimately lead to shut down of the conversations on those platforms.

An effective solution to the above problem would need to address two issues. Firstly, the method of detection of the offensiveness and the further action has to be automated so that it scales with the size of community. Secondly, the subjectivity involved in rating a given conversation under different levels of toxicity makes it a hard task to generalize the solution.

Over and above all the above issues lies the diversity present in the online communities. One can divide the online communities based on multiple factors. For example, one can divide it based on the social media platform, based on the age group, based on interest groups, etc. There are various social media platforms that collectively form a major part of the Web. One of the platforms that has been studied extensively by the NLP research community is Reddit.

1.1 Reddit

Reddit is a social news aggregation, web content rating, and discussion website. Registered members submit content to the site such as links, text posts, and images, which are then voted up or down by other members. As of 2018 there were about 330 million Reddit users, commonly known as "redditors". Posts are organized by subject into user-created boards called "subreddits", which cover a variety of topics including news, science, movies, video games, music, books, fitness, food, and image-sharing.

As a network of communities, Reddit's core content consists of posts from its users.Users can comment on others' posts to continue the conversation.A key feature to Reddit is that users can cast positive or negative votes, called upvotes and downvotes, for each post and comment on the site. The number of upvotes or downvotes determines the posts' visibility on the site, so the most popular content is displayed to the most people. Users can also earn "karma" for their posts and comments, which reflects the user's standing within the community and their contributions to Reddit. Submissions with more up-votes appear towards the top of their subreddit and, if they receive enough votes, ultimately on the site's front page. By default for those users, the front page will display the subreddit r/popular, featuring top-ranked posts across all of Reddit, excluding not-safe-for-work communities and others that are most commonly filtered out by users (even if they are safe for work). The subreddit r/all does not filter topics. Registered users who subscribe to subreddits see the top content from the subreddits to which they subscribe on their personal front pages.

Front-page rank—for both the general front page and for individual subreddits—is determined by a combination of factors, including the age of the submission, positive ("upvoted") to negative ("downvoted") feedback ratio, and the total vote-count.

Despite strict rules prohibiting harassment, Reddit's administrators spend considerable resources on moderating the site. This would sometimes require a moderator to go through thousands of comments per post to remove hate speech.

We'll discuss the basic structure and the terminology associated with Reddit below:

1.1.1 Subreddit

Reddit is divided into thousands of completely user-created and moderated communities called subreddits. These are all topic specific forums for content posted and discussed by the users of the subreddit. Science, Politics, WorldNews etc. are some of the popular subreddits with millions of subscribed users each. Subreddits are referred to as "r/subredditname" commonly. When a new user joins Reddit, the account is automatically subscribed to a bunch of popular subreddits called default subreddits. Each subreddit has its own set of rules that are enforced by moderators of the subreddit. Users make their submissions to the subreddit of their choice. Most subreddits are public, and very few private subreddits exist. Often, when a subreddit violates Reddit rules or are responsible for toxic behaviour, the entire subreddits are usually "quarantined" or outright banned. Quarantined subreddits do not make it the r/popular or r/all subreddits and are harder to subscribe and view.

1.1.2 Posts

Registered users of the website can contribute to subreddits by creating self-posts that have a 10,000 character limit or by linking external sources as content. Common external content are usually images, GIFs, videos and links. Posts are viewed by moderators of that subreddit to ensure that they follow the subbreddit and Reddit's policies, and can choose to remove the posts if they feel that the posts are in violation of the abvoe mentioned rules. Each post has its own comment section which is moderated as well. The comment section is the core part of Reddit's features and is the location of most interaction on the website.

1.1.3 Comments

Like posts, any registered user can comment. Each post has its own comment section. The entire comment section can be visualized as a forest of trees. Each tree indicates a comment thread. A new comment can be in response to the post itself or in response to a comment at the root level or in response to any of the sub comments. In other words, every comment thread resembles a tree. Moderators can review comments for rule violations, similar to that of posts, can choose to remove them if they wish to.

Comments and posts can be reported by normal users, therefore sending the reported comments and posts to the moderators to review.

1.1.4 Moderators

Every subreddit on the website has a set of users that are appointed as the moderators by the owner/creator(s) of the subreddit. Their role is to ensure that the community rules and guidelines set by that subreddit are followed by the users. They can also set further community guidelines for users to follow. They can also "sticky" comments and posts, pushing them to top of the comment section and subreddit view respectively. A moderator has the ability to remove comments and posts. Usually moderators do so if they find a comment or post to be in violations of the guidelines.

1.1.5 Upvotes and Downvotes

Every post and every comment on Reddit, unless archived by the moderators or if it is older than 6 months, can be downvoted or upvoted once by any registered user. An upvote adds 1 to the total count score and a downvote results in a -1. When a subreddit is sorted by "Hot" or "Best", which is the default sorting order, all posts and comments are sorted based on a ranking function that takes the post score and time into account. Newer posts are placed higher in this order than older posts with the same post score.

1.1.6 Karma

Karma is the sum of a users total comment score and total post score. The effect of downvotes on reducing Karma is limited to prevent brigading by users to reduce Karma of a single user. Although it is considered to be a gimmick by some, Karma can still affect a user's ability to post and comment on the platform as well as the user reputation. Low Karma might result in a user's post being removed automatically or by a moderator on certain subreddits as some of them tend to have a minimum karma threshold. Reddit introduced karma as a method to combat the problem of spamming and bot activity. It can be viewed as a measure of a user's reputation and his/her contribution to the Reddit community. A higher amount of karma translates to greater freedom on the platform. Karma is calculated using a complicated algorithm that's not open source and frequently updated. Karma can be gained or lost from both comments and posts, where an upvote will gain karma and a downvote will result in the user losing karma.

1.1.7 The Front Page

The front page of Reddit is a collection of the 'hottest' content across reddit. If a user is signed in then the front page acts like a personalized feed and shows the 'hottest' content from all the various subreddits that the user is subscribed to. The content on the front page is sorted by an algorithm called 'hot'. Submission time greatly impacts the 'hotness' of a post and the algorithm tends to rank new stories higher than the older ones. The upvote count of the post is also a significant factor.

1.2 Research Focus and Potential

Our research focuses on several aspects. First, we try to model the toxicity of Reddit comments. At the end of this phase, we aim to accurately classify a given comment under various levels of toxicity. This model can be used by moderators for automating their task. The list of toxic subreddits can be used by the Reddit team to analyze and understand toxic subreddits. Further, since our model gives scores for multiple aspects of toxicity, a moderator can use this model in a way that fits their use-case. For example, some moderators may want to keep moderately toxic comments but only remove severely toxic ones. By setting an appropriate threshold for the scores of these two classes, moderators can achieve their ends. The second aspect of our research involves a broader analysis of how toxicity affect communities as a whole. We perform experiments using our base model to get a high-level picture on toxic behaviour and moderation within Reddit communities. This aspect is useful to researchers to understand the effect of moderators on toxic behaviour. Based on conclusions, it leads to questions on why moderation is ineffective in curbing toxic behaviour or even possibly encouraging it. Research can also be focused on understanding the effect of the minority consisting of the most toxic users on the overall website. Further, this analysis could be applied on a user level to understand the different groups of toxic users on the website. This information could then be used to create a toxicity network of all subreddits, linking subreddits with one type of toxic behaviour near each other.

1.3 Issues and Challenges

- Subjectivity in evaluating levels of toxicity. What one person may find toxic may not be found toxic by someone else. While the dataset is crowdsourced to ensure that the subjectivity debate is handled, there is no fixed metric to determine if a comment is toxic.

- Scalability to handle large toxic communities. A model should be able to rapidly evaluate comments in order to keep up with the stream of new comments.

- Various forms of toxicity (eg. emotes can change the level of toxicity).

- Diverse communities with various topics of discussion.

- Usage of language different from the one the original model is trained on.

- Presence of erroneous words and improper grammar (inconsistency). If handled incorrectly or not handled at all, a model can learn improperly from data.

- Varying comment parameters like length, words, part of speech, tense, etc.

- Restrictions imposed on automated collection of comments from social media websites

- Removed and deleted comments. Such comments could have been toxic but are unavailable due to the moderator or the user deleting them.

- Smaller communities on Reddit have sparse activity on them, making collecting data about them very hard.

- Limited labeled dataset to train the model. The only available dataset that is labeled for toxicity is the Wikipedia dataset available on Kaggle.

Chapter 2

Literature Survey

2.1 Related Work

2.1.1 Recurrent Neural Networks

RNNs work well for NLP tasks by capturing sequential dependencies and patterns. But most RNNs fall prey to the classical problem of vanishing gradients. Basically, RNNs fail to learn long-term dependencies because the gradients fall off exponentially and the contributions of the inner layers die out. Many variations of vanilla RNN were introduced. One such variation was introduced by Hochreiter and Schmidhuber [1997], called LSTM (long short-term memory). LSTM architecture consists of memory cells that store long term dependencies and also, a set of gates that control the flow of information to and from these memory cells. This model successfully bypasses the vanishing gradient problem. But the performance of LSTM becomes slightly worse compared to its competitors when it comes to smaller labelled datasets, which is why we decided to not use this model.

2.1.2 Analysis of Toxic behaviour

Analysis of toxic human behavior on social media is not new in the field of research. Paper by Xu and Zhu [2010] covers approach to detecting and filtering hate speech on social media. It uses automatic sentence-level filtering approach that is able to semantically remove the offensive language by utilizing the grammatical relations among words. The accuracy of the automatic approach is shown to match that of manual filtering, thus making it practically viable. The overhead of applying the approach in real-life filtering is also shown to be reasonable. Since most of the grammatical relations among words is captured by word embeddings, we referred this paper only to get a basic idea of hate speech detection.

Coming back to hate speech detection, paper by Djuric et al. [2015] make use of word embeddings to learn distributed low-dimensional representations of comments. The paper aims to addresses issues of high-dimensionality and sparsity that impact the current state-of-the-art, resulting in highly efficient and effective hate speech detectors. This is beneficial to our analysis as we use embedding layer followed by classifier to detect toxicity.

Toxicity not only plagues social media discussions but also is a 'poison' in other domains. Paper by Kwak et al. [2015] explores cyberbullying and other toxic behavior in team competition online games. The work also aims to serve as basis for building systems to detect, prevent and counter-act toxic behaviour. Most of the negative aspects of toxicity is explained and the in-depth analysis of the study itself can help our analysis with respect to social media.

2.1.3 Comment Relevancy and Sentiment Analysis

Relevancy of comments was covered by Siersdorfer et al. [2010]. The paper does an in-depth study of commenting and comment rating behavior on a sample of more than 6 million comments on 67,000 YouTube videos. Dependencies are analyzed between comments, views, comment ratings and topic categories. Further studies involve the influence of sentiment expressed in comments on the ratings for these comments using the SentiWordNet thesaurus. The study shows that user rating on these comments give a good starting ground to filter unrated comments. We believe that study of labelled comments would, thus, help sort out irrelevant comments thus leading to toxic comments.

Sentiment analysis has been used on social media data to for several different purposes. Warner and Hirschberg [2012] used sentiment analysis to detect hate speech. It is also shown that hate speech is usually caused by a small set of high frequency stereotypical words and with proper word sense disambiguation, it is possible to detect hate speech among communities. While sentiment analysis stands to be one of the most used techniques in hate speech detection and similar NLP tasks, we believe that it is redundant when coupled with more advanced models that we decided to use.

2.1.4 Dissection and Analysis of the Reddit platform

Paper by Weninger et al. [2013] is first and one of the very few papers that conduct an extensive research on Reddit. The paper specifically aims to study underlying structure of Reddit and observe how the discussion threads can be used in various general tasks like search engine optimization.

Paper by Buntain and Golbeck [2014] again goes back to studying Reddit. This paper mainly focuses on user interactions on Reddit. It studies the posting behaviour of users and determines the 'answer-person' role on Reddit through automated means. We derive some of the useful insights from this paper that may be useful in our analysis. One of the insights is that the users rarely exhibit significant participation in more than one communities.

Reddit has dealt with harassment since its inception in 2005. There have been several controversies concerning bullying, toxic behaviour and harassment. In 2015, enforcement of the anti-harassment policy led to the banning of r/fatpeoplehate and several other subreddits dedicated to harassing minorities. A study Chandrasekharan et al. [2017] found that banning on these subreddits lead to an overall decrease in hate speech throughout the site in the coming months. They also found that existing accounts who were active in the previously unbanned subreddits decreased their hate speech usage by at least 80%.

2.1.5 Pseudo-Labeling

Lack of sufficiently labelled training data can often be hindrance while building accurate machine learning models. This is especially the case in NLP tasks where most of the labels are subjective and the ground truth data is not enough to train deep architectures with low generalization error. Different approaches exist to work well on smaller labelled datasets. One of the approaches involve making use of both labelled and unlabelled data to make model generalization better. This approach of semi-supervised learning, as explained by Lee [2013], is called Pseudo-labelling. The model is shown to work better than the conventional semi-supervised learning methods on small labelled MNIST dataset. This method is optimal for our use-case as the entirety of Reddit data is unlabelled.

2.1.6 Word Embeddings

One of the big challenges of NLP is to find an appropriate representation of words that captures word similarities as they occur in corpus, while also being sufficiently low dimensional. Paper by Pennington et al. [2014] proposes GloVe: Global Vectors for Word Representation, which works well in capturing semantic dependencies. The model combines global matrix factorization and local context window methods to obtain highly efficient word vectors. But we were unable to incorporate this model in our analysis because fastText, another word representation method, gave more reliable results when working with Wikipedia dataset.

Papers by Joshi et al. [2017] and Ju and Yu [2018] are the recent works in NLP that use word embeddings for sentiment analysis. The former studies automated sarcasm detection by conducting experiments using a variety of available models that involve embeddings and classifiers. The latter conducts sentiment analysis using convolutional neural networks paired with different embedding methods like word2vec, GloVe and character level embeddings. These works inspired us to spend a considerable time to study and refine word embeddings because the word representations can cause huge impact on what is learned by the complex models that follow it.

2.1.7 Gated Reccurent Units

While LSTMs work well in most of the datasets that are sufficiently well labelled, they fail to outperform the competitors on smaller datasets. Papers by Chung et al. [2014] does a comparative research on different RNN models. It suggests that GRU (Gated Recurrent Unit) is comparable to well-established LSTM. GRU is a gated model, similar to LSTM, but it does not use memory unit. Further, it makes use of lesser gates than LSTM, thus being slightly better in terms of performance. It has also been found out that GRU works slightly better in case of smaller datasets, which is a reason we decided to go with GRU. In our analysis, we use a variation of GRU, called bi-directional GRU, which is better than vanilla GRU as it is able to capture sequential dependencies in both forward and reverse directions.

2.1.8 The Wikipedia Data-Set

The paper by Wulczyn et al. [2017] describes the dataset that is used in our analysis. The dataset was compiled by a Jigsaw team at Google that crowdsourced the creation of the Wikipedia dataset containing 100,000+ comments labelled with 6 types of toxic behaviour. This is the only labelled dataset that we came across, which uses several levels of toxicity as labels, thus being appropriate for our analysis. We use this labelled dataset in conjunction with pseudo-labelling on unlabelled Reddit data, so as to train our base model.

2.2 Summary

The following table is an overview of our study based on the above papers -

Table 2.1: Summary of Literature Survey

Begin of Table	
Name of the Paper	Contributions
Long Short-Term Memory (Hochreiter and Schmidhuber [1997])	• introduced LSTM. • provided a way to bypass vanishing gradient problem. • suggested an algorithm for learning LSTM weights.

Continuation of Table 2.1	
Name of the Paper	Contributions
Filtering Offensive Language in Online Communities using Grammatical Relations (Xu and Zhu [2010])	• Hate speech detection and filtering. • used automatic sentence-level filtering approach. • matched manual filtering in terms of applicability with reasonable overhead.
How Useful are Your Comments? Analyzing and Predicting YouTube Comments and Comment Ratings (Siersdorfer et al. [2010])	• studied relevancy of comments. • analyzed dependencies between comments, views, comment ratings and topic categories. • showed that user rating of rated comments can be used to filter unrated comments.
Detecting Hate Speech on the World Wide Web (Warner and Hirschberg [2012])	• Hate speech detection. • used sentiment analysis and word sense disambiguation.
An Exploration of Discussion Threads in Social News Sites: A Case Study of the Reddit Community (Weninger et al. [2013])	• studied structure and evolution of Reddit comment threads. • serves as basis for more analysis.

Continuation of Table 2.1	
Name of the Paper	Contributions
Pseudo-Label : The Simple and Efficient Semi-Supervised Learning Method for Deep Neural Networks (Lee [2013])	• Semi-supervised learning technique called pseudo-labeling. • similar to Entropy Regularization. • state-of-the-art performance on small labelled MNIST dataset.
Identifying Social Roles in reddit Using Network Structure (Buntain and Golbeck [2014])	• studied posting behaviour of users on Reddit. • automated 'answer-person' role detection. • showed that users rarely exhibit significant participation in more than one communities.
GloVe: Global Vectors for Word Representation (Pennington et al. [2014])	• introduced GloVe, word representation method. • combined global matrix factorization and local context window methods, both of which were known to work well. • performance of 75% on word analogy task.
Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling (Chung et al. [2014])	• Experimental study on different RNN architectures like LSTM, GRU and tanh units. • Polyphonic music modeling and speech signal modeling tasks. • established that GRUs are comparable to LSTM in terms of performance, and better than traditional tanh units.

Continuation of Table 2.1	
Name of the Paper	Contributions
Hate Speech Detection with Comment Embeddings (Djuric et al. [2015])	• Hate speech detection. • used word embeddings to learn distributed low-dimensional representations of comments. • reported an AUC score of 0.8007.
Exploring Cyberbullying and Other Toxic Behavior in Team Competition Online Games (Kwak et al. [2015])	• Toxicity detection in Online Games. • serve as basis for building systems to detect, prevent and counter-act toxic behaviour. • studies Bystander Effect and vague nature of toxic playing, in-group favoritism and out-group hostility, intra-group conflicts and socio-political factors, team-cohesion and performance.
You Can't Stay Here: The Efficacy of Reddit's 2015 Ban Examined Through Hate Speech (Chandrasekharan et al. [2017])	• studied hate speech with respect to two subreddits. • analyzed if ban by moderators is an effective method of reducing hate speech. • inferred that most banned accounts discontinued usage of Reddit and those that remained reduced hate speech by 80%. • also inferred that those users that migrated to other subreddits did not cause significant chance in hate speech.

Continuation of Table 2.1

Name of the Paper	Contributions
Ex Machina: Personal Attacks Seen at Scale (Wulczyn et al. [2017])	• provided Wikipedia dataset description with several levels of toxicity as labels. • suggested metric (ROC curve) to evaluate model.
Automatic Sarcasm Detection: A Survey (Joshi et al. [2017])	• Automatic sarcasm detection. • analyzed how sarcasm affects sentiment of a sentence. • Semi-supervised pattern extraction to identify implicit sentiment, use of hashtag-based supervision, and use of context beyond target text.
Sentiment Classification with Convolutional Neural Network using Multiple Word Representations (Ju and Yu [2018])	• used Convolutional Neural Network for sentiment classification. • used word2vec, GloVe, character level embeddings together as internal word representation. • reported accuracy of 82.13% on MR-Polarity dataset, 94.60% on MR-Subj dataset and 83.1% on Twitter dataset.

End of Table

2.3 Problem Statement

The primary goal of this project is to assess and analyze toxic behaviour on Reddit. We aim to find the most toxic subreddits and find the correlation between toxic behaviour and the popularity of a comment. Also, we try to find the effect online moderators have on toxic behaviour and compare it with unmoderated comments. Finally we look at the effects created by users who are far more toxic than the general user.

2.4 Objectives

We list the objectives and the goals of the project. The following list are the set of experiments we conducted.

- Top 200 subreddit analysis: We found the most toxic subreddits from the 200 most popular subreddits. Along with the above results, we also found that, in several subreddits toxic comments are likely to have higher comment scores than non-toxic comments. Hence, we provide a list showing this anomaly in the 10 most popular subreddits.

- Effects of moderation on toxic behaviour: In this experiment, we attempted to find the effect of comment moderation on toxic behaviour in the comment section. We did this by comparing two large datasets containing moderated and unmoderated, streamed comments.

- Pareto analysis. We compare the origin of toxic behaviour to a Pareto distribution to observe if a large portion of toxicity comes from a minority on Reddit.

- Effect of toxicity on user activity: Comments threads in Reddit can be visualized as trees. In this experiment, we observe the impact of toxic behaviour in top-level comments on the structure of a typical comment tree.

Chapter 3

Dataset Description

Reddit is a social media site that is structured around various communities and their interests. The content on Reddit is curated, voted upon and discussed by the users of the media site. Although one can access the content anonymously, posting, voting and commenting requires the user to be logged into an account. We chose to work on Reddit data as Reddit is one of the largest social networks in the world with a very active community and because of the fact that a lot of user generated data is publicly available. Despite strict rules prohibiting harassment, Reddit's administrators spend considerable resources on moderating the site. This would sometimes require a moderator to go through thousands of comments per post to remove hate speech.

However, moderators are human and are thus unable to be impartial judges of the enforcement of subreddit rules and guidelines. Moderator bias and censorship is an extremely common complaint on Reddit and more so political subreddits where opposing opinion is often removed in the guise of maintaining civil behaviour and reducing toxicity.

We'll again mention the basic structure and the terminology associated with Reddit before moving onto the experimental dataset.

- Subreddit: Reddit is divided into thousands of completely user-created and moderated communities called subreddits. These are all topic specific forums for content posted and discussed by the users of the subreddit. Science, Politics, WorldNews etc. are some of the popular subreddits with millions of subscribed users each. When a new user joins reddit, the account is automatically subscribed to a bunch of popular subreddits called default subreddits. Each subreddit has its own set of rules that are enforced by moderators of the subreddit.

- Moderators: Every subreddit on the website has a set of users that are appointed as the moderators by the owner(s) of the subreddit. Their job is to ensure that the community rules and guidelines set by that subreddit are followed by the users.

- Posts: Registered users of the website can contribute to subreddits by creating self-posts that have a 10,000 character limit or by linking external sources as content. Each post has its own comment section which is moderated as well.

- Comments: Each post has its own comment section. This section is hierarchically threaded and each thread has a tree-like structure. A new comment can be in response to the post itself or in response to a comment at the root level or in response to any of the sub comments.

- Upvotes and Downvotes: Every post and every comment on reddit, unless archived by the moderators, can be downvoted or upvoted once by any registered user. An upvote adds 1 to the total count and a downvote results in a -1. When sorted by 'hot' or 'best', which is the default sorting order, all posts and comments are sorted based on a ranking function that takes the upvotes count and time into account.

This phase of the project involved building scripts capable of data collection and analysis of datasets to be used for training. We also built several basic models capable of detecting toxic comments. An attempt was made into looking at major features of toxic comments. The project can be found here: https://github.com/aneesh297/MajorProject (currently private. Request for permission)

3.1 Reddit Bot

We built a similar bot for Reddit, another social media platform. Unlike Twitter, application process was much easier and took very little time. When items (links or text posts) are submitted to a subreddit, the users, called "redditors", can vote for or against them (upvote/downvote). Each subreddit has a front page that shows newer submissions that have been rated highly. Redditors can also post comments about the submission, and respond back and forth in a conversation-tree of comments;

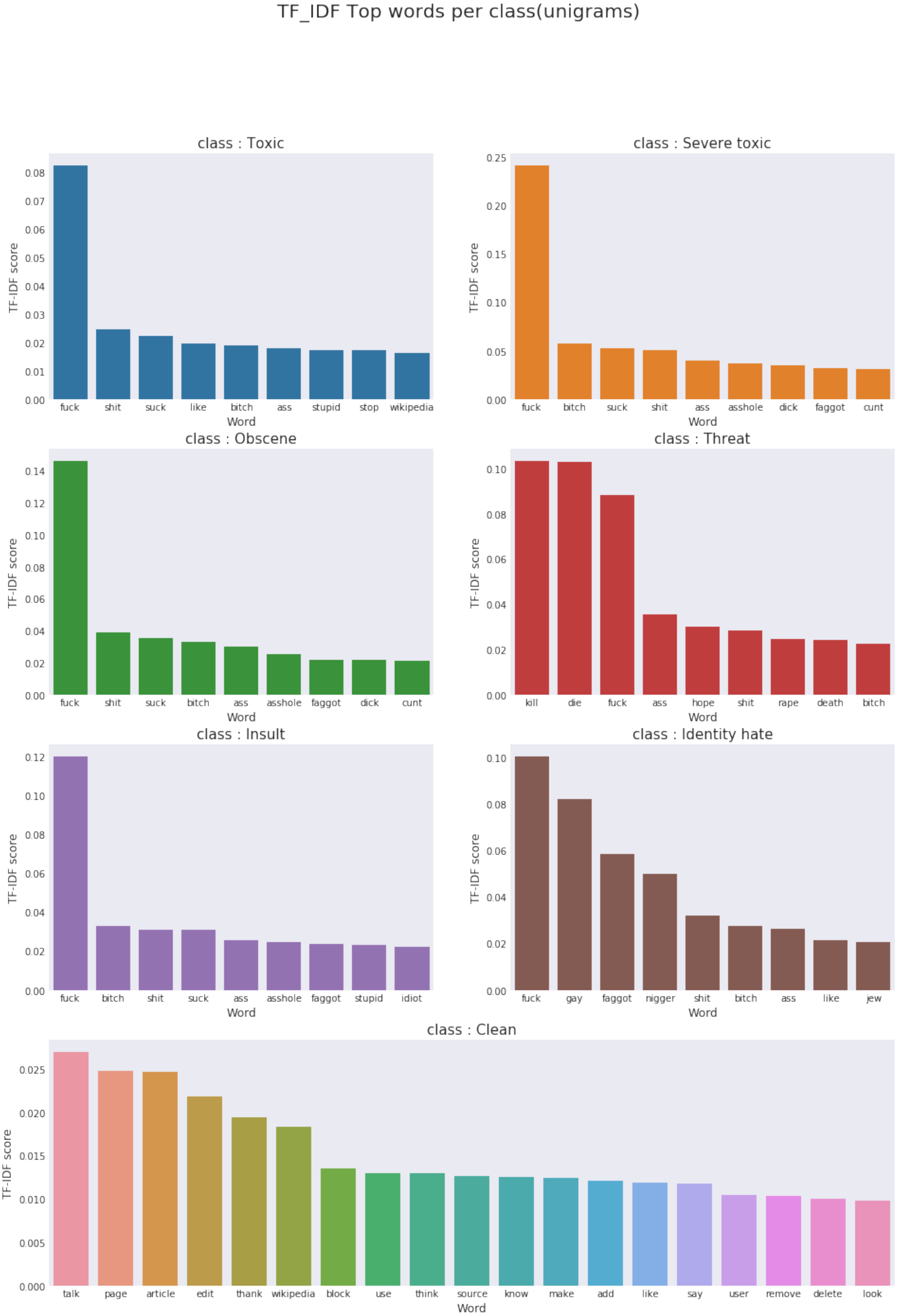


Figure 3.1: TF-IDF top words per class

the comments themselves can also be upvoted and downvoted. The comment section is hierarchically threaded and each thread has a tree-like structure. Essentially the comment section resembles a forest of comment trees. A new comment can be in response to the post itself or in response to a comment at the root level or in response to any of the sub comments. Depending on the experiment we can choose to either fetch comments individually thus destroying the tree structure or choose to maintain the tree structure to preserve dependencies on upper level comments.

Hence, each comment has two contexts, one with respect to the post and another (optionally) to the comment it is replying to. Therefore, the detection of toxicity needs the context of the comment which it is replying to.

We created several bots to collect comments each built separately for an experiment. Separate bots for each experiments were necessary due to the fact that we needed different data for every experiment.

For the first experiment, we created a bot to collect comments from the 200 most popular subreddits. We collected 2000 comments per subreddit from the top posts of the month. This totals to around 400,000 comments totally. We had to filter removed and deleted comments as they would otherwise add to blank comments being collected. Comments are normally removed by moderators for violating subreddit rules or if they feel like removing them. Deleted comments are deleted by a user manually or they can choose to do so if they delete their account.

For the second experiment, we needed both moderated and unmoderated comments. Collecting moderated comments is easy. This can be done by collecting older comments from posts on subreddits that are actively moderated (almost all popular subreddits). We chose r/All for the comment collection as most of Reddit's popular content flows through here and everyone sees this subreddit. For comment collection, we collected 100,000 comments from r/All's top posts of the month that were at least a day old. This gives enough time for moderators to process and act on comments.

For unmoderated comments, we have to find a source of comments where moderators don't go through them ever or is too new for moderators to process. The former is hard to achieve as there are very few popular subreddits that don't have moderators that don't remove comments. Therefore we go with the latter. Python Reddit API wrapper provides a way to stream comments from Reddit, giving a real time stream of comments. These comments are generated instantaneously and are therefore untouched by moderation and user editing. We collected 100,000 streamed comments for the purpose of this experiment. We streamed the comments from r/All as well to ensure uniformity.

The third experiment requires a large amount of comments from each individual subreddit. This is because we wish to capture every possible instance of user interaction in that particular time period. This lets us construct an accurate distribution of user activity to build the Pareto distribution. If we didn't collect all comments from a user in that subreddit, we wouldn't be able to correctly gauge the amount of toxic behaviour that user is responsible. Also, we consider each subreddit individually as Buntain and Golbeck [2014] found that user activity is usually present in one community. This means that they rarely exhibit significant partipation in more than one subreddit.

We choose the top 10 popular subreddits for the experiment as only they are capable of generating the required number of comments. However, one downside is that there are more unique users who are likely to comment more than once. We extract 10,000 comments from each of the top 10 subreddits using the top posts of the month filter. We filter out comments for removed and deleted comments naturally. We also filter one time commenters to get rid of noise in the data. However, this does significantly reduce the amount of data we have.

For the fourth experiment, we need to collect individual comment trees. This is because we need to observe changes in the tree structure of comment threads caused by toxic behaviour. Using the API, we traverse each comment thread in a recursive manner, collecting all comments on a particular level. We store these comments in a tree data structure to match the comment thread structure. This allows for intuitive access of data while processing the data. However, it does slow down access time and requires additional code to support tree operations.

3.2 Twitter Bot

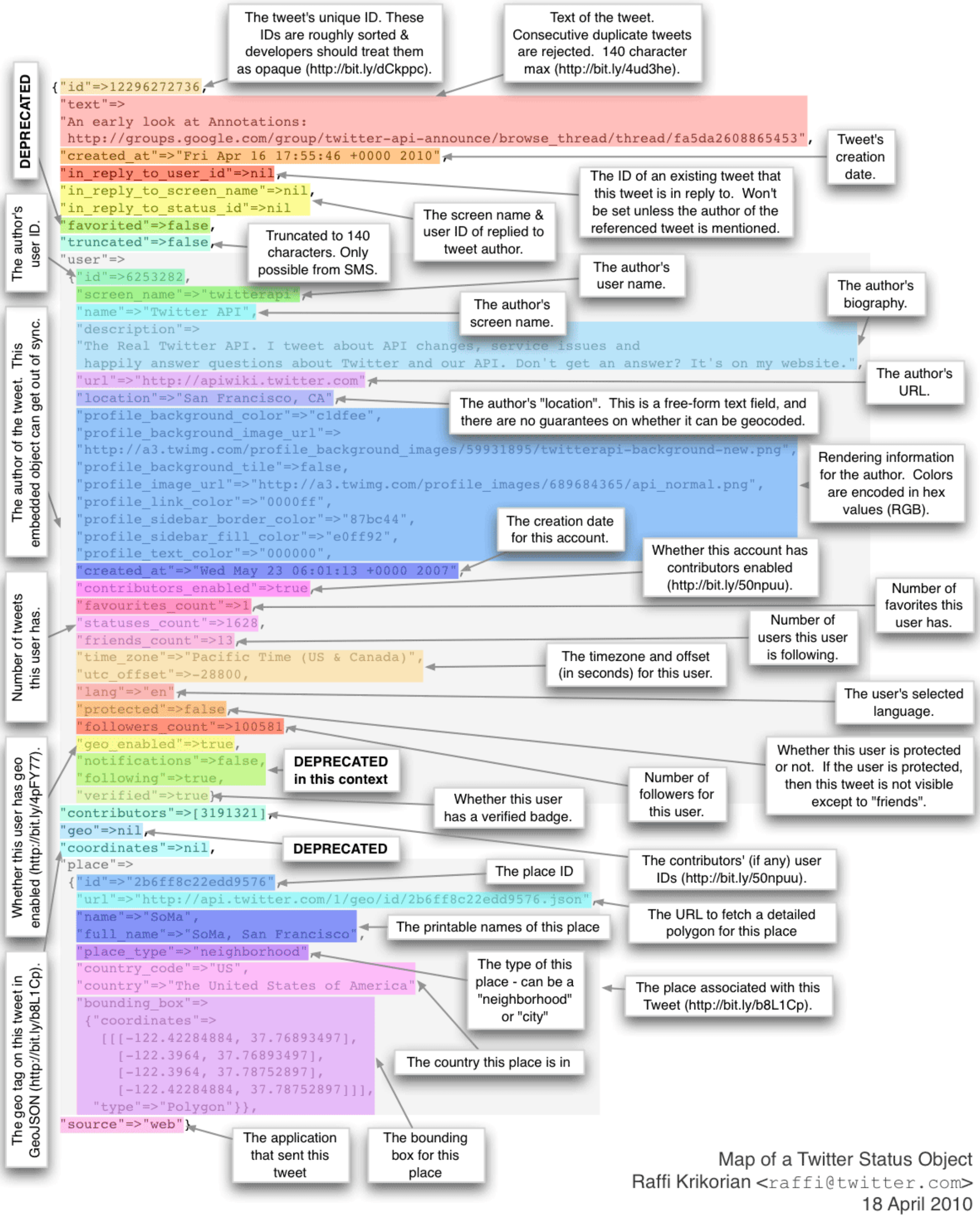
We built a Twitter bot (handle: @majorproject3) that is capable of collecting tweets in real time. The process to making the bot involved several steps. It first required ap-

plying for a developer license. Once the license application was reviewed and granted, we received our API access keys. Using these keys we can now access the twitter account from scripts.
Data collection is made easier by several Twitter API wrappers for Python. There are several filters that allow us to find tweets that we require. Searching by hashtags is also allowed. There are several useful components of a tweet that can be used as detailed in figure 3.2. These features can also be used to profile a toxic user.
We decided to not use Twitter bot and thus, not analyze toxicity on Twitter data because of the complexity involved in the structure of tweets. Moreover, Twitter allowed very few requests to be made by API per unit time. So, we were not able to collect enough comments to train our model. The experiments performed on Reddit would not be applicable to Twitter due to no separate communities being present on Twitter.

3.3 Online Learning

Real time streaming is used in our project to obtain fresh dataset and to support online learning. Since social media data is prone to obsolescence because of fast moving trends and because of topics of discussion that might have never existed at an earlier time, it is important to keep updating the model's weights and vocabulary to make the results relevant to the current trends. Moreover, new forms of toxicity may emerge that might have never been classified as toxic before. Social media specific changes could occur, like emergence of new hash tags in case of twitter and emergence of new subreddits in case of Reddit.

There are problems associated with updating model to the latest standards. It requires supervised learning but the real time data available is unlabelled. This restricts our model in a way that it will only be able to predict the level of toxicity of a real time comment and cannot learn from it. To overcome this problem, manual labelling can be done but doing that will not promote online learning. For onlinne learning, we could assume a base model as ground truth (eg. Google's Perspective) and train our model using its predictions.



```
{"id"=>12296272736,
"text"=>
"An early look at Annotations:
 http://groups.google.com/group/twitter-api-announce/browse_thread/thread/fa5da2608865453",
"created_at"=>"Fri Apr 16 17:55:46 +0000 2010",
"in_reply_to_user_id"=>nil,
"in_reply_to_screen_name"=>nil,
"in_reply_to_status_id"=>nil
"favorited"=>false,
"truncated"=>false,
"user"=>
 {"id"=>6253282,
  "screen_name"=>"twitterapi",
  "name"=>"Twitter API",
  "description"=>
  "The Real Twitter API. I tweet about API changes, service issues and
   happily answer questions about Twitter and our API. Don't get an answer? It's on my website.",
  "url"=>"http://apiwiki.twitter.com",
  "location"=>"San Francisco, CA",
  "profile_background_color"=>"c1dfee",
  "profile_background_image_url"=>
  "http://a3.twimg.com/profile_background_images/59931895/twitterapi-background-new.png",
  "profile_background_tile"=>false,
  "profile_image_url"=>"http://a3.twimg.com/profile_images/689684365/api_normal.png",
  "profile_link_color"=>"0000ff",
  "profile_sidebar_border_color"=>"87bc44",
  "profile_sidebar_fill_color"=>"e0ff92",
  "profile_text_color"=>"000000",
  "created_at"=>"Wed May 23 06:01:13 +0000 2007",
  "contributors_enabled"=>true,
  "favourites_count"=>1,
  "statuses_count"=>1628,
  "friends_count"=>13,
  "time_zone"=>"Pacific Time (US & Canada)",
  "utc_offset"=>-28800,
  "lang"=>"en",
  "protected"=>false,
  "followers_count"=>100581,
  "geo_enabled"=>true,
  "notifications"=>false,
  "following"=>true,
  "verified"=>true},
"contributors"=>[3191321],
"geo"=>nil,
"coordinates"=>nil,
"place"=>
 {"id"=>"2b6ff8c22edd9576",
  "url"=>"http://api.twitter.com/1/geo/id/2b6ff8c22edd9576.json",
  "name"=>"SoMa",
  "full_name"=>"SoMa, San Francisco",
  "place_type"=>"neighborhood",
  "country_code"=>"US",
  "country"=>"The United States of America",
  "bounding_box"=>
   {"coordinates"=>
     [[[-122.42284884, 37.76893497],
       [-122.3964, 37.76893497],
       [-122.3964, 37.78752897],
       [-122.42284884, 37.78752897]]],
    "type"=>"Polygon"}},
"source"=>"web"}
```

DEPRECATED
The tweet's unique ID. These IDs are roughly sorted & developers should treat them as opaque (http://bit.ly/dCkppc).
Text of the tweet. Consecutive duplicate tweets are rejected. 140 character max (http://bit.ly/4ud3he).
Tweet's creation date.
The ID of an existing tweet that this tweet is in reply to. Won't be set unless the author of the referenced tweet is mentioned.
The screen name & user ID of replied to tweet author.
Truncated to 140 characters. Only possible from SMS.
The author's user ID.
The author's user name.
The author's screen name.
The author's biography.
The author's URL.
The author's "location". This is a free-form text field, and there are no guarantees on whether it can be geocoded.
Rendering information for the author. Colors are encoded in hex values (RGB).
The author of the tweet. This embedded object can get out of sync.
The creation date for this account.
Whether this account has contributors enabled (http://bit.ly/50npuu).
Number of favorites this user has.
Number of tweets this user has.
Number of users this user is following.
The timezone and offset (in seconds) for this user.
The user's selected language.
Whether this user is protected or not. If the user is protected, then this tweet is not visible except to "friends".
Number of followers for this user.
DEPRECATED in this context
Whether this user has a verified badge.
Whether this user has geo enabled (http://bit.ly/4pFY77).
The contributors' (if any) user IDs (http://bit.ly/50npuu).
DEPRECATED
The place ID
The URL to fetch a detailed polygon for this place
The printable names of this place
The type of this place - can be a "neighborhood" or "city"
The place associated with this Tweet (http://bit.ly/b8L1Cp).
The country this place is in
The geo tag on this tweet in GeoJSON (http://bit.ly/b8L1Cp).
The application that sent this tweet
The bounding box for this place

Map of a Twitter Status Object
Raffi Krikorian <raffi@twitter.com>
18 April 2010

Figure 3.2: Map of a Twitter Object

3.4 Kaggle

Kaggle recently hosted a competition titled Toxic Comment Classification Challenge which required the contestant to build models to predict different types of toxicity. The dataset provided in the contest was comments from Wikipedia's talk page for edits. It contains 160,000 comments with around 30000 toxic comments. Comments

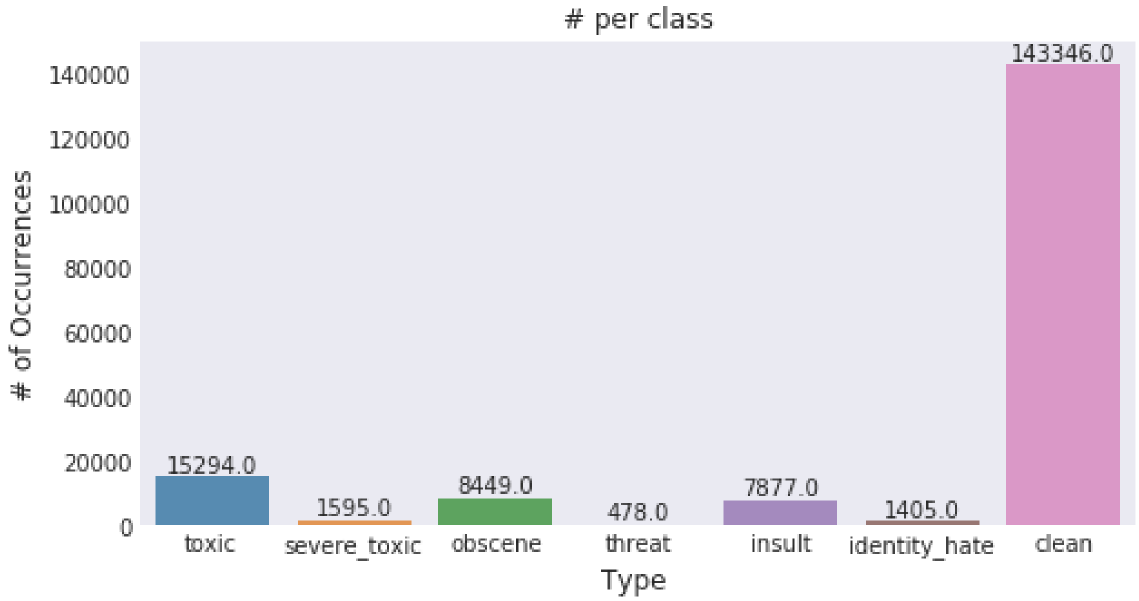


Figure 3.3: Distribution of the Wikipedia Dataset

are labeled as Toxic, Severe Toxic, Obscene, Insult, or Threat. The distribution of the data set can be seen in Figure 3.3. A comment can have more than one label. For example, a severe toxic comment is always also toxic. A clean comment has all labels set to 0.

The contest was evaluated on the mean column-wise Receiver Operating Characteristic Area Under the Curve (ROC AUC).AUC - ROC curve is a performance measurement for classification problem at various thresholds settings. ROC is a probability curve and AUC represents degree or measure of separability. It tells how much model is capable of distinguishing between classes. Higher the AUC, better the model is at predicting 0s as 0s and 1s as 1s. The ROC curve is created by plotting the true positive rate (TPR) against the false positive rate (FPR) at various threshold settings. The True Positive Rate is commonly known as Recall. Intuitively this metric corresponds to the proportion of positive data points that are correctly considered as positive, with respect to all positive data points. In other words, the higher TPR, the fewer positive data points we will miss. The False Positive Rate is known as Precision. Intuitively this metric corresponds to the proportion of negative data points that are mistakenly considered as positive, with respect to all negative data points. In other words, the higher False Positive Rate, the more negative data points will be miss-classified.

The AUROC has several equivalent interpretations:

- The expectation that a uniformly drawn random positive is ranked before a uniformly drawn random negative.
- The expected proportion of positives ranked before a uniformly drawn random negative.
- The expected true positive rate if the ranking is split just before a uniformly drawn random negative.
- The expected proportion of negatives ranked after a uniformly drawn random positive.
- The expected false positive rate if the ranking is split just after a uniformly drawn random positive.

Since the contest requires to predict several categories of toxicity, the average of the AUCs of each individual predicted category is taken. A perfect model has a AUROC score of 1.

We used this competition as a testing ground for our models due to its well labeled dataset.

Chapter 4

Analysis and Design

4.1 Model Design and Description

We built two models using the Wikipedia training data set. The first model is a linear regression classifier that uses tf-idf weights. This was our primary model for most of predictions due to it be light to load and faster for inference.

4.1.1 Logistic Regression Classifier with tf-idf weights

The first model uses tf-idf to assign weights to tokens in the data set.

Tf-idf stands for term frequency-inverse document frequency. This weight is a statistical measure used to evaluate how important a word is to a document in a collection or corpus. The tf-idf value increases proportionally to the number of times a word appears in the document and is offset by the number of documents in the corpus that contain the word, which helps to adjust for the fact that some words appear more frequently in general. Tf–idf is one of the most popular term-weighting schemes today.

Typically, the tf-idf weight is composed of two terms: the first computes the normalized Term-Frequency (TF), i.e. the number of times a word appears in a document, divided by the total number of words in that document; the second term is the Inverse Document Frequency (IDF), computed as the logarithm of the number of the documents in the corpus divided by the number of documents where the specific term appears.

A Term Frequency

Term Frequency, which measures how frequently a term occurs in a document. Since every document is different in length, it is possible that a term would appear much more times in long documents than shorter ones. Thus, the term frequency is often divided by the document length (aka. the total number of terms in the document) as a way of normalization:

$$tf(t) = \frac{Number\ of\ times\ term\ t\ appears\ in\ a\ document}{Total\ number\ of\ terms\ in\ the\ document} \tag{4.1}$$

B Inverse Document Frequency

Inverse Document Frequency, which measures how important a term is. While computing TF, all terms are considered equally important. However it is known that certain terms, such as "is", "of", and "that", may appear a lot of times but have little importance. Thus we need to weigh down the frequent terms while scale up the rare ones, by computing the following:

$$idf(t) = log_e(\frac{Total\ number\ of\ documents}{Number\ of\ document\ with\ term\ t\ in\ it}) \tag{4.2}$$

In our scenario, tf-idf helps us find any words that are more frequent in toxic comments while ignoring words that are more frequent in general. It is possible to use these weights as features to pass to models.

There are two kinds of features that can be generated using tf-idf. One, using just words tokens, we can find frequent words. The second, uses character n-grams to generate features. Hence, the features generated will be for 2-6 character n-grams. This allows us to correct for miss-spelled words. This is because, while miss-spelled words are different from their correct counterparts, they still maintain the overall structure of the word. Hence they would have a few similar characters with the original word. Without this, they miss-spelled words would have been otherwise categorized as different features thus creating unnecessary features. However, generation of these features takes much longer due to the generation of a lot of character tokens. The training time of the model is also somewhat increased by this. Another approach for someone who wants to reduce the training time and the feature generation time could use a spell-checker/corrector to find and correct miss-spelled words. These features can then be used to train a classifier which can predict the presence and type of toxicity of a comment.

We used a logistic regression classifier with a Stochastic Average Gradient solver. the logistic model (or logit model) is a widely used statistical model that in its basic form uses a logistic function to model a binary dependent variable, although many more complex extensions exist. Mathematically, a binary logistic model has a dependent variable with two possible values, such as pass/fail, win/lose, alive/dead or healthy/sick; these are represented by an indicator variable, where the two values are labeled "0" and "1". In this case, the results of the logistic model is probability of a comment to be toxic in that particular category. Hence, we need 6 separate logistic regression models to predict each type of toxicity. By combining the results of all models, we obtained the average AUC score required for the submission.

In the logistic model, the log-odds (the logarithm of the odds) for the value labeled "1" is a linear combination of one or more independent variables ("predictors"); the independent variables can each be a binary variable (two classes, coded by an indicator variable) or a continuous variable (any real value). The binary logistic regression model has extensions to more than two levels of the dependent variable: categorical outputs with more than two values are modeled by multinomial logistic regression, and if the multiple categories are ordered, by ordinal logistic regression, for example the proportional odds ordinal logistic model.

Stochastic Gradient Descent (SGD) is a simple yet very efficient approach to discriminative learning of linear classifiers under convex loss functions such as (linear) Support Vector Machines and Logistic Regression. Even though SGD has been around in the machine learning community for a long time, it has received a considerable amount of attention just recently in the context of large-scale learning.

Stochastic gradient descent, also known as incremental gradient descent, is an iterative method for optimizing a differentiable objective function, a stochastic approximation of gradient descent optimization. It is called stochastic because samples are selected randomly (or shuffled) instead of as a single group (as in standard gradient descent) or in the order they appear in the training set.

Batch methods, such as limited memory BFGS, which use the full training set to compute the next update to parameters at each iteration tend to converge very well to local optima. They are also straight forward to get working provided a good off the shelf implementation (e.g. minFunc) because they have very few hyper-parameters to tune. However, often in practice computing the cost and gradient for the entire training set can be very slow and sometimes intractable on a single machine if the dataset is too big to fit in main memory. Another issue with batch optimization methods is that they don't give an easy way to incorporate new data in an 'online' setting. Stochastic Gradient Descent (SGD) addresses both of these issues by following the negative gradient of the objective after seeing only a single or a few training examples.

In SGD the learning rate is typically much smaller than a corresponding learning rate in batch gradient descent because there is much more variance in the update. Choosing the proper learning rate and schedule (i.e. changing the value of the learning rate as learning progresses) can be fairly difficult. One standard method that works well in practice is to use a small enough constant learning rate that gives stable convergence in the initial epoch (full pass through the training set) or two of training and then halve the value of the learning rate as convergence slows down. An even better approach is to evaluate a held out set after each epoch and anneal the learning rate when the change in objective between epochs is below a small threshold. This tends to give good convergence to a local optima.

One final but important point regarding SGD is the order in which we present the data to the algorithm. If the data is given in some meaningful order, this can bias the gradient and lead to poor convergence. Generally a good method to avoid this is to randomly shuffle the data prior to each epoch of training.

SGD has been successfully applied to large-scale and sparse machine learning problems often encountered in text classification and natural language processing.

The advantages of Stochastic Gradient Descent are:

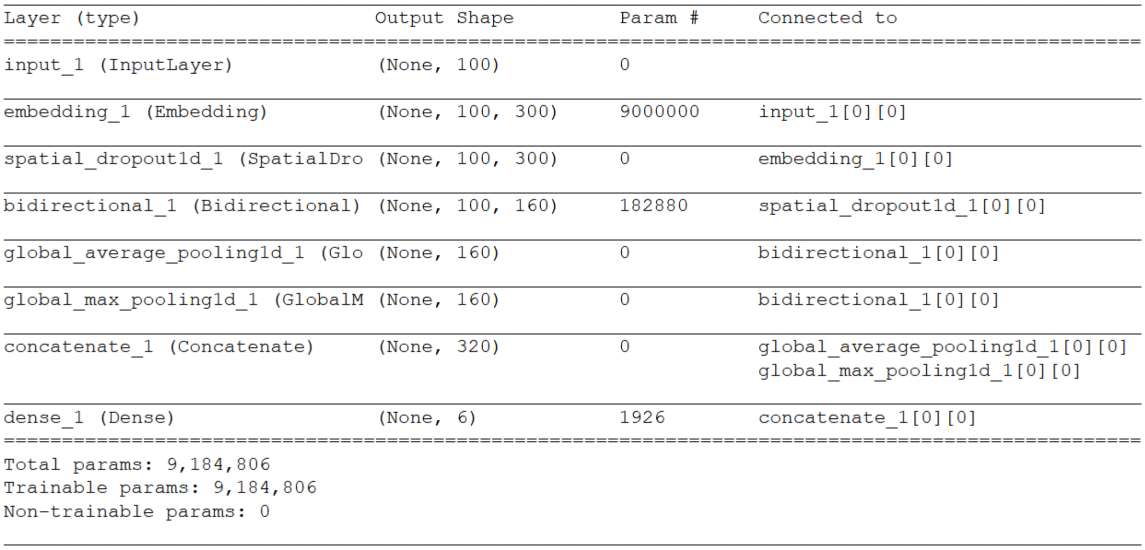
- It is highly efficient
- Ease of implementation (lots of opportunities for code tuning).

The disadvantages of Stochastic Gradient Descent include:

- SGD requires a number of hyperparameters such as the regularization parameter and the number of iterations.

- SGD is sensitive to feature scaling.

The model uses both the word and character tf-idf features of the data set as the parameters to the model. On Kaggle The model when tested on the Wikipedia dataset gave a 0.97565 public score and a 0.97644 private score on the Kaggle leaderboard. We used this model primarily for our experiments as this model was much lighter to load and faster to run. We've also tried using a Support Vector Machine model but it had inferior performance. The features detected by tf-idf for toxic comments can be found in Figure 3.1.



```
Layer (type)                    Output Shape         Param #     Connected to
==================================================================================================
input_1 (InputLayer)            (None, 100)          0
__________________________________________________________________________________________________
embedding_1 (Embedding)         (None, 100, 300)     9000000     input_1[0][0]
__________________________________________________________________________________________________
spatial_dropout1d_1 (SpatialDro (None, 100, 300)     0           embedding_1[0][0]
__________________________________________________________________________________________________
bidirectional_1 (Bidirectional) (None, 100, 160)     182880      spatial_dropout1d_1[0][0]
__________________________________________________________________________________________________
global_average_pooling1d_1 (Glo (None, 160)          0           bidirectional_1[0][0]
__________________________________________________________________________________________________
global_max_pooling1d_1 (GlobalM (None, 160)          0           bidirectional_1[0][0]
__________________________________________________________________________________________________
concatenate_1 (Concatenate)     (None, 320)          0           global_average_pooling1d_1[0][0]
                                                                 global_max_pooling1d_1[0][0]
__________________________________________________________________________________________________
dense_1 (Dense)                 (None, 6)            1926        concatenate_1[0][0]
==================================================================================================
Total params: 9,184,806
Trainable params: 9,184,806
Non-trainable params: 0
__________________________________________________________________________________________________
```

Figure 4.1: Model Description for bi-GRU (model 2)

4.1.2 Bi-GRU with fastText embeddings and Pseduo-Labeling

The second model was built using gated recurrent units. Later on, we also used the help of pseudo-labeling to further train the model using Reddit data.

Gated recurrent units (GRUs) are a gating mechanism in recurrent neural networks, introduced in 2014 by Cho et al.. GRU aims to solve vanishing gradient problem by making use of two gates, namely, update and reset gate.

The GRU is like a long short-term memory (LSTM) with forget gate but has fewer parameters than LSTM, as it lacks an output gate. Basically, these are two vectors which decide what information should be passed to the output. The special thing about them is that they can be trained to keep information from long ago, without washing it through time or remove information which is irrelevant to the prediction.

The update gate helps the model to determine how much of the past information (from previous time steps) needs to be passed along to the future. That is really powerful because the model can decide to copy all the information from the past and eliminate the risk of vanishing gradient problem.

The reset gate is used from the model to decide how much of the past information to forget. If trained well, GRUs can be used even in the most complex scenarios without facing vanishing gradient problem. The performance of GRU on polyphonic music modeling and speech signal modeling was found to be similar to that of long short-term memory (LSTM).

However,the LSTM is to some extent stronger than the GRU as it can easily perform unbounded counting, while the GRU cannot. That's why the GRU fails to learn simple languages that are learnable by the LSTM. However, the English language isn't a simple language. GRUs have been shown to exhibit better performance on smaller datasets. A basic GRU's structure can be observed in Figure 4.2.

A GRU mostly benefits from all the things a Recurrent Neural Network benefits from. Their internal memory capability allow them to exhibit temporal dynamic behaviour for a time sequence. Unlike feedforward neural networks, RNNs can use their internal state (memory) to process sequences of inputs. This makes them ideal for application such as speech recognition, time-series forecasting, image/music/dance generation and conversation modeling/question answering.

We used a variation of GRU, called Bidirectional GRU, which basically trains the model twice per sequence, once forward and then reverse, so that the model isn't learns information in both directions.

We use FastText to create word-embeddings to feed to the bi-GRU as input.

Word embedding is the collective name for a set of language modeling and feature learning techniques in natural language processing (NLP) where words or phrases from the vocabulary are mapped to vectors of real numbers. Conceptually it involves

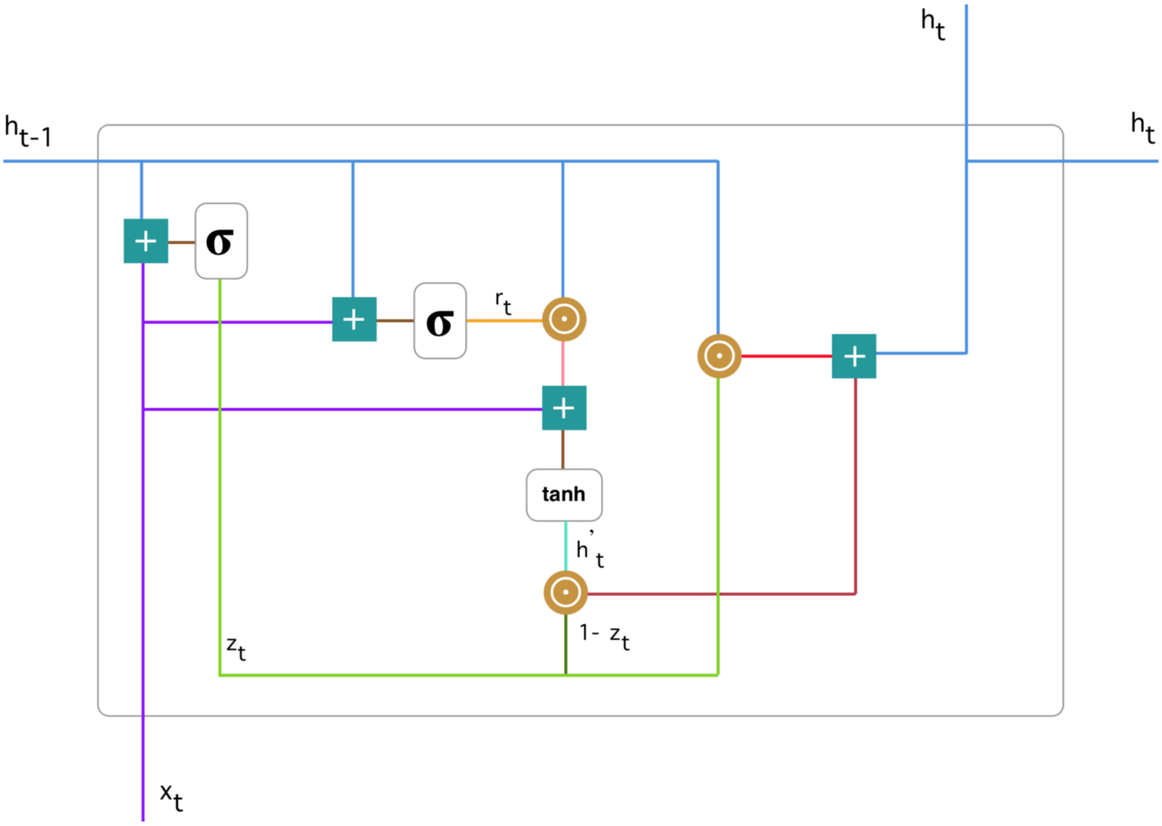


Figure 4.2: Single Gated Recurrent Unit

a mathematical embedding from a space with one dimension per word to a continuous vector space with a much lower dimension.

Methods to generate this mapping include neural networks, dimensionality reduction on the word co-occurrence matrix, probabilistic models, explainable knowledge base method,and explicit representation in terms of the context in which words appear.

Word and phrase embeddings, when used as the underlying input representation, have been shown to boost the performance in NLP tasks such as syntactic parsing and sentiment analysis.

One of the main limitations of word embeddings (word vector space models in general) is that possible meanings of a word are conflated into a single representation (a single vector in the semantic space). Sense embeddings are proposed as a solution to this problem: individual meanings of words are represented as distinct vectors in the space.

There are many branches and many research groups working on word embeddings. In 2013, a team at Google led by Tomas Mikolov created word2vec, a word embedding toolkit which can train vector space models faster than the previous approaches. Most new word embedding techniques rely on a neural network architecture instead of more traditional n-gram models and unsupervised learning.

Word2vec is a group of related models that are used to produce word embeddings. These models are shallow, two-layer neural networks that are trained to reconstruct linguistic contexts of words. Word2vec takes as its input a large corpus of text and produces a vector space, typically of several hundred dimensions, with each unique word in the corpus being assigned a corresponding vector in the space. Word vectors are positioned in the vector space such that words that share common contexts in the corpus are located in close proximity to one another in the space.

Word2vec can utilize either of two model architectures to produce a distributed representation of words: continuous bag-of-words (CBOW) or continuous skip-gram. In the continuous bag-of-words architecture, the model predicts the current word from a window of surrounding context words. The order of context words does not influence prediction (bag-of-words assumption). In the continuous skip-gram architecture, the model uses the current word to predict the surrounding window of context words. The skip-gram architecture weighs nearby context words more heavily than more distant context words. According to the authors' note,CBOW is faster while skip-gram is slower but does a better job for infrequent words.

The word embedding approach is able to capture multiple different degrees of similarity between words. Mikolov et al. [2013] at Google found that semantic and syntactic patterns can be reproduced using vector arithmetic. Patterns such as "Man is to Woman as Brother is to Sister" can be generated through algebraic operations on the vector representations of these words such that the vector representation of "Brother" - "Man" + "Woman" produces a result which is closest to the vector representation of "Sister" in the model. Such relationships can be generated for a range of semantic relations (such as Country–Capital) as well as syntactic relations (e.g. present tense–past tense). An intuitive representation and understanding of word vectors can be found in Figure 4.3

fastText is a library for learning of word embeddings and text classification created by Mikolov et al. [2018] at Facebook's AI Research (FAIR) lab. The model allows to create an unsupervised learning or supervised learning algorithm for obtaining

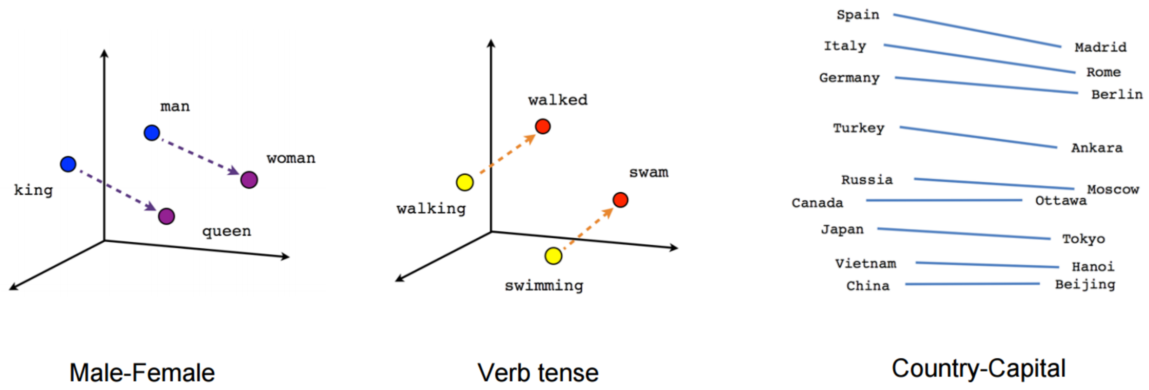


Figure 4.3: An intuitive understanding of word vectors.
Source: Analytics Vidya

vector representations for words. Facebook makes available pretrained models for 294 languages. fastText uses a neural network for word embedding. We use crawl-300d-2M.vec: where 2 million word vectors are trained on Common Crawl. It contains 600 billion tokens. The vectors are of 300 dimensions each.

fastText uses a hashtable for either word or character ngrams. This leads to the extremely large sizes of the model, making it hard to load. One of the key features of fastText word representation is its ability to produce vectors for any words, even made-up ones. Indeed, fastText word vectors are built from vectors of substrings of characters contained in it. This allows to build vectors even for misspelled words or concatenation of words.

Bidirectional GRU's are a type of bidirectional recurrent neural networks with only the input and forget gates. It allows for the use of information from both previous time steps and later time steps to make predictions about the current state. Figure 4.4 displays the typical structure of a bidirectional recurrent neural network. Replacing every A and A' in the diagram with a gated recurrent unit yields the bidirectional GRU.

This model (Bi-GRU) achieved a cross-validation AUC score of 0.9871. It uses FastText word embeddings to convert words into vectors. Training of this model was done on Google Colab. Colaboratory (also known as Colab) is a free Jupyter notebook environment that runs in the cloud and stores its notebooks on Google Drive. It offers free GPU access (NVIDIA K80) for training. We haven't used this model to make predictions for observations as it is heavier to load and requires more time to make

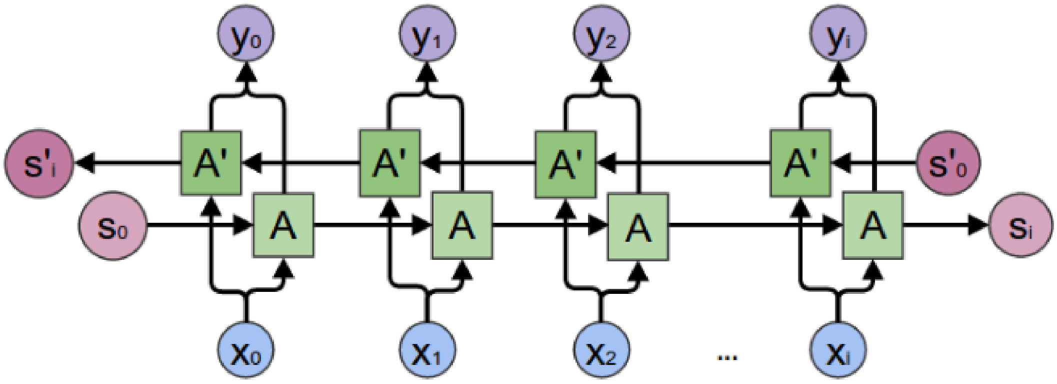


Figure 4.4: General structure of a bidirectional gated recurrent unit

predictions. The structure of the model can be found on Figure 4.1.

Once this model is trained on Wikipedia dataset, we use the learnt weights to obtain labels for the Reddit comment data. These labels are used as ground truth to retrain the model along with the Wikipedia data. This methodology of semi-supervised learning is called pseudo labelling and is known to work well. In this technique, instead of manually labeling the unlabelled data, we give approximate labels on the basis of the labelled data. Finally, the re-trained model is used for analysis.

The following is a brief explaination of pseudo-labelling, introduced by Lee [2013]. It is a simple and efficient way of semi-supervised learning for neural networks. It does not require any complex training scheme or any computationally expensive similarity matrix. The proposed method was found to show the state-of-the-art performance. Basically, the proposed network is trained in a supervised fashion with labeled and unlabeled data simultaneously. For unlabeled data, Pseudo-Labels, just picking up the class which has the maximum predicted probability every weights update, are used as if they were true labels. It favors a low-density separation between classes, a commonly assumed prior for semi-supervised learning. In principle, this method can combine almost all neural network models and training methods. Figure 4.5 should give a good enough idea to understand the pseduo-labeling process.

Pseudo-Labeling is similar to entropy regularization. The conditional entropy of the class probabilities can be used for a measure of class overlap. By minimizing the entropy for unlabeled data, the overlap of class probability distribution can be

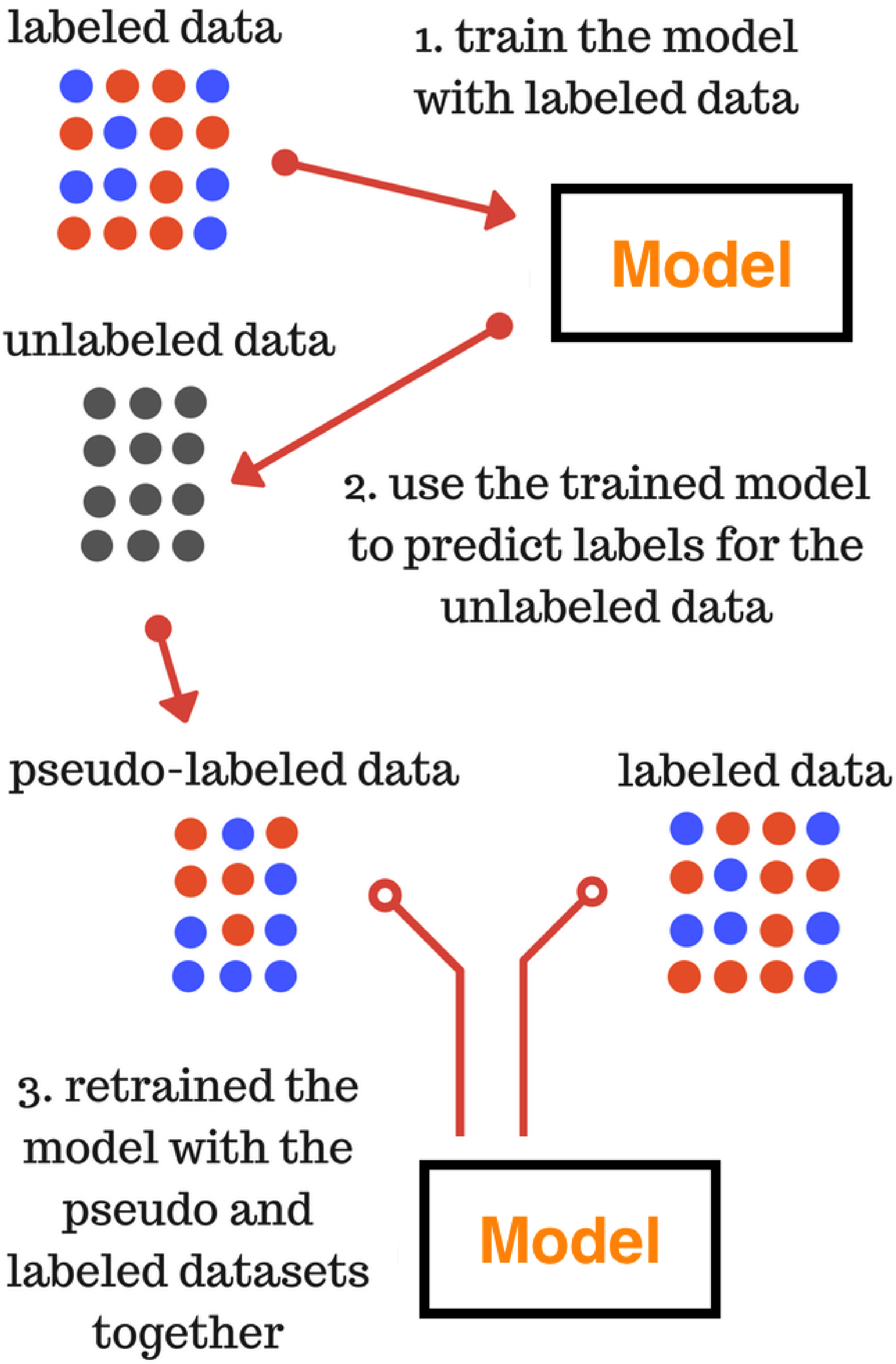

labeled data
1. train the model with labeled data
Model
unlabeled data
2. use the trained model to predict labels for the unlabeled data
pseudo-labeled data
labeled data
3. retrained the model with the pseudo and labeled datasets together
Model

Figure 4.5: The pseudo-labelling process

reduced. It favors a low-density separation between classes, a commonly assumed prior for semi-supervised learning.

Entropy regularization (Grandvalet and Bengio [2005]) is another semi-supervised learning technique, which encourages the classifier to make confident predictions on unlabeled data. For example, we'd prefer an unlabeled point to be assigned a high probability of being in a particular class, rather than diffuse probabilities spread over multiple classes. The purpose is to take advantage of the assumption that the data are clustered according to class (called the "cluster assumption" in semi-supervised learning). So, nearby points have the same class, and points in different classes are more widely separated, such that the true decision boundaries run through low density regions of input space.

Pseudo-Labeling works well under two conditions. Firstly, the base model has to be trained on sufficient amount of labelled data so that it can learn accurate clusters for each class. Then pseudo-labeling helps by refining those clusters according to the predictions that the model makes on unlabelled data. This leads us to the second condition that the base model has to make accurate initial predictions so that the refining done by pseudo-labeling is accurate.

We use the pseudo-labels in a fine-tuning phase with Dropout. The pre-trained network is trained in a supervised fashion with labeled and unlabeled data simultaneously. For unlabeled data, Pseudo-Labels recalculated every weights update are used for the same loss function of supervised learning task.

4.2 Minor Observations

We made several observations while designing these models. All these observations are relevant to the Wikipedia data set.

- The length of the comment has no relation with the toxicity of a comment. This finding isn't surprising as there are ways to be toxic using few or many words. Also, the chance a toxic word will appear in comment increases with the length of the comment. Therefore, even long comments can be toxic due to occurrences of such words. The above finding can be observed and understood in figure. 4.6.

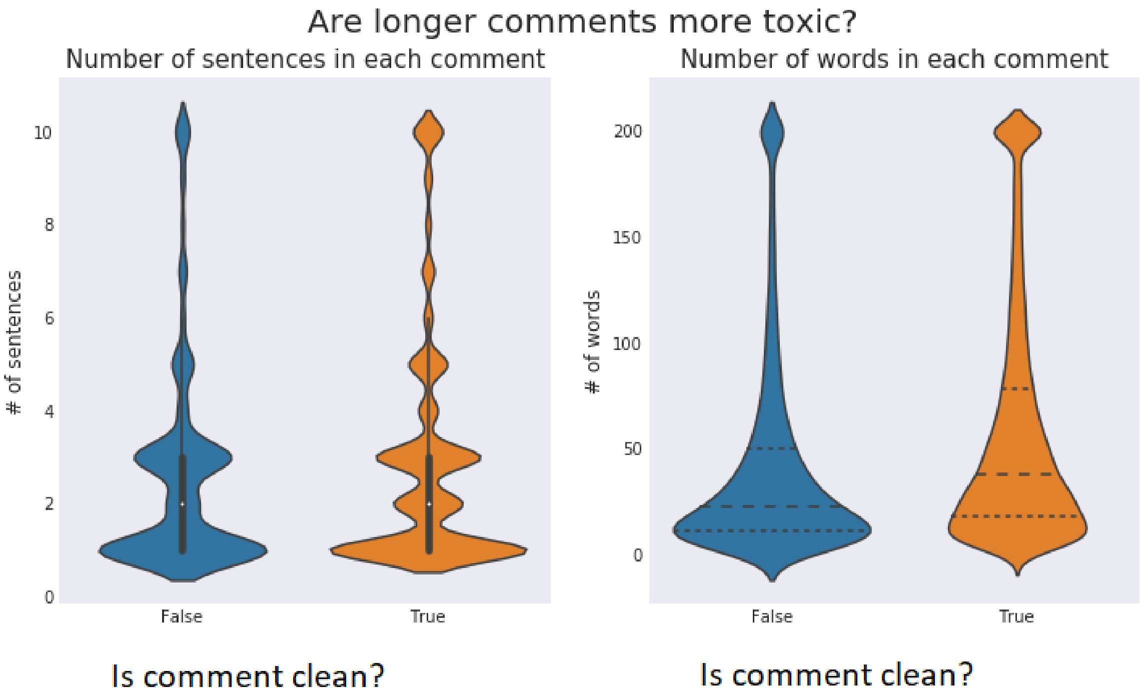


Figure 4.6: Long sentences or more words do not seem to be a significant indicator of toxicity.

- A comment with low number of unique words (<30%) is much more likely to be toxic. As toxic behaviour requires the use of certain words, a toxic user is likely to string several of them together, decreasing the unique count of words. This also can be used to conclude that toxic comments are less likely to contribute to discussions to the lack of unique content being presented. This should be a motivator for social media administrators to remove toxic content to encourage productive discussion. In other words spammers are more toxic.

This can be illustrated in figure 4.7. The first chart is a split violin chart. It is a variation of the traditional box chart/violin chart which allows us to split the violin in the middle based on a categorical variable. There is a bulge near the 0-10% mark which indicates a large number of toxic comments which contain very little variety of words. The second chart is an overlay of two kernel density estimation plots of percentage of unique words out of all the words in the comment, done for both clean and toxic comments.

Even though the number of clean comments dominates the data set(90%), there are only 75 clean comments that are spam, which makes it a powerful indicator of a toxic comment.

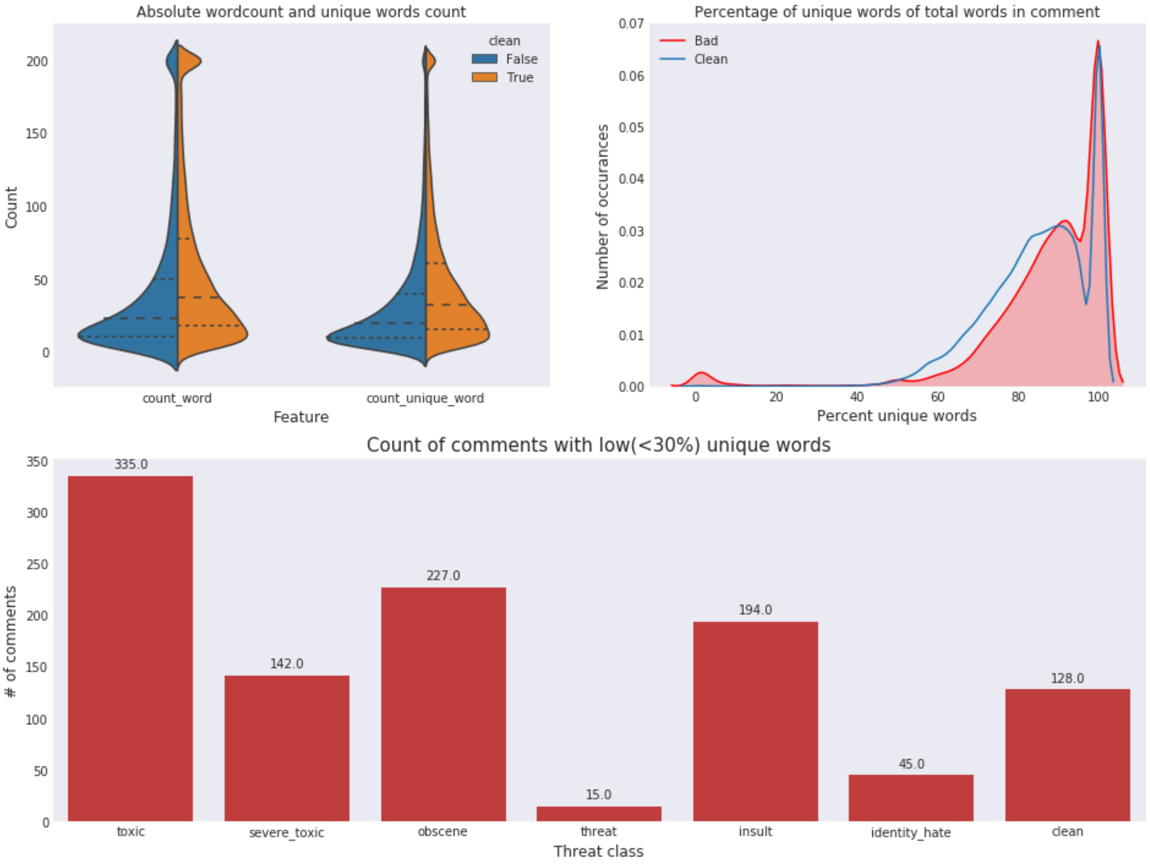


Figure 4.7: Uniqueness vs Toxicity

- Figure 3.1 gives the top features detected by TF-IDF for word features for the respective catgories. Naturally, as expected, most offensive words in the English language have the highest tf-idf scores assigned to them as they are commonly used in toxic sentences. W

  There is some variation of top word features detected for each category. This makes sense as there is some difference in the forms of toxic behaviour.

  However, clean tf-idf word features seem to be a bit different from the other categories as some one of the words seem to be relevant to Wikipedia, that is, the original data set used for training. Therefore, there is some minor bias in detecting clean comments due to the original data set.

- Certain features like a toxic user's IP address, user names, links, article IDs can be identified by the model as toxic. This is harmful to the model's integrity as these are leaky features that can lead to over-fitting. These features must be removed before the model is trained.

- Coming to observations on Reddit, in most subreddits, toxic comments are found to be voted much lower than non-toxic comments. Intuitively, it makes sense that people will react negatively to toxic comments. However, there are certain subreddits that we found this pattern was not obeyed. In subreddits known to be toxic, toxic comments have higher number of votes.
- The model sometimes fails to discriminate obscene language being used in a non-toxic manner from toxic behaviour.
- The model does detect higher levels of toxicity in subreddits that are known to the community to be toxic. However, due to censorship or high levels of moderation, the data collected fails to paint a full picture of the comments. One way to address this issue would be to collect only real-time comments, as they cannot be removed or censored immediately after being posted. However, this makes collection of data harder in less popular subreddits as there is less activity on that subreddit to collect the number of comments required. We observe the effect of moderation on toxic behaviour in the next section.

Chapter 5

Experiment Setup, Results and Discussion

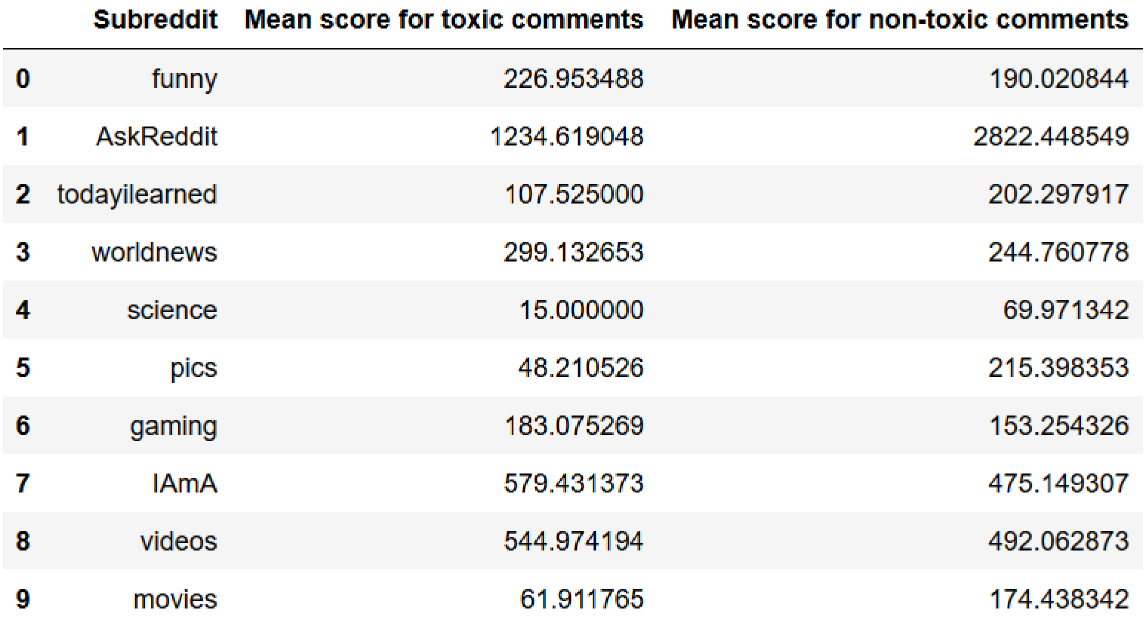
5.1 Top 200 subreddit analysis

A lot of Reddit's interaction occurs within the subreddits with the highest subscriber counts. Therefore, the first experiment was to find the varying toxicity level in each subreddit from the top 200 subreddits. We used the list of top 200 subreddits provided on Reddit. Since the topic covered on every subreddit is different, the type and number of users participating will differ across every subreddit. Naturally the type and level of toxic behaviour on each subreddit will vary as well. Knowing which community is especially toxic would be ideal for a new user on the website. This information would be useful to advertisers on Reddit as they wouldn't want to be associated publicly with toxic subreddits.

There are several things we can measure here. First, we can measure average toxic behaviour across a subreddit. This would let us build rank subreddits in the order of their toxicity. We do this by collecting comments from top of the month posts from each of the top 200 subreddits. Figure 5.2 contains the top 10 most toxic subreddits in each of the given categories. We note that some subreddits feature on several of these graphs. This indicates that the forms of toxic behaviour are to some extent correlated and that those subreddits are especially toxic.

Second, we can find which subreddit encourages/approves of toxic behaviour more by finding the correlation between the upvote count and toxicity of the comment. Subreddit that encourage negative behaviour are likely to have a positive correlation between toxic behaviour and upvotes. The results of this experiment can be found in

Figure 5.1.



	Subreddit	Mean score for toxic comments	Mean score for non-toxic comments
0	funny	226.953488	190.020844
1	AskReddit	1234.619048	2822.448549
2	todayilearned	107.525000	202.297917
3	worldnews	299.132653	244.760778
4	science	15.000000	69.971342
5	pics	48.210526	215.398353
6	gaming	183.075269	153.254326
7	IAmA	579.431373	475.149307
8	videos	544.974194	492.062873
9	movies	61.911765	174.438342

Figure 5.1: Mean scores for toxic and non-toxic comments

We listed the top 10 most toxic subreddits in each category in below figure. Regarding the relation between toxic comments and upvote counts found in Fig 4., we found several subreddits where toxic comments often had higher upvote counts. This can indicate that toxic behaviour is either encouraged or is not truly toxic enough to offend.

5.2 Effects of moderation

Moderators on every subreddit are responsible for ensuring that all posts and comment follow subreddit guidelines and to remove toxic comments. However, there are many subreddits where moderators abuse their powers to discriminate against users they do not like by banning or censoring them. Also, several political subreddits allow moderators to ban users who are opposed to the subreddit's political narrative. These complaints have been a constant thing on Reddit since its inception. Therefore, we wanted to observe if moderators do reduce the amount of toxic behaviour.

To do this, we would have to compare moderated and unmoderated comments on all of Reddit. To find unmoderated data, we will have to find comments that are too new to have seen a moderator or find an unmoderated subreddit. However, most

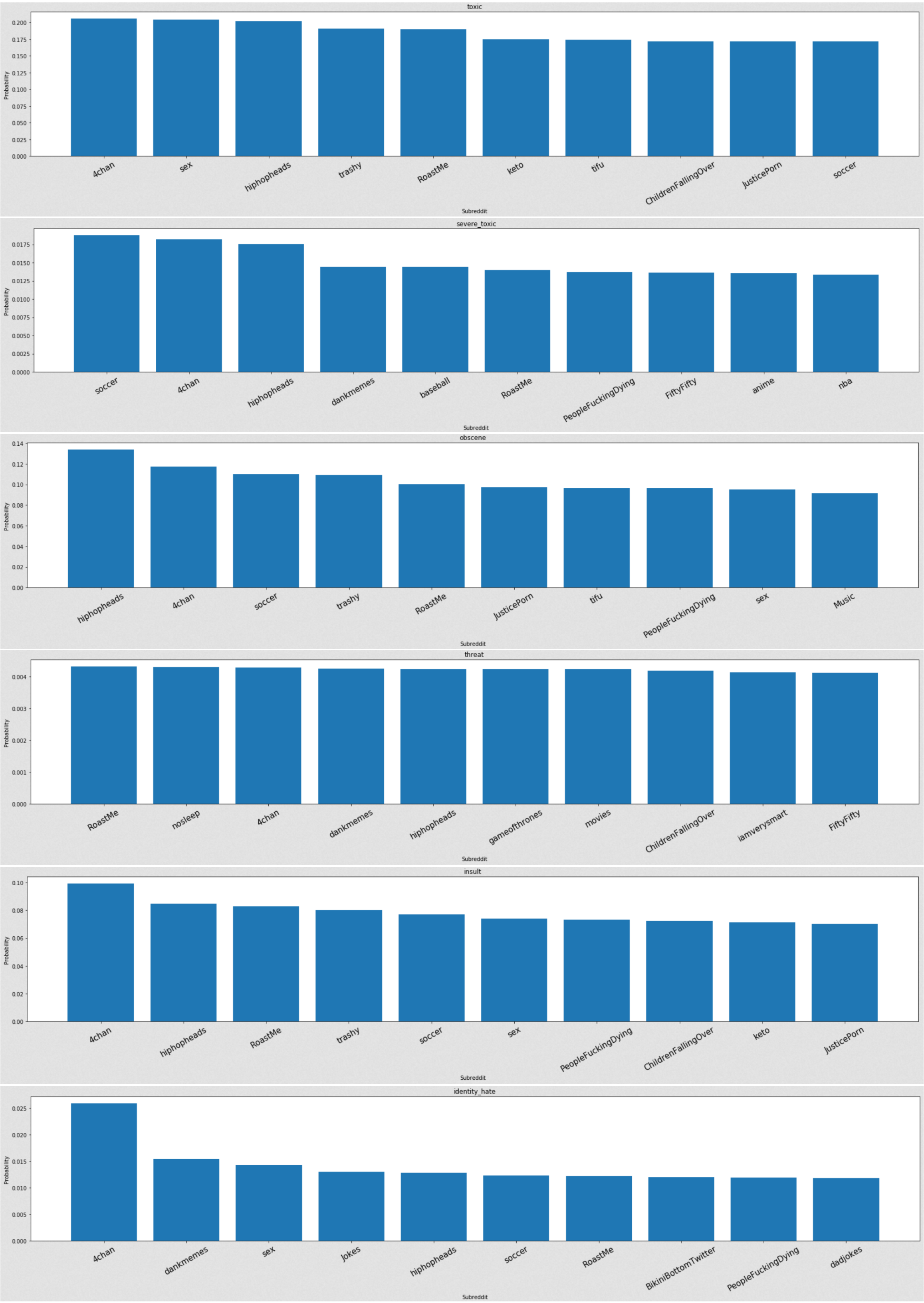


Figure 5.2: Top 10 toxic subreddits in 200 most popular subreddits

subreddits have moderators to enforce at least site-wide rules. New, unmoderated comments can be extracted using the Reddit streaming API to find new comments that are generated at every second. These comments are fresh, and no moderator has had the chance to review them for rule violations yet. Moderated data, on the other hand, can be collected from posts that are usually older than a couple of hours. To be safe, we collected comments from top of r/All from posts that were at least two days old to allow all comments to have been looked through by a moderator.

We collected 100,000 comments from r/All through the Reddit streaming API for unmoderated comments. To ensure uniformity, we collected comments at three different times of the day. Next, to collect moderated comments, we collected 100,000 comments from the top month from r/All excluding posts that were newer than two days.

We measured the toxicity of every comment in these two sets and compared the two sets. We found that moderated comments have an average probability of toxic behaviour to be 13.8% and unmoderated comments have a mean toxicity of 12.2%. The above results show that moderation either is ineffectual at removing toxicity or is really increasing the toxicity by removing less toxic comments.

The results indicate that there may be something wrong with how the moderators of subreddits approach removing comments.

5.3 Pareto principle

The Pareto principle (also commonly known as the 80/20 rule) is a principle that states that around 80% of the effects come from around 20% of the causes. A typical example of the Pareto principle in action is in economics, where the richest 20% of most of the world's countries control 80% or more of the total wealth. The original observation of the Pareto Principle was linked to the relationship between wealth and population. According to what Pareto observed, 80 percent of the land in Italy was owned by 20 percent of the population. After surveying a number of other countries, he found the same applied abroad.

For the most part, the Pareto Principle is an observation that things in life are not always distributed evenly. For instance, the efforts of 20 percent of a corporation's staff could drive 80 percent of the firm's profits. In terms of personal time management, 80 percent of your work-related output could come from only 20 percent of your time at work. In Pareto's case, he used the rule to explain how 80 percent of the wealth is controlled by 20 percent of the country's population.

There is a practical reason for applying the Pareto Principle. Simply, it can give you a window into who to reward or what to fix. For example, if 20 percent of the kinks are leading to 80 percent of the crashes, you can identify and fix those kinks. Similarly, if 20 percent of your customers are driving 80 percent of your sales, you may want to focus on those customers and reward them for their loyalty.

In this case, finding the top 20% most toxic users and taking some significant action against them or monitoring them will give a better understanding of toxic behaviour on your social media site and also vastly reduce the amount of toxic behaviour on your site.

While the 80/20 split is true for Pareto's observation, that doesn't necessarily mean that it always has to equal 100. For instance, 30 percent of the workforce (or 30 out of 100 workers) may only complete 60 percent of the output. The remaining workers may not be as productive or may just slacking off on the job. This further reiterates that the Pareto Principle is merely an observation and not necessarily a law.

Another example would be the observation made by Microsoft where 20% of the most reported bugs were responsible for 80% of errors and crashes in a system. Our idea here was to verify if the Pareto principle would hold when it comes to toxic behaviour from users on Reddit. Therefore, we wanted to measure if 20% of the users are responsible for 80% of the toxicity on the website.

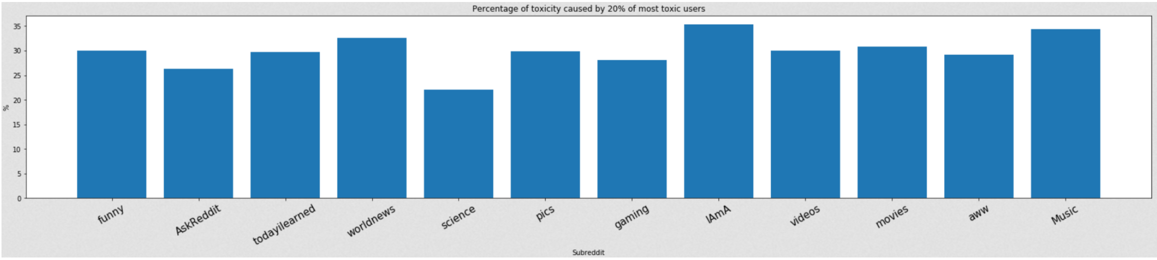


Figure 5.3: Percentage of toxicity caused by the top 20% toxic users

Users who are toxic on one subreddit may not be active on other subreddits or not be toxic on other subreddits. Also, it has been found that users rarely interact frequently in more than one subreddit Buntain and Golbeck [2014]. Therefore, it would make sense to collect comment data from each subreddit separately instead of r/All. We collected 10,000 comments each from the top 10 most popular subreddits and subjected them individually to the test. To remove users who have sporadic interaction with the community or are new accounts, we removed comments from users who have not commented more than once on that subreddit.

Among the top 10 subreddits, we found that the top 20% of the toxic users are only responsible for 25-35% of the total toxicity on the subreddit. The results of our experiment can be found in Figure 5.3 showing the range of amount of toxic behaviour the top 20% toxic users are responsible for. This means that the Pareto principle does not hold for toxicity on Reddit. However, it does appear that there is a significant minority which is responsible for a more substantial proportion of toxic behaviour.

While we cannot suggest anything about the top 20% toxic users of Reddit, we can however guess where else the Pareto principle might apply. The Karma score of an user is the sum of their comment and post upvote and downvote scores. A real world analogy would be wealth or money as users with high Karma get certain privileges like the ability to post and comment in certain subreddits or the ease in becoming a moderator. Therefore, we speculate that the karma score of an user could follow the Pareto distribution. That is, 20% of the users have 80% of the Karma on Reddit. Since, this experiment would have nothing to do with toxic behaviour, we did not verify this.

5.4 Effect of Toxicity on User Activity

We wanted to observe the effect of toxic behaviour on activity in comment threads. The entire comment section can be visualized as a forest of trees. Each tree indicates a comment thread. A new comment can be in response to the post itself or in response to a comment at the root level or in response to any of the sub comments. In other words, every comment thread resembles a tree. Moderators can review comments for rule violations, similar to that of posts, can choose to remove them if they wish to. Toxic comments can start comment fights that deviate the thread from productive and healthy discussion that the platform aims to promote. We can observe the changes caused by toxic comments indirectly by observing the change in the shape

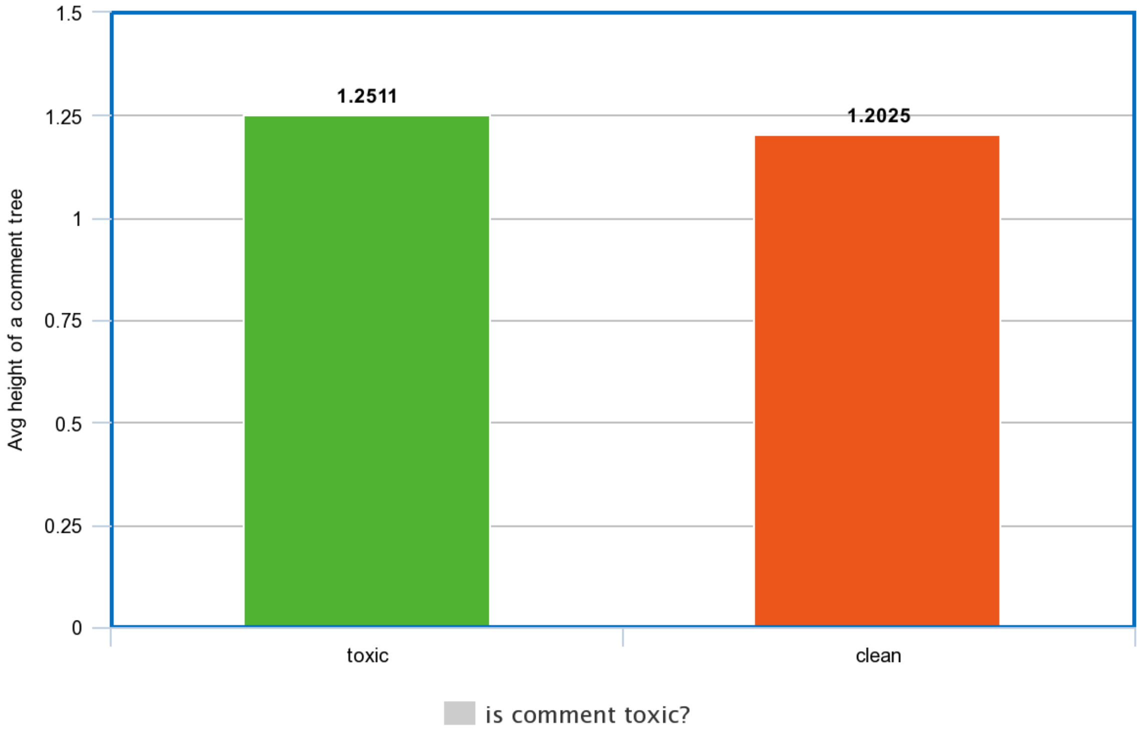


Figure 5.4: Average height of comment trees for clean and toxic top level comments

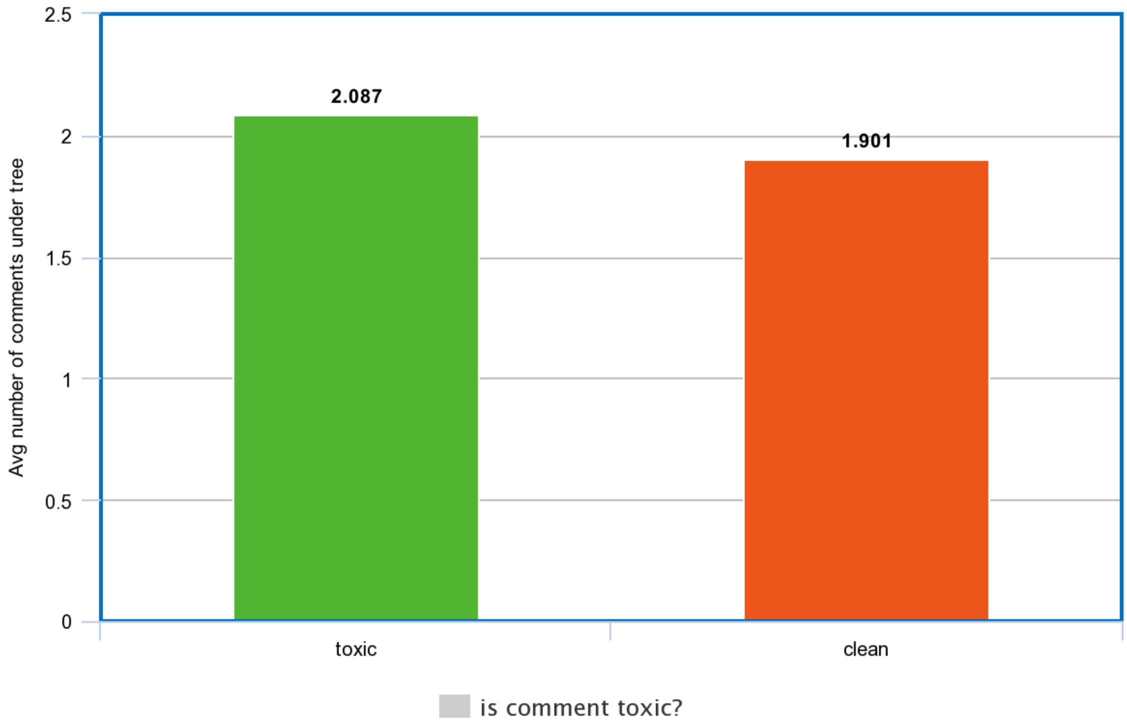


Figure 5.5: Average number of comments under the top level comment in a comment tree

and structure of a comment tree.

First, we collected individual comment threads from the top posts of the r/all subreddit. We calculate the effect on activity by measuring the change in height and the number of sub-comments under toxic and non-toxic top level comments.

For the data collection stage, we need to collect individual comment trees. This is because we need to observe changes in the tree structure of comment threads caused by toxic behaviour. Using the API, we traverse each comment thread in a recursive manner, collecting all comments on a particular level. We store these comments in a tree data structure to match the comment thread structure. This allows for intuitive access of data while processing the data. However, it does slow down access time and requires additional code to support tree operations.

We found that top level comments that are toxic have 9% more sub-comments under them than top level comments that are non-toxic. This can observed in Figure 5.5 Also, the height of toxic level comment trees are 4% larger than non-toxic top level comment trees. This result can be found in Figure 5.4.

This leads us to speculate that top level toxic comments can somewhat deviate discussion by noting the increase in the size of the comment trees. However, we cannot conclude that the ensuing discussion in the below comment tree is non-productive discussion. This is an experiment we would like to conduct in future work.

Chapter 6

Conclusion

We conclude by summarising the questions that this project originally intended to address as well as some other observations made from different experiments.

Most of the activity and interactions across reddit happens over the most popular subreddits across the website and we chose the top 200 subreddits for our toxicity analysis. Because of the varying nature of these subreddits, we naturally expected significantly differing levels of toxicity and our analysis showed the same results. We have shown the results in terms of top 10 most toxic subreddits for all the different classes of toxicity.

The voting system in reddit also opens up the window for us to study the correlation between toxic comments and their upvotes and we used this to find the mean score for toxic and non toxic comments across various subreddits. This score can help identify communities that are generally likely to be more toxic as they tend to have a higher mean score for toxic comments than the other subreddits.

Reddit is a website where most of the content is curated by moderators for the respective subreddits. Their job is to ensure that the community rules and guidelines are adhered to. We conducted an experiment to study the effects of moderation on the overall toxicity of different subreddits and as it turns out, the average probability of unmoderated comments being toxic was slightly lower than that of moderated comments, indicating the possibility of a wrong approach used by moderators in curating the comments. We also found that toxic top level comments do affect the number of comments below them therefore signifying an impact on productive discussion.

Lastly, we wanted to test the Pareto principle (known as the 80/20 rule) with respect to the toxicity on reddit, i.e, if 20% of the users were responsible for 80% of the toxic comments across the website.Our analysis should that this was not the case as the top 20% toxic users were only responsible for about 30% of the toxic comments.

The results of our analysis can be used as the basis for further research into the nature of social networking sites. Toxic behaviour analysis would be useful for assessing the quality of different forums and this can be useful for the administrators to make their websites a more productive and friendlier place and also to draw in more advertisers and promoters.

6.1 Future Work

While we covered a lot of experiments that dealt with the intricacies of Reddit, there is still quite a lot of information to be gained from trying to understand toxic behaviour on Reddit. Analysis of comment threads could be extended to not just top level comments and instead at any location at the comment tree. Also, it would be interesting to know if there is a difference in the number of toxic comments in the comment trees of top level toxic and non-toxic comments.

While we didn't find a Pareto distribution in toxic behaviour across a subreddit, maybe such a distribution exists elsewhere. We speculate it could be the Karma count of a user, as it resembles wealth (and wealth forms a Pareto distribution all over the world), and users gain certain incentives, like money, when they reach a certain Karma count. For example, posting and comment requirements in certain subreddit or the ability to become a moderator.

Our results indicate that moderation is either ineffective at rooting out toxic behaviour or could even be contributing to increasing it. While this appears counter-intuitive, a much deeper analysis can be conducted into just moderation on individual subreddits itself to try observe where this discrepancy arises. Another topic of interest would be to figure out why moderation is ineffective in removing toxic behaviour on Reddit and how moderation can be improved to better tackle this issue.

The top 200 subreddit analysis provides significant insight into the presence of toxic behaviour on the most popular subreddits on the website. This data is useful to both researchers and to the internal Reddit team. To researchers, it provides an avenue to find toxic communities by the topic and further dissect individual subreddits. There is also scope in furthering context based toxicity detection by using data from different subreddits for different context. This can be used hand-in-hand with pseudo-labeling to further improve a detection model. To the internal Reddit team, this provides a lot of data about which subreddits contribute to toxic behaviour on the website and thus require careful monitoring to ensure that the website's bottom-line isn't affected by it. Further, it lets the Reddit team to craft better suggestions to new users when recommending subreddits to them.

Finally, all the experiments can be modified in some form or the other and can be applied on other social media sites like Twitter or Facebook. While most of the experiments don't translate one-to-one, as we found out with Twitter, it should still be possible to convert design similar experiments to applied on these sites.

Bibliography

Cody Buntain and Jennifer Golbeck. Identifying social roles in reddit using network structure. In *Proceedings of the 23rd international conference on world wide web*, pages 615–620. ACM, 2014.

Eshwar Chandrasekharan, Umashanthi Pavalanathan, Anirudh Srinivasan, Adam Glynn, Jacob Eisenstein, and Eric Gilbert. You can't stay here: The efficacy of reddit's 2015 ban examined through hate speech. *Proceedings of the ACM on Human-Computer Interaction*, 1(CSCW):31, 2017.

Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using rnn encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*, 2014.

Junyoung Chung, Çaglar Gülçehre, KyungHyun Cho, and Yoshua Bengio. Empirical evaluation of gated recurrent neural networks on sequence modeling. *CoRR*, abs/1412.3555, 2014. URL `http://arxiv.org/abs/1412.3555`.

Nemanja Djuric, Jing Zhou, Robin Morris, Mihajlo Grbovic, Vladan Radosavljevic, and Narayan Bhamidipati. Hate speech detection with comment embeddings. In *Proceedings of the 24th international conference on world wide web*, pages 29–30. ACM, 2015.

Yves Grandvalet and Yoshua Bengio. Semi-supervised learning by entropy minimization. In *Advances in neural information processing systems*, pages 529–536, 2005.

Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural Comput.*, 9(8):1735–1780, November 1997. ISSN 0899-7667. doi: 10.1162/neco.1997.9.8.1735. URL `http://dx.doi.org/10.1162/neco.1997.9.8.1735`.

Aditya Joshi, Pushpak Bhattacharyya, and Mark J. Carman. Automatic sarcasm detection: A survey. *ACM Comput. Surv.*, 50(5):73:1–73:22, September 2017. ISSN 0360-0300. doi: 10.1145/3124420. URL `http://doi.acm.org/10.1145/3124420`.

Hyunjun Ju and Hwanjo Yu. Sentiment classification with convolutional neural network using multiple word representations. In *Proceedings of the 12th International Conference on Ubiquitous Information Management and Communication*, IMCOM '18, pages 9:1–9:7, New York, NY, USA, 2018. ACM. ISBN 978-1-4503-6385-3. doi: 10.1145/3164541.3164610. URL `http://doi.acm.org/10.1145/3164541.3164610`.

Haewoon Kwak, Jeremy Blackburn, and Seungyeop Han. Exploring cyberbullying and other toxic behavior in team competition online games. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*, pages 3739–3748. ACM, 2015.

Dong-Hyun Lee. Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks. In *Workshop on Challenges in Representation Learning, ICML*, volume 3, page 2, 2013.

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

Tomas Mikolov, Edouard Grave, Piotr Bojanowski, Christian Puhrsch, and Armand Joulin. Advances in pre-training distributed word representations. In *Proceedings of the International Conference on Language Resources and Evaluation (LREC 2018)*, 2018.

Jeffrey Pennington, Richard Socher, and Christopher D. Manning. Glove: Global vectors for word representation. In *Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, 2014. URL `http://www.aclweb.org/anthology/D14-1162`.

Stefan Siersdorfer, Sergiu Chelaru, Wolfgang Nejdl, and Jose San Pedro. How useful are your comments?: analyzing and predicting youtube comments and comment ratings. In *Proceedings of the 19th international conference on World wide web*, pages 891–900. ACM, 2010.

William Warner and Julia Hirschberg. Detecting hate speech on the world wide web. In *Proceedings of the Second Workshop on Language in Social Media*, pages 19–26. Association for Computational Linguistics, 2012.

Tim Weninger, Xihao Avi Zhu, and Jiawei Han. An exploration of discussion threads in social news sites: A case study of the reddit community. In *2013 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM 2013)*, pages 579–583. IEEE, 2013.

Ellery Wulczyn, Nithum Thain, and Lucas Dixon. Ex machina: Personal attacks seen at scale. In *Proceedings of the 26th International Conference on World Wide Web*, pages 1391–1399. International World Wide Web Conferences Steering Committee, 2017.

Zhi Xu and Sencun Zhu. Filtering offensive language in online communities using grammatical relations. In *Proceedings of the Seventh Annual Collaboration, Electronic Messaging, Anti-Abuse and Spam Conference*, pages 1–10, 2010.

Chapter 7

Bio-Data

Aneesh Aithal

Email ID: aneesh297@gmail.com

Contact No.: 8762524146

Adithya S Kamath

Email ID: adithyakamath@gmail.com

Contact No.: 9945699116

Publications:

- "Improved Transfer Learning through Shallow Network Embedding for Classification of Leukemia Cells," *ICAPR 2017*

Harshith Kumar

Email ID: kharshith47@gmail.com

Contact No.: 7760575030