DETECTING ONLINE HATE SPEECH USING BOTH SUPERVISED AND WEAKLY-SUPERVISED APPROACHES

A Thesis

by

LEI GAO

Submitted to the Office of Graduate and Professional Studies of
Texas A&M University
in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE

Chair of Committee,	Ruihong Huang
Committee Members,	Richard Furuta
	Ali Mostafavi
Head of Department,	Dilma Da Silva

May 2018

Major Subject: Computer Science

ABSTRACT

In the wake of a polarizing election, social media is laden with hateful content. Context accompanying a hate speech text is useful for identifying hate speech, which however has been largely overlooked in existing datasets and hate speech detection models. We provide an annotated corpus of hate speech with context information well kept. Then we propose two types of supervised hate speech detection models that incorporate context information, a logistic regression model with context features and a neural network model with learning components for context. Further, to address various limitations of supervised hate speech classification methods including corpus bias and huge cost of annotation, we propose a weakly supervised two-path bootstrapping approach for online hate speech detection by leveraging large-scale unlabeled data. This system significantly outperforms hate speech detection systems that are trained in a supervised manner using manually annotated data. Applying this model on a large quantity of tweets collected before, after, and on election day reveals motivations and patterns of inflammatory language.

DEDICATION

To my parents, my grandfather, and my grandmother.

ACKNOWLEDGMENTS

Foremost, I would like to express my sincere gratitude to my advisor Prof. Ruihong Huang for the continuous support of my thesis study and research, for her patience, motivation, enthusiasm, and immense knowledge. Besides my advisor, I would like to thank the rest of my thesis committee: Prof. Richard Furuta, and Prof. Ali Mostafavi, for their kindness, patience and insightful comments. Last but not the least, I would thank my parents for supporting me in pursuing a Master's degree.

CONTRIBUTORS AND FUNDING SOURCES

Contributors

This work was supported by a thesis committee consisting of Professor Ruihong Huang and Professor Richard Furuta of the Department of Computer Science and Engineering and Professor Ali Mostafavi of the Department of Civil Engineering.

The tweets collected for Chapter 3 was provided by former TAMU grad student Girish Kasiviswanathan. The annotation of tweets and online comments in Chapter 2 and Chapter 3 were provided by Lexi Kuppersmith from Stanford University.

All other work conducted for the thesis was completed by the student independently.

Funding Sources

This thesis research was conducted without financial support.

NOMENCLATURE

OGAPS	Office of Graduate and Professional Studies at Texas A&M University
TAMU	Texas A&M University
LSTM	Long Short Term Memory Recurrent Neural Network
CNN	Convolutional Neural Network
LIWC	Linguistic Inquiry and Word Count

TABLE OF CONTENTS

Page

LIST OF FIGURES

FIGURE Page

LIST OF TABLES

TABLE Page

1. INTRODUCTION AND LITERATURE REVIEW*

1.1 Introduction

Following a turbulent election season, 2016's cyber world is awash with hate speech. Automatic detection of hate speech has become an urgent need since human supervision is unable to deal with large quantities of emerging texts. Apart from censorship, the goals of enabling computers to understand inflammatory language are many. Sensing increased proliferation of hate speech can elucidate public opinion surrounding polarizing events. Identifying hateful declarations can bolster security in revealing individuals harboring malicious intentions towards specific groups. Context information, by our definition, is the text, symbols or any other kind of information related to the original text. For instance, the contextual information of a tweet can be tweets it replied to, its hashtags or even the trending topic when the tweet was posted. As to an online news comment, the contextual information can be the other comments in the same discussion thread, the news article the comment was written for, etc. While intuitively, context accompanying hate speech is useful for detecting hate speech, context information of hate speech has been overlooked in existing datasets and automatic detection models.

Online hate speech tends to be subtle and creative, which makes context especially important for automatic hate speech detection. For instance,

(1) *barryswallows: Merkel would never say NO*

This comment is posted for the news titled by "German lawmakers approve 'no means no' rape law after Cologne assaults". With context, it becomes clear that this comment is a vicious insult towards female politician. For another example, Here "say No" means saying no to those attackers.

(2) *ozart: @endpcinamerica What do you mean evolved? They stillare primates....*

*Part of this chapter is reprinted with permission from "Recognizing Explicit and Implicit Hate Speech Using a Weakly Supervised Two-path Bootstrapping Approach" by Lei Gao, Alexis Kuppersmith, Ruihong Huang, The 8th International Joint Conference on Natural Language Processing 2017, Copyright 2018 by IJCNLP 2017, and "Detecting Online Hate Speech Using Context Aware Models" by Lei Gao, Ruihong Huang, Recent Advances in Natural Language Processing 2017. Copyright 2018 by RANLP 2017.

The title of the news article that this comment was written for is "Fury as feminist blames toddler alligator death on white entitlement". The discussion of the thread containing this comment is concentrated on "feminists". By looking at the context, we understand that this user uses "they" in lieu of "feminists" and describe them as "primates", which makes the comment hateful. However, almost all the publicly available hate speech annotated datasets do not contain context information [1, 2, 3, 4].

We have created a new dataset consisting of 1,528 Fox News user comments, which were taken from 10 complete discussion threads for 10 widely read Fox News articles. It is different from previous datasets from the following two perspectives. First, it preserves rich context information for each comment, including its user screen name, all comments in the same thread and the news article the comment is written for. Second, there is no biased manual data selection and all comments in each news comment thread were annotated. The comments were ordered temporal order between these comments are kept as origin. We observed that, apart from those explicit hate speech which uses a slur or a derogatory term, there are some implicit hate speech which conveys its hatefulness by the compositional meaning of that sentence. Here we provide 2 examples of implicit hate speech in this corpus. The following is one example of implicit hate speech comments. It shows creativeness in creating hate speech by using "abortion" as a way of insult.

(3) *VivianLee: @fatboyhd2k I think the parents had a failed abortion–only it's brain was aborted–the body stayed. That's why it's the way it is.*

(4)*Wyerd: American feminists are about as useful as soundless Velcro. Grew up with daddy issues no doubt.*

This comment uses metaphor to degrade feminists by describing them as "soundless Velcro". Those metaphorical hateful language is considered implicit hate speech by our definition

After preparing corpus, we start to think about the suitable model for hate speech detection with contextual information. However, current models of hate speech detection does not take contextual information into account. In this thesis, we plan to explore two types of models, feature based

logistic regression models and neural network models, in order to incorporate context information in automatic hate speech detection.

After we built the model for the relatively small Fox News Comments corpus, we start to think about working on larger corpus like Tweets. We first carried out a pilot annotation of 5,000 randomly selected tweets. We find that there are some implicit hate speech in tweets as well. For example,

(5) *Hillary's welfare army doesn't really want jobs. They want more freebies.*

(6) *Affirmative action means we get affirmatively second rate doctors and other professionals.*

The annotation results of our pilot annotation showed that around 0.6% (31 tweets) of tweets are hateful, which means online hate speech on Twitter relatively infrequent (among large amounts of online contents). Due to the sparsity, annotating a hateful tweets corpus is extremely expensive. The annotation effort inspires us to think about the current available corpus and the drawback of supervised approach. We find that recent studies on supervised methods for online hate speech detection [1, 5] have relied on manually annotated data sets, which are not only costly to create but also likely to be insufficient to obtain wide-coverage hate speech detection systems. This is mainly because online hate speech is and tends to transform rapidly following a new trigger event. The mass-scale (Yahoo!Finance online comments) hate speech annotation effort from Yahoo! [5] revealed that only 5.9% of online comments contained hate speech, which means the sparsity of hateful speech is not only a problem in tweets but also for online comments. In recent studies [1, 6], the data selection methods and annotations are often biased towards a specific type of hate speech or hate speech generated in certain scenarios in order to increase the ratio of hate speech content in the annotated data sets, which however made the resulting annotations too distorted to reflect the true distribution of hate speech. Though we can apply filters when selecting data to annotate in order to increase the portion of hate speech content, the resulting annotations will not reflect the true distribution of hate speech. For instance, in [1], around 16,000 tweets that contain a pre-defined slur term or hate related topic keywords were annotated, roughly a third of these tweets

contain hate content. Furthermore, inflammatory language changes dramatically follows new hate "trigger" events, which will significantly devalue annotated data.

To address the various limitations of supervised hate speech detection methods, we plan to present a weakly supervised two-path bootstrapping approach for online hate speech detection that requires minimal human supervision and can be easily retrained and adapted to capture new types of inflammatory language. Our two-path bootstrapping architecture consists of two learning components, an explicit slur term learner and a neural net classifier (LSTMs [7]), that can capture both explicit and implicit phrasings of online hate speech. Specifically, our bootstrapping system starts with automatically labeled online hateful content that are identified by matching a large collection of unlabeled online content with several hateful slur terms. Then two learning components will be initiated simultaneously. A slur term learner will learn additional hateful slur terms from the automatically identified hateful content. Meanwhile, a neural net classifier will be trained using the automatically labeled hateful content as positive instances and randomly sampled online content as negative instances. Next, both string matching with the newly learned slur terms and the trained neural net classifier will be used to recognize new hateful content from the large unlabeled collection of online contents. Then the newly identified hateful content by each of the two learning components will be used to augment the initially identified hateful content, which will be used to learn more slur terms and retrain the classifier. The whole process iterates until stopping condition is met.

1.2 Motivation and Observations

In this section, we introduce the motivation of this research, identify the research challenges, and present the state of the art.

1.2.1 Bias in building corpus

Recently, a few datasets with human labeled hate speech have been created, however, most of existing datasets do not contain context information. Due to the sparsity of hate speech in everyday posts, researchers tend to sample candidates from bootstrapping instead of random sampling, in

order to increase the chance of seeing hate speech. Therefore, the collected data instances are likely to be from distinct contexts.

For instance, in the Primary Data Set described in [8] and later used by [5], 10% of the dataset is randomly selected while the remaining consists of comments tagged by users and editors. Later on, [9, 8, 5] have worked on corpus collected on Yahoo. Specifically, the Primary Data Set is composed of 10% randomly selected comments and other "reported" comments from users and editors. It is unknown whether contextual information is kept within this corpus.

[6] built a balanced data set of 24.5k tweets by selecting from Twitter accounts that claimed to be racist or were deemed racist using their followed news sources. [10] collected hateful tweets related to the murder of Drummer Lee Rigby in 2013. [1] provided a corpus of 16k annotated tweets in which 3.3k are labeled as sexist and 1.9k are labeled as racist. They created this corpus by bootstrapping from certain key words ,specific hashtags and certain prolific users. [11] created a dataset of 9000 human labeled paragraphs that were collected using regular expression matching in order to find hate speech targeting Judaism and Israel. [12] extracted data instances from instagram that were associated with certain user accounts. They first sampled a large amount of users, then used profanity words to filter and find users who have a high potential to post hate speech. [3] presented a very large corpus containing over 115k wikipedia comments that include around 37k randomly sampled comments and the remaining 78k comments were selected from wikipedia blocked comments.Contextual information is not included in this corpus.

1.2.2 Lack of contextual knowledge

Most of existing hate speech detection models are feature based and use features derived from the target text itself. [10] experimented with different classification methods including Bayesian Logistic Regression, Random Forest Decision Trees and SVMs, using features such as n-grams, reduced n-grams, dependency paths, and hateful terms which improves n-gram baseline of 3% compared to baseline n-gram SVM model. [1] proposed a logistic regression model using character n-gram features. The evaluation shows using gender as feature gives an improvement of 0.04% F1 score. [8] used the paragraph2vec for joint modeling of comments and words, then the generated

embeddings were used as feature in a logistic regression model.They improved the AUC score of about 1% compared to bag of words feature. [5] experimented with various syntactic, linguistic and distributional semantic features including word length, sentence length, part of speech tags, and embedding features, in order to improve performance of logistic regression classifiers. They also proposed a comment2vec model in which every comment is mapped into a unique vector in a matrix representing comments and every word is mapped into a matrix representing words. The comment vector and word vector are concatenated to predict the next word in a context. The authors combined those features and feed them to a logistic regression classifier. The evaluations on WWW2015 corpus shows that all features combined model successfully improved F1 score of 0.9% and AUC score of 0.2% compared to character level feature baseline model. Recently, [13] surveyed current approaches for hate speech detection, and raised a concern on the lack of a benchmark data set for hate speech detection. More importantly, it also interestingly called to attention on modeling context information for resolving difficult hate speech instances. Lastly, it is worth mentioning that [13] made a survey of hate speech detection using NLP technique. They well summarized current approaches for hate speech detection and raises concern on the lack of a benchmark data set for hate speech detection. More importantly, they also called for attention on contextual knowledge for some difficult instances of hate speech.

1.2.3 Lack of distant supervision

Previous studies on hate speech recognition mostly used supervised approaches. Due to the sparsity of hate speech overall in reality, the data selection methods and annotations are often biased towards a specific type of hate speech or hate speech generated in certain scenarios. For instance, [14] conducted their experiments on 1525 annotated sentences from a company's log file and a certain newsgroup. [11] labeled around 9000 human labeled paragraphs from Yahoo!'s news group post and American Jewish Congress's website, and the labeling is restricted to anti-Semitic hate speech. [9] studied use of profanity on a dataset of 6,500 labeled comments from Yahoo! Buzz. [6] built a balanced corpus of 24582 tweets consisting of anti-black and non-anti black tweets. The tweets were manually selected from Twitter accounts that were believed to be

racist based upon their reactions to anti-Obama articles. [10] collected hateful tweets related to the murder of Drummer Lee Rigby in 2013. [1] collected tweets using hateful slurs, specific hashtags as well as suspicious user IDs. Consequently, all of the 1,972 racist tweets are by 9 users, and the majority of sexist tweets are related to an Australian TV show.

[8] is the first to study hate speech using a large-scale annotated data set. They have annotated 951,736 online comments from Yahoo!Finance, with 56,280 comments labeled as hateful. [5] followed [8]'s work. In addition to the Yahoo!Finance annotated comments, they also annotated 1,390,774 comments from Yahoo!News. Comments in both data sets were randomly sampled from their corresponding websites with a focus on comments by users who were reported to have posted hateful comments. We instead aim to detect hate speech w.r.t. its real distribution, using a weakly supervised method that does not rely on large amounts of annotations.

The commonly used classification methods in previous studies are logistic regression and Naive Bayes classifiers. [8] and [5] applied neural network models for training word embeddings, which were further used as features in a logistic regression model for classification. We will instead train a neural net classifier [15, 16, 17] in a weakly supervised manner in order to capture implicit and compositional hate speech expressions.

To the best of our knowledge, [18] is one of the few works which did use approaches other than supervised models in detecting hate speech. They used a bootstrapping method to discover offensive language from a large-scale Twitter corpus. However, their bootstrapping model is driven by mining hateful Twitter users, instead of content analysis of tweets as in our approach. Furthermore, they recognize hateful Twitter users by detecting explicit hateful indicators (i.e., keywords) in their tweets while our bootstrapping system aim to detect both explicit and implicit expressions of online hate speech.

1.3 Proposed Solution

In this section, we propose our solution for the problems of hate speech detection in both Fox News Comments and tweets.

1.3.1 Context-aware Online Hate Speech Detection Models

1.3.1.1 Logistic Regression Models

In logistic regression models, we extract four types of features, word-level and character-level n-gram features as well as two types of lexicon derived features. We extract these four types of features from the target comment first. Then we extract these features from two sources of context texts, specifically the title of the news article that the comment was posted for and the screen name of the user who posted the comment.

For logistic regression model implementation, we use l2 loss. We adopt the balanced class weight as described in Scikit learn [19]. Logistic regression model with character-level n-gram features is presented as a strong baseline for comparison since it was shown very effective [1, 5].

- Word-level and Character-level N-gram Features. For character level n-grams, we extract character level bigrams, tri-grams and four-grams. For word level n-grams, we extract uni-grams and bigrams.

- LIWC Feature. Linguistic Inquiry and Word Count, also called LIWC, has been proven useful for text analysis and classification [20]. In the LIWC dictionary, each word is labeled with several semantic labels. In our experiment, we use the LIWC 2015 dictionary which contain 125 semantic categories. Each word is converted into a 125 dimension LIWC vector, one dimension per semantic category. The LIWC feature vector for a comment or its context is a 125 dimension vector as well, which is the sum of all its words' LIWC vectors.

- NRC Emotion Lexicon Feature. NRC emotion lexicon contains a list of English words that were labeled with eight basic emotions (anger, fear, anticipation, trust, surprise, sadness, joy, and disgust) and sentiment polarities (negative and positive) [21]. We use NRC emotion lexicon to capture emotion clues in text. Each word is converted into a 10 dimension emotion vector, corresponding to eight emotion types and two polarity labels. The emotion vector for a comment or its context is a 10 dimension vector as well, which is the sum of all its words' emotion vectors.

1.3.1.2 Neural Network Models

Our neural network model mainly consists of three parallel LSTM [7] layers. It has three different inputs, including the target comment, its news title and its username. Comment and news title are encoded into a sequence of word embeddings. We use pre-trained word embeddings in word2vec [22]. Username is encoded into a sequence of characters. We use one-hot encoding of characters.

Comment is sent into a bi-directional LSTM with attention mechanism [23]. News title and username are sent into a bi-directional LSTM. Note that we did not apply attention mechanism to the neural network models for username and news title because both types of context are relatively short and attention mechanism tends to be useful when text input is long. The three LSTM output layers are concatenated, then connected to a sigmoid layer, which outputs predictions. Note that the three inputs are not always necessary. In our experiments, we will also evaluate the model with some inputs disabled.

The number of hidden units in each LSTM used in our model is set to be 100. The recurrent dropout rate of LSTMs is set to 0.2. In addition, we use binary cross entropy as the loss function and a batch size of 128. The neural network models are trained for 30 epochs.

1.3.1.3 Ensemble Models

As described above, the logistic regression model and neural network model are performing differently in identifying hate speech. To study the difference of logistic regression model and neural network model and potentially get performance improvement, we will build and evaluate ensemble models.

1.3.2 The Two-path Bootstrapping System for Online Hate Speech Detection

1.3.3 Overview

Figure 1.1 illustrates that our weakly supervised hate speech detection system starts with a few pre-identified slur terms as seeds and a large collection of unlabeled data instances. Specifically, we experiment with identifying hate speech from tweets. Hateful tweets will be automatically

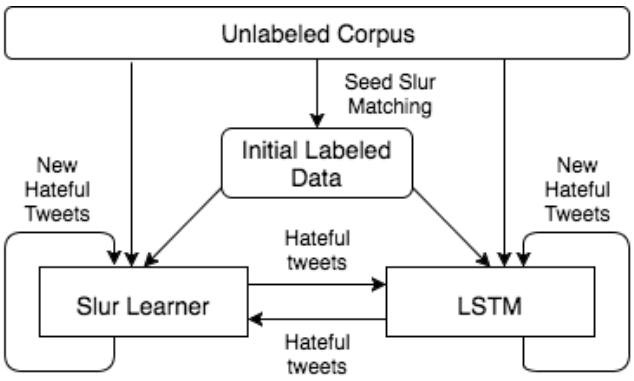

Figure 1.1: Diagram of Co-training Model

identified by matching the large collection of unlabeled tweets with slur term seeds. Tweets that contain one of the seed slur terms are labeled as hateful.

The two-path bootstrapping system consists of two learning components, an explicit slur term learner and a neural net classifier (LSTMs [7]), that can capture both explicit and implicit descriptions of online hate speech. Using the initial seed slur term labeled hateful tweets, the two learning components will be initiated simultaneously. The slur term learner will continue to learn additional hateful slur terms. Meanwhile, the neural net classifier will be trained using the automatically labeled hateful tweets as positive instances and randomly sampled tweets as negative instances. Next, both the newly learned slur terms and the trained neural net classifier will be used to identify new hateful content from the unlabeled large collection of tweets. The newly labeled hateful tweets by each of the two learning components will be used to augment the initial slur term seed identified hateful tweet collection, which will be used to learn more slur terms and retrain the classifier in the next iteration. The whole process then iterates.

After each iteration, we have to determine if a stopping criterion is met and we should terminate the bootstrapping process. In general, a tuned threshold score is applied or a small annotated dataset is used to evaluate the learned classifiers.

1.4 Related Works About Hate Speech Detection

In this section, we summarize the works related to our thesis project.

1.4.1 Automatic Hate Speech Detection

[14] gives a detailed definition of offensive language. Enumerating all possible classifiers, they built a three-level classifier. After evaluating the classifier with cross validation on 1.5k annotated online messages, they achieved a precision of 97% and recall of 100%.

[18] first used a bootstrapping method to discover offensive language in a large-scale Twitter corpus. The authors use 200 seed words to initialize bootstrapping and bootstrapped more tweets from identifying hateful Twitter users. By implementing topical and lexical features, they trained a logistic classifier on their bootstrapped data and evaluated the true positive rate on 4k randomly sampled tweets. They achieved a true positive score of 75%.

The cyber security community and data mining community have worked on a similar topic: cyberbullying detection. One of the main focuses of cyberbullying detection is offensive language detection. Generally speaking, the definition of offensive language overlaps with that of hate speech. [24] takes a supervised classification approach using SVM with features of n-grams, manually designed regular expressions, dependency parsing features and automatically derived blacklists. They achieved a precision of 98% and a recall of 94% on their data set.

[11] discussed the definition of hate speech and formulated the hate speech detection problem as a word sense disambiguation problem. They believe that if a word has a stereotype sense, then the sense of all words can be determined from paragraph labels. In the experiment, they worked on 9000 human labeled paragraphs. They used a template based feature extractor to generate features and to feed them to a SVM classifier. Their approach achieved 68% precision and 60% recall with best choice of features which they claim is competitive against human annotators.

[9] focused on use of profanity. They investigate profanity usage in Yahoo! Buzz communities and found that different communities use profanity in different ways or in different contexts at different frequencies.

[6] focused on detection of racist hate speech. They built a balanced data set of 24.5k tweets by selecting from Twitter accounts that claimed to be racist or were deemed racist though followed news sources. They use a bag of word model as a feature and perform cross validated on Naive

Bayes classifier, achieving an accuracy of 76%.

[10] collected hateful tweets related to the murder of Drummer Lee Rigby in 2013 and applied two different classifier to classify hateful tweets: a Bayesian Logistic Regression classifier and a Random Forest Decision Tree. By incorporating features of reduced n-gram, dependency parsing, and hateful terms, the combined classifiers achieved an F score of 0.95. Their corpus consisted of 1901 tweets, 222 of which were hateful.

[8] first applied the neural network model in hate speech classification on annotated online news comments. They used the paragraph2vec for joint modeling of comments and words, then they used the trained embedding to feed a logistic regression classifier. They achieved a AUC score of 0.80 on the Yahoo Finance comments data set.

[5] followed Djuric et.al 's approach on hate speech classification. The authors implemented many features including linguistic features of word length and sentence length, syntactic features of part of speech tagging, and distributional semantics features. They proposed a comment2vec model in which every comment is mapped into a unique vector in a matrix representing comments and every word is mapped into a matrix representing words. The comment vector and word vector are concatenated to predict the next word in a context. The authors combined those features and feed them to a logistic regression classifier. The evaluations on different corpus shows that this approach has very good performance of about 79% precision and 81% recall. [1] provided a corpus of 16k annotated tweets in which 3.3k are sexist and 1.9k are racist. They first collected tweets which contained hateful slurs, then they identified hashtags and users highly related to hate speech and found more tweets with the same hashtags and users. The authors trained a logistic classifier using character n-gram features and achieved a precision of 73% and a recall of 78%. They also found that the gender of a user can be a good indicator of hatefulness.

1.4.2 Supervised Neural Network Models For Text Classification

Hate speech detection is within the definition of text classification. Recently, neural network models have been widely applied in text classification. Here we introduce related works using neural network models in text classification.

[25] proposed a character level convolutional networks model called ConvNets for text classification. They explore treating text as a sequence of raw signal at character level, and applying one-dimensional ConvNets to it. They are the first to apply ConvNets only on characters. The experiments in their paper shows that when trained on largescale datasets, deep ConvNets do not require the knowledge of words, which is one step further compared to the conclusion in their previous research that their approach do not require the knowledge about the syntactic or semantic structure of a language.

[16] introduced a recurrent convolutional neural network model (R-CNN) for text classification without handcrafted features. They apply a recurrent structure to capture contextual information as far as possible when learning word embeddings. The main idea is to mimimal the noise when learning those representations compared to traditional window-based neural networks. A max-pooling layer is applied to automatically judges which words play key roles in text classification to capture the key components in texts.

[26] proposed a hierarchical attention network(HAN) for document classification. One of the two major contributions of their model is that it has a hierarchical structure that mirrors the hierarchical structure of documents. It also applied two levels of attention mechanisms at both sword and sentence-level in order to attend differentially to more and less important content when constructing the document representation.

[27] proposed a very deep convolutional network for text classification named VD-CNN which consists of 29 layers of CNN. The CNNs are fully connected with max pooling layers in every two CNN layers. Short cut connections are enabled between every other layer. The experiments shows that their model performed better than some of the state-of-the-art models in text classification.

1.4.3 Semi-Supervised and Weakly-Supervised Models For Text Classification

[28] applied Latent Dirichlet Allocation (LDA) based topic model on unlabeled large scale data and used LDA predicted labels as a source of supervision on text classification tasks. The LDA model is an unsupervised probabilistic generative model for collections of discrete data. They encoded the predicted topic to training documents using a technique called sprinkling.

[29] constructed hinge-loss Markov random fields(HL-MRFs) for weakly supervised tweet classification. All tweet posts, users, and their relationships are used to construct the graph of HL-MRFs. By applying this relational approach, they went beyond the stance-indicative patterns and found more stance-indicative tweets.

[30] exploited unlabeled data as an additional resource by using LSTM and CNN in a two-view embedding learning. The LSTM and CNN trained word embedding serves as the feature embedding which encodes topic level information to potentially help the supervised LSTM text classification model performs better.

[31] proposed a semi-supervised model for text classification using deep adversarial neural network. Their approach mainly applied the adversarial and virtual adversarial in training time to enable perturbations to the word embeddings in a recurrent neural network as a way to enrich training instances.

1.4.4 Other Techniques and Resources

Word embedding, as proposed in [22], is a technique that maps words or phrases from the vocabulary to a vector of real numbers. It is essentially a mathematical embedding from a space with one dimension per word to a continous space with low dimension that words with similar meaning have closer distance. Word embedding has been widely used in natural language processing in almost all kinds of tasks and applications.

Linguistic Inquiry and Word Count(LIWC), as introduced in [32], is a linguistic dictionary composed of 5,690 words and word stems created using computerized text analysis methods. In LIWC, each word or word stem defines one or more word categories. Those word categories are usually arranged hierarchically. For example, the word 'cried' is part of four word categories: sadness, negative emotion, overall affect, and a past tense verb. Hence, 'cried' will fill into each of those four category and the category scale scores will be calculated.

2. SUPERVISED MODEL FOR HATE SPEECH DETECTION*

2.1 The Fox News User Comments Corpus

2.1.1 Corpus Overview

The Fox News User Comments corpus consists of 1528 annotated comments (435 labeled as hateful) that were posted by 678 different users in 10 complete news discussion threads in the Fox News website. The 10 threads were manually selected and represent popular discussion threads during August 2016. All of the comments included in these 10 threads were annotated. The number of comments in each of the 10 threads is roughly equal. Rich context information was kept for each comment, including its user screen name, the comments and their nested structure and the original news article. The data corpus along with annotation guidelines is posted on github[‡].

2.1.2 Annotation Guidelines

Our annotation guidelines are similar to the guidelines used by [5]. We define hateful speech to be the language which explicitly or implicitly threatens or demeans a person or a group based upon a facet of their identity such as gender, ethnicity, or sexual orientation. Specifically, employs stereotypes, exemplifies xenophobia/hateful patriotism, makes a threat towards a specific person/group, bad screen names, using "They" in lieu of identifying group, minority identity implied through context, implies or states inferiority or uncivillized nature, generalizing about a group, Othering, characterizes groups as crazy or irrational, expresses fear of a power grab by women or minorities, dehumanization, infantilization can be all considered hateful. Detailed explanation with examples are shown in Appendix. The labeling of hateful speech in our corpus is binary. A comment will be labeled as hateful or non-hateful.

*Part of this chapter is reprinted with permission from "Detecting Online Hate Speech Using Context Aware Models" by Lei Gao, Ruihong Huang, Recent Advances in Natural Language Processing 2017. Copyright 2018 by RANLP 2017.

‡https://github.com/sjtuprog/fox-news-comments

2.1.3 Annotation Procedure

We identified two native English speakers for annotating online user comments. The two annotators first discussed and practices before they started annotation. We first asked the two annotators to thoroughly discuss over the annotation guidelines and conduct practice annotations for several iterations. Then in order to measure inter-agreements, we asked to annotate comments from four discussion threads that contain 648 comments in total. They achieved a surprisingly high Kappa score [33] of 0.98 on 648 comments from 4 threads. The two annotators are both female. They are both high school students with good knowledge of US politics.We think that thorough discussions in the training stage is the key for achieving this high inter-agreement. For those comments which annotators disagreed on, we label them as hateful as long as one annotator labeled them as hateful. Then one annotator continued to annotate the remaining 880 comments from the remaining six discussion threads. We are aware of the annotation bias of annotators as described in [2]. However, comparision of annotations is not considered in this thesis since more annotators are not available.

2.1.4 Characteristics in Fox News User Comments corpus

Hateful comments in the Fox News User Comments Corpus is often subtle, creative and implicit. Therefore, context information is necessary in order to accurately identify such hate speech.

2.1.4.1 Context Dependent Comments

The hatefulness of many comments depended on understanding their contexts. For instance,

(1) *mastersundholm: Just remember no trabjo no cervesa*

This comment is posted for the news "States moving to restore work requirements for food stamp recipients". This comment implies that Latino immigrants abuse the usage of food stamp policy, which is clearly a stereotyping.

(2)*jmgandolfo: @SFgunrmn ALL of them are.*

This comment is proceeded by the comment “what a efen loon, but most femanazis are.”. Because the previous comment expresses hate against feminists and this comment supports it,

therefore this comment is also hateful.

2.1.4.2 Implicit and creative language

Many hateful comments use implicit and subtle language, which contain no clear hate indicating word or phrase. In order to recognize such hard cases, we hypothesize that neural net models are more suitable by capturing overall composite meanings of a comment. For instance, the following comment is a typical implicit stereotyping against women.

(3) *MarineAssassin: Hey Brianne - get in the kitchen and make me a samich. Chop Chop*

This comment is a typical implicit stereotyping against women. It describes women as householders and shows no respect.

(4) *PostApocalypticHero: Expect more and more women to be asking .. "why are men no longer interested in me"! We're not going touch you until you pull our pants down!*

This comment contains an implicit insult to women.

2.1.4.3 Long Comments with Regional Focus of hatefulness

11% of our annotated comments have more than 50 words each. In such long comments, the hateful indicators usually appear in a small region of a comment while the majority of the comment is neutral. For example,

(5) *TMmckay: I thought ...115 words...* ***Too many blacks winning, must be racist and needs affirmative action to make whites equally win!***

In this comment, almost only the bold last sentence contain hate indicators.

2.1.4.4 Disrespectful screen names

Certain user screen names indicate hatefulness, which imply that comments posted by these users are likely to contain hate speech. In the following example, commie is a slur for communists.

(6)*nocommie11: Blah blah blah. Israel is the only civilized nation in the region to keep the unwashed masses at bay.*

This screen name indicates that this user is likely to post hateful comments.

(7)*LibtardTroller: Good. If I have to work for money and food then so should the leeches of society. Food stamps should really be eliminated totally. Then you'll see unemployment truly drop. If you have kids you can't afford, have no family or friends for help, and need the government to survive, you made poor choices and deserve to live with the consequences.*

Libtard is a derogatory term for liberals. Troller is a self-identification of being malicious.

2.2 Context-aware Online Hate Speech Detection Models

We explore both feature based logistic regression models and neural net models in order to incorporate context information for hate speech detection.

2.2.1 Logistic Regression Models

In logistic regression models, we extract four types of features, word-level and character-level n-gram features as well as two types of lexicon derived features. We extract these four types of features from the target comment first. Then we extract these features from two sources of context texts, specifically the title of the news article that the comment was posted for and the screen name of the user who posted the comment.

For logistic regression model implementation, we use l2 loss. We adopt the balanced class weight as described in Scikit learn [19]. Logistic regression model with character-level n-gram features is presented as a strong baseline for comparison since it was shown very effective [1, 5].

2.2.1.1 Word-level and Character-level N-gram Features

For character level n-grams, we extract character level bigrams, tri-grams and four-grams. For word level n-grams, we extract unigrams and bigrams.

2.2.1.2 LIWC Feature

Linguistic Inquiry and Word Count, also called LIWC, has been proven useful for text analysis and classification [20]. In the LIWC dictionary, each word is labeled with several semantic labels. In our experiment, we use the LIWC 2015 dictionary which contain 125 semantic categories. Each word is converted into a 125 dimension LIWC vector, one dimension per semantic category. The

LIWC feature vector for a comment or its context is a 125 dimension vector as well, which is the sum of all its words' LIWC vectors.

2.2.1.3 NRC Emotion Lexicon Feature

NRC emotion lexicon contains a list of English words that were labeled with eight basic emotions (anger, fear, anticipation, trust, surprise, sadness, joy, and disgust) and sentiment polarities (negative and positive) [21]. We use NRC emotion lexicon to capture emotion clues in text. Each word is converted into a 10 dimension emotion vector, corresponding to eight emotion types and two polarity labels. The emotion vector for a comment or its context is a 10 dimension vector as well, which is the sum of all its words' emotion vectors.

2.2.2 Neural Network Models

Our neural network model mainly consists of three parallel LSTM [7] layers. It has three different inputs, including the target comment, its news title and its username. Comment and news title are encoded into a sequence of word embeddings. We use pre-trained word embeddings in word2vec[22]. Username is encoded into a sequence of characters. We use one-hot encoding of characters.

Comment is sent into a bi-directional LSTM with attention mechanism [23]. News title and username are sent into a bi-directional LSTM. Note that we did not apply attention mechanism to the neural network models for username and news title because both types of context are relatively short and attention mechanism tends to be useful when text input is long. The three LSTM output layers are concatenated, then connected to a sigmoid layer, which outputs predictions.

The number of hidden units in each LSTM used in our model is set to be 100. The recurrent dropout rate of LSTMs is set to 0.2. In addition, we use binary cross entropy as the loss function and a batch size of 128. The neural network models are trained for 30 epochs.

2.2.3 Ensemble Models

To study the difference of logistic regression model and neural network model and potentially get performance improvement, we will build and evaluate ensemble models.

We evaluate our model by 10 fold cross validation using our newly created Fox News User Comments Corpus. Both types of models use the exact same 10 folds of training data and test data. We report experimental results using multiple metrics, including accuracy, precision/recall/F1-score, and accuracy area under curve (AUC).

2.2.4 Experimental Results

2.2.4.1 Logistic Regression Models

Features	**Input Contents**	**Accuracy**	**Precision**	**Recall**	**F1**	**AUC**
char (baseline)	comment	0.738	0.549	0.469	0.504	0.733
+word	comment	0.735	0.548	0.443	0.488	0.736
+LIWC+NRC	comment	0.732	0.533	0.465	0.495	0.740
+word+LIWC+NRC	comment	**0.747**	**0.568**	**0.476**	**0.517**	**0.750**
	+ username	0.747	**0.576**	0.474	0.518	0.765
	+ title	0.745	0.558	0.496	0.523	0.761
	+ title+ username **(Best)**	**0.750**	0.572	**0.516**	**0.542**	**0.778**

Table 2.1: Performance of Logistic Regression Models

Table 2.1 shows the performance of logistic regression models. The first section of table 2.1 shows the performance of logistic regression models using features extracted from a target comment only. The result shows that the logistic regression model was improved in every metric after adding both word-level n-gram features and lexicon derived features. However, the improvements are moderate.

The second section shows the performance of logistic regression models using the four types of features extracted from both a target comment and its contextsThe result shows that the logistic regression model using features extracted from a comment and both types of context achieved the best performance and obtained improvements of 2.8% and 2.5% in AUC score and F1-score respectively.

Model	Input Contents	Accuracy	Precision	Recall	F1	AUC
LSTM	comment	0.726	0.524	0.398	0.450	0.678
bi-LSTM	comment	0.720	0.513	**0.440**	0.473	0.682
bi-LSTM with attention	comment	**0.750**	**0.591**	0.437	**0.499**	**0.735**
	+ username	0.742	0.566	0.437	0.489	0.748
	+ title **(best)**	**0.766**	**0.614**	**0.499**	**0.548**	0.760
	+ title + username	0.755	0.589	0.496	0.532	**0.766**

Table 2.2: Performance of Neural Network Models

2.2.4.2 Neural Network Models

Table 2.2 shows the performance of neural network models. The first section of table 2.2 shows the performance of several neural network models that use comments as the only input. The model names are self-explanatory. We can see that the attention mechanism coupled with the bi-directional LSTM neural net greatly improved the online hate speech detection, by 5.7% in AUC score.

The second section of table 2.2 shows performance of the best neural net model (bi-directional LSTM with attention) after adding additional learning components that take context as input. The results show that adding username and news title can both improve model performance. Using news title gives the best F1 score while using both news title and username gives the best AUC score.

2.2.4.3 Ensemble Models

Table 2.3 shows performance of ensemble models by combining prediction results of the best context-aware logistic regression model and the best context-aware neural network model. We used two strategies in combining prediction results of two types of models. Specifically, the Max Score Ensemble model made the final decisions based on the maximum of two scores assigned by the two separate models; instead, the Average Score Ensemble model used the average score to make final decisions.

We can see that both ensemble models further improved hate speech detection performance

Model	Accuracy	Precision	Recall	F1	AUC
Char (Baseline)	0.738	0.549	0.469	0.504	0.733
Best Neural Network Model	0.766	0.614	0.499	0.548	0.760
Best Logistic Regression Model	0.750	0.572	0.516	0.542	0.778
Max Score Ensemble	0.740	0.539	**0.678**	**0.600**	0.794
Average Score Ensemble	**0.779**	**0.650**	0.496	0.560	**0.804**

Table 2.3: Performance of Ensemble Models

compared with using one model only and achieved the best classification performance. Compared with the logistic regression baseline, the Max Score Ensemble model improved the recall by more than 20% with a comparable precision and improved the F1 score by around 10%, in addition, the Average Score Ensemble model improved the AUC score by around 7%.

2.3 Analysis

2.3.1 Logistic Regression Models

As shown in table 2.1, given comment as the only input content, the combination of character n-grams, word n-grams, LIWC feature and NRC feature achieves the best performance. It shows that in addition to character level features, adding more features can improve hate speech detection performance. However, the improvement is limited. Compared with baseline model, the F1 score only improves 1.3%.

In contrast, when context information was taken into account, the performance greatly improved. Specifically, after incorporating features extracted from the news title and username, the model performance was improved by around 4% in both F1 score and AUC score. This shows that using additional context based features in logistic regression models is useful for hate speech detection.

2.3.2 Neural Network Models

As shown in table 2.2, given comment as the only input content, the bi-directional LSTM model with attention mechanism achieves the best performance. Note that the attention mechanism

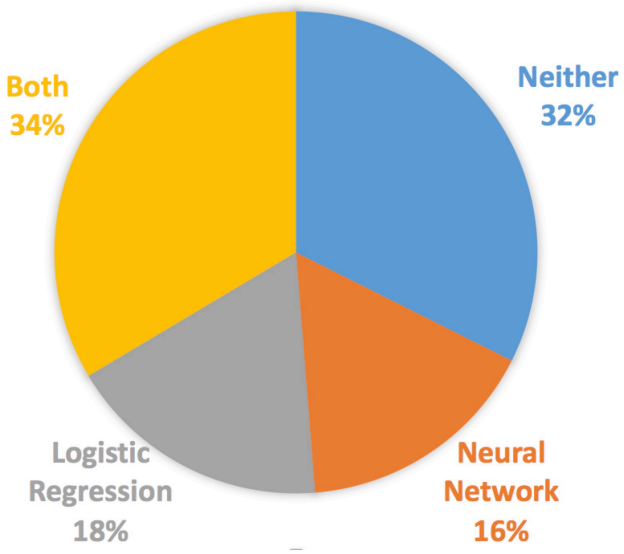

Figure 2.1: System Prediction Results of Comments Annotated as Hateful

significantly improves the hate speech detection performance of the bi-directional LSTM model. We hypothesize that this is because hate indicator phrases are often concentrated in a small region of a comment, which is especially the case for long comments.

2.3.3 Ensemble Models

As shown in table 2.3, both ensemble models significantly improved hate speech detection performance. the Max Score Ensemble model of neural network and logistic regression greatly improves the performance of our model. Compared to baseline system, the recall improves over 20%. The Average Ensemble Model achieves the best accuracy, precision and AUC score while the Max Ensemble Model achieves the best recall and F1 score. Since Max Score Ensemble can be regarded as a union of predictions from two models, we can go into details of actual contributions of each model when they are combined. Figure 2.1 shows the system prediction results of comments that were labeled as hateful in the dataset. we studied the distribution of hateful comments in the Max Score Ensemble model which reveals why the recall has a great improvement. For each prediction of hateful comment, we label it by the model which successfully predicts it. In total, there are 435 hateful comments in the dataset, the max ensemble model recognized 294 of them

correctly. Among the 294 true positive cases, 146 comments were labeled by both the neural net model and the logistic regression model while 71 and 77 comments were only recognized by the neural net model and the logistic regression model respectively. It can be seen that the two models perform differently. We further examined predicted comments and find that both types of models have unique strengths in identifying certain types of hateful comments.

2.3.3.1 Strengths of Logistic Regression Models

The feature based logistic regression models are capable of making good use of character-level n-gram features, which are powerful in identifying hateful comments that contains OOV words, capitalized words or misspelled words. By looking into those predicted samples, we find that the logistic regression is able to capture character level features, which greatly helps discover all capitalized rage comments and misspelled hateful content. Misspelled words, which is also OOV words, cannot be make good use of by neural net model. We provide two examples from the hateful comments that were only labeled by the logistic regression model:

(8)*kmawhmf:FBLM.*

Here FBLM means fuck Black Lives Matter. This hateful comment contains only character information which can exactly be made use of by our logistic regression model.

(9)*wulfeman007 @Spyderia BARRACK INSANE OBAMA HAS ALREADY HAD HIS NUTS CUT FOR HIS TRANSFORMATION AFTER HE LEAVES OFFICE*

This comment conveys its hatefulness by using all capitalized words.

(10)*SFgunrmn: what a efen loon, but most femanazis are.*

This comment deliberately misspelled feminazi for femanazis, which is a derogatory term for feminists. It shows that logistic regression model is capable in dealing with misspelling.

2.3.3.2 Strengths of Neural Network Models

The LSTM with attention mechanism are suitable for identifying specific small regions indicating hatefulness in long comments. In addition, the neural net models are powerful in capturing

implicit hateful language as well. The following are two hateful comment examples that were only identified by the neural net model:

(11)*freedomscout: @LarJass* ***Many religions are poisonous to logic and truth, that much is true****...and human beings still remain fallen human beings even they are Redeemed by the Sacrifice of Jesus Christ. So there's that. But the fallacies of thinking cannot be limited or attributed to religion but to error inherent in human motivation, the motivation to utter self-centeredness as fallen sinful human beings.* ***Nearly all of the world's many religions are expressions of that utter sinful nature****...Christianity and Judaism being the sole exceptions.*

This comment is expressing the stereotyping against religions which are not Christian or Judaism. The hatefulness is concentrated within the two bolded segments, which is usually the first argument and the last argument that best summarizes major points in this long comment.

(12)*freedomscout: The ones harmed the most are the children who are taught in government schooling that "transgenderism" is a valid description of a true reality. First, the very fact that "transgenderism" is not a true reality in itself but a condition of mind makes the claim of validity false...thus being deceptive for the children, youth, adolescents and adults who are deceived by it. Deception about anything critically important in the learning, growth, self-understanding and self-appreciation, and view of the civil society is evil and must be refused and refuted. To confuse children and youth, especially on the issues of their own nature is inexcusable and abusive to them. Because the falsehoods regarding "transgenderism" are being foisted on the culture by intention and willing complicity, the evil effects are widespread and become repeated in social media of all kinds.The result for stability of family and civil society is degradation of both.*

This comment is a typical case of long complex comment. It is composed of several arguments against transgenderism. Instead of expressing hate using emotional words or derogatory slurs, this kind of comments calmly argues against equality of sexual orientation, which is a hateful ideology.

(13)*mamahattheridge: blacks Love being victims.*

In this comment, the four words themselves are not hateful at all. But when combined together,

it is clearly hateful against black people.

(14)*MothraSaveUs: There is no law that will civilize the uncivilized.*

This comment is posted under the news of German Cologne assaults. It implies refugees to be "the uncivilized".

2.3.3.3 Error Analysis

After combining the two models, still the precision and recall are not good enough to apply to real world. We studied the false positive cases predicted by our model and we find that some of them are typical overfitting cases. When topics like BLM, muslim are mentioned more times in one comment, our classifier has higher tendency to label it as hateful since those topics are controversial and they are very likely to contain hate speech. For hate speech we did not find, many of them are replying to another comment. Our model did not model the interaction between each comment. We plan to do it in future work.

3. WEAKLY-SUPERVISED MODEL FOR HATE SPEECH DETECTION *

3.1 The Two-path Bootstrapping System for Online Hate Speech Detection

3.1.1 Overview

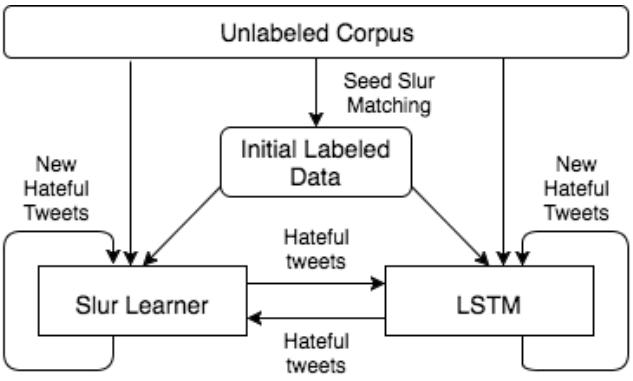

Figure 3.1: Diagram of LSTM-Slur Learner Co-training Model

Figure 3.1 illustrates that our weakly supervised hate speech detection system starts with a few pre-identified slur terms as seeds and a large collection of unlabeled data instances. Specifically, we experiment with identifying hate speech from tweets. Hateful tweets will be automatically identified by matching the large collection of unlabeled tweets with slur term seeds. Tweets that contain one of the seed slur terms are labeled as hateful.

The two-path bootstrapping system consists of two learning components, an explicit slur term learner and a neural net classifier (LSTMs [7]), that can capture both explicit and implicit descriptions of online hate speech. Using the initial seed slur term labeled hateful tweets, the two learning components will be initiated simultaneously. The slur term learner will continue to learn additional hateful slur terms. Meanwhile, the neural net classifier will be trained using the au-

*Part of this chapter is reprinted with permission from "Recognizing Explicit and Implicit Hate Speech Using a Weakly Supervised Two-path Bootstrapping Approach" by Lei Gao, Alexis Kuppersmith, Ruihong Huang, The 8th International Joint Conference on Natural Language Processing 2017, Copyright 2018 by IJCNLP 2017

tomatically labeled hateful tweets as positive instances and randomly sampled tweets as negative instances. Next, both the newly learned slur terms and the trained neural net classifier will be used to identify new hateful content from the unlabeled large collection of tweets. The newly labeled hateful tweets by each of the two learning components will be used to augment the initial slur term seed identified hateful tweet collection, which will be used to learn more slur terms and retrain the classifier in the next iteration. The whole process then iterates.

After each iteration, we have to determine if a stopping criterion is met and we should terminate the bootstrapping process. In general, a tuned threshold score is applied or a small annotated dataset is used to evaluate the learned classifiers. We adopt the latter method. Specifically, the bootstrapping system stops when the precision of the LSTM classifier is lower than 0.6 when evaluated using an existing small annotated tweet set [1].

Our model can be applied to detect hate speech in any text as long as the hate speech is relatively sparse. This is because when trainging the two learning components, we are using the unlabeled as non-hateful cases.

3.1.2 Automatic Data Labeling of Initial Data

Seeing a hate slur term in a tweet strongly indicates that the tweet is hateful. Therefore, we use 20 manually selected slur terms to match with a large unlabeled tweet collection in order to quickly construct the initial small set of hateful tweets. Table 3.1 shows the 20 seed slurs we used.

bimbo	chink	commie	coon	cunt
fag	faggot	feminazi	honky	islamist
libtard	muzzie	negro	nigger	paki
skank	subhuman	tranny	twat	wanker

Table 3.1: Seed Slurs

We obtained our initial list of slurs from Hatebase as mentioned in [34], the Racial Slurs

Database as mentioned in [35], and the page of List of LGBT slang terms in Wikipedia. We ranked the slur terms by their frequencies in tweets, eliminating ambiguous and outdated terms. The slur "gypsy", for example, refers to derogatorily to people of Roma descent, but currently in popular usage is an idealization of a trendy bohemian lifestyle. The word "bitch" is ambiguous, sometimes a sexist slur but other times innocuously self-referential or even friendly.

For these reasons, we only selected the top 20 terms we considered reliable (shown in Table 3.1). We use both the singular and the plural form for each of these seed slur terms.

Seed slurs are only used once to get the initial set of hateful tweets. For initialization of bootstrapping, human annotated tweets can also be an option.

3.1.3 Slur Term Learner

The slur term learning component extracts individual words from a set of hateful tweets as new slurs. Intuitively, if a word occurs significantly more frequently in hateful tweets than in randomly selected tweets, this term is more likely to be a hateful slur term. We also observed that when people are expressing hate, they tend to use synonyms of slur in parallel. For example, (1) *@annyaparada stupid bitch feminist faggot whore*. Following this intuition, we assign a score to each unique unigram that appears 10 or more times in hateful tweets, and the score is calculated as the relative ratio of its frequency in the labeled hateful tweets over its frequency in the unlabeled set of tweets. Then the slur term learner recognizes a unigram with a score higher than a certain threshold as a new slur. Specifically, we use the threshold score of 100 in identifying individual word slur terms.

The newly identified slur terms will be used to match with unlabeled tweets in order to identify additional hateful tweets. A tweet that contains one of the slur terms is deemed to be a hateful tweet.

While we were aware of other more sophisticated machine learning models, one purpose of this research is to detect and learn new slur terms from constantly generated user data. Therefore, the simple and clean string matching based slur learner is designed to attentively look for specific words that alone can indicate hate speech. In addition, this is in contrast with the second learning

component that uses a whole tweet and model its compositional meanings in order to recognize implicit hate speech. These two learners are complementary in the two-path bootstrapping system.

We were aware of the other more sophisticated machine learning models, but we intended to use the simple and clean string match based slur learning component for several reasons. First, this design was linguistically motivated. Seeing a slur in a text will immediately indicate the hatefulness of the text. Previous hate speech annotation and detection methods have emphasized on the extreme role of slur terms in conveying hate. Second, we meant to capture explicit hate speech using this approach that looks for matching with a clearly hateful slur term and ignores the other contents of a tweet, which well completes the LSTM approach that captures implicit hate speech by modeling compositional meanings of the whole tweet.

3.1.4 The LSTM Classifier

We aim to recognize implicit hate speech expressions and capture composite meanings of tweets using a sequence neural net classifier. Specifically, our LSTM classifier has a single layer of LSTM units. The output dimension size of the LSTM layer is 100. A sigmoid layer is built on the top of the LSTM layer to generate predictions. The input dropout rate and recurrent state dropout rate are both set to 0.2. In each iteration of the bootstrapping process, the training of the LSTM classifier runs for 10 epochs.

The input to our LSTM classifier is a sequence of words. We pre-process and normalize tokens in tweets following the steps suggested in [36]. In addition, we preprocessed of emoji and smiley using regular expression. Then we retrieve word vector representations from the downloaded pre-trained word2vec embeddings [22].

The LSTM classifier is trained using the automatically labeled hateful tweets as positive instances and randomly sampled tweets as negative instances, with the ratio of POS:NEG as 1:10. Then the classifier is used to identify additional hateful tweets from the large set of unlabeled tweets. The LSTM classifier will deem a tweet as hateful if the tweet receives a confidence score of 0.9 or higher. Both the low POS:NEG ratio and the high confidence score are applied to increase the precision of the classifier in labeling hateful tweets and control semantic drift in the bootstrap-

ping learning process. To further combat semantic drift, we applied weighted binary cross-entropy as the loss function in LSTM.

3.1.5 One vs. Two Learning Paths

As shown in Figure 1.1, if we remove one of the two learning components, the two-path learning system will be reduced to a usual self-learning system with one single learning path. For instance, if we remove the LSTM classifier, the slur learner will learn new slur terms from initially seed labeled hateful tweets and then identify new hateful tweets by matching newly learned slurs with unlabeled tweets. The newly identified hateful tweets will be used to augment the initial hateful tweet collection and additional slur terms can be learned from the enlarged hateful tweet set. The process will iterates. However as shown later in the evaluation section, single-path variants of the proposed two-path learning system are unable to receive additional fresh hateful tweets identified by the other learning component and lose learning momentum quickly.

3.1.6 Tackling Semantic Drifts

Semantic drift is the most challenging problem in distant supervision and bootstrapping. First of all, we argue that the proposed two-path bootstrapping system with two significantly different learning components is designed to reduce semantic drift. According to the co-training theory [37], the more different the two components are, the better. In evaluation, we will show that such a system outperforms single-path bootstrapping systems. Furthermore, we have applied several strategies in controlling noise and imbalance of automatically labeled data, e.g., the high frequency and the high relative frequency thresholds enforced in selecting hate slur terms, as well as the low POS:NEG training sample ratio and the high confidence score of 0.9 used in selecting new data instances for the LSTM classifier.

3.2 Evaluations

3.2.1 Definition of hate speech

Youtube, the largest online video provider, defines hate speech to be the language that refers to content that promotes violence or hatred against individuals or groups based on certain at-

tributes, such as race or ethnic origin, religion, disability, gender, age, veteran status, sexual orientation/gender identity.

We mainly adopt the definition of hate speech described in [1]. They assert that a statement is hateful if it:

(1) *uses a racial or sexist slur*

(2) *attacks or seeks to silence a minority*

(3) *promotes hate speech or violent crime*

(4) *criticizes a minority irrationally*

(5) *contains stereotyping of a certain minority*

(6) *defends sexism, racism, xenophobia or any other dangerous extremism*

3.2.2 Tweets Collection

We randomly sampled 10 million tweets from 67 million tweets collected from Oct. 1st to Oct. 24th using Twitter API. These 10 million tweets were used as the unlabeled tweet set in bootstrapping learning. Then we continued to collect 62 million tweets spanning from Oct.25th to Nov.15th, essentially two weeks before the US election day and one week after the election. The 62 million tweets will be used to evaluate the performance of the bootstrapped slur term learner and LSTM classifier. The timestamps of all these tweets are converted into EST. By using Twitter API, the collected tweets were randomly sampled to prevent a bias in the data set.

3.2.3 Supervised Baselines

We trained two supervised models using the 16 thousand annotated tweets that have been used in a recent study [1]. The annotations distinguish two types of hateful tweets, sexism and racism, but we merge both categories and only distinguish hateful from non-hateful tweets.

First, we train a traditional feature-based classification model using logistic regression (LR). We apply the same set of features as mentioned in [1]. The features include character-level bigrams,

trigrams, and four-grams.

In addition, for direct comparisons, we train a LSTM model using the 16 thousand annotated tweets, using exactly the same settings as we use for the LSTM classifier in our two-path bootstrapping system.

3.2.4 Evaluation Methods

We apply both supervised classifiers and our weakly supervised hate speech detection systems to the 62 million tweets in order to identify hateful tweets that were posted before and after the US election day. We evaluate both precision and recall for both types of systems. Ideally, we can easily measure precision as well as recall for each system if we have ground truth labels for each tweet. However, it is impossible to obtain annotations for such a large set of tweets. The actual distribution of hateful tweets in the 62 million tweets is unknown.

Instead, to evaluate each system, we randomly sampled 1,000 tweets from the whole set of hateful tweets that *had been tagged as hateful* by the corresponding system. Then we annotate the sampled tweets and use them to estimate precision and recall of the system. In this case,

$$precision = \frac{n}{1000}$$

$$recall \propto precision \cdot N$$

Here, n refers to the number of hateful tweets that human annotators identified in the 1,000 sampled tweets, and N refers to the total number of hateful tweets the system tagged in the 62 million tweets. We further calculated system recall by normalizing the product, $precision \cdot N$, with an estimated total number of hateful tweets that exist in the 62 million tweets, which was obtained by multiplying the estimated hateful tweet rate of 0.6%* with the exact number of tweets in the test set. Finally, we calculate F-score using the calculated recall and precision.

Consistent across the statistical classifiers including both logistic regression classifiers and

*We annotated 5,000 tweets that were randomly sampled during election time and 31 of them were labeled as hateful, therefore the estimated hateful tweet rate is 0.6% (31/5,000).

LSTM models, only tweets that receive a confidence score over 0.9 were tagged as hateful tweets.

Classifier	Precision	Recall	F1	# of Predicted Tweets	# of Estimated Hateful
Supervised Baselines					
Logistic Regression	0.088	0.328	0.139	**1,380,825**	121,512
LSTMs	**0.791**	0.132	0.228	62,226	49,221
The Two-path Weakly Supervised Learning System					
LSTMs	0.419	0.546	0.474	483,298	202,521
Slur Matching	0.565	0.398	0.468	261,183	147,595
Union	0.422	**0.580**	**0.489**	509,897	**214,997**
Union*	0.626*	0.258*	0.365*	-	-
Variations of the Two-path Weakly Supervised Learning System					
Slur Matching Only	0.318	0.143	0.197	166,535	52,958
LSTMs Only	0.229	0.303	0.261	491,421	112,535

Table 3.2: Performance of Different Models

3.2.5 Human Annotations

When we annotate system predicted tweet samples, we essentially adopt the same definition of hate speech as used in [1] and annotate according to their annotation guideline, which considers tweets that explicitly or implicitly propagate stereotypes targeting a specific group whether it is the initial expression or a meta-expression discussing the hate speech itself (i.e. a paraphrase). In order to ensure our annotators have a complete understanding of online hate speech, we asked two annotators to first discuss over a very detailed annotation guideline of hate speech, then annotate separately. This went for several iterations.

Then we asked the two annotators to annotate the 1,000 tweets that were randomly sampled from all the tweets tagged as hateful by the supervised LSTM classifier. The two annotators reached an inter-agreement Kappa [33] score of 85.5%. Because one of the annotators become unavailable later in the project, the other annotator annotated the remaining sampled tweets.

Its	Prev	Slur Match	LSTMs
1	8,866	422	3,490
2	12,776	4,890	13,970
3	27,274	6,299	21,579
4	50,721	9,895	22,768

Table 3.3: Number of Labeled Tweets in Each Iteration

3.2.6 Experimental Results

3.2.6.1 Supervised Baselines

The first section of Table 3.2 shows the performance of the two supervised models when applied to 62 million tweets collected around election time. We can see that the logistic regression model suffers from an extremely low precision, which is less than 10%. While this classifier aggressively labeled a large number of tweets as hateful, only 121,512 tweets are estimated to be truly hateful. In contrast, the supervised LSTM classifier has a high precision of around 79%, however, this classifier is too conservative and only labeled a small set of tweets as hateful.

3.2.6.2 The Two-path Bootstrapping System

Next, we evaluate our weakly supervised classifiers which were obtained using only 20 seed slur terms and a large set of unlabeled tweets. The two-path weakly supervised bootstrapping system ran for four iterations. The second section of Table 3.2 shows the results for the two-path weakly supervised system. The first two rows show the evaluation results for each of the two learning components in the two-path system, the LSTM classifier and the slur learner, respectively. The third row shows the results for the full system. We can see that the full system **Union** is significantly better than the supervised LSTM model in terms of recall and F-score. Furthermore, we can see that a significant portion of hateful tweets were identified by both components and the weakly supervised LSTM classifier is especially capable to identify a large number of hateful tweets. Then the slur matching component obtains an precision of around 56.5% and can identify roughly 3 times of hateful tweets compared with the supervised LSTM classifier. The last column

Intersection	LSTM Only	Slur Only
234,584	248,714	26,599

Table 3.4: Number of Hateful Tweets in Each Segment

of this section shows the performance of our model on a collection of human annotated tweets as introduced in the previous work [1]. The recall is rather low because the data we used to train our model is quite different from this dataset which contains tweets related to a TV show [1]. The precision is only slightly lower than previous supervised models that were trained using the same dataset.

Table 3.3 shows the number of hateful tweets our bootstrapping system identified in each iteration during training. Specifically, the columns **Slur Match** and **LSTMs** show the number of hateful tweets identified by the slur learning component and the weakly supervised LSTM classifier respectively. We can see that both learning components steadily label new hateful tweets in each iteration and the LSTM classifier often labels more tweets as hateful compared to slur matching.

Furthermore, we found that many tweets were labeled as hateful by both slur matching and the LSTM classifier. Table 3.4 shows the number of hateful tweets in each of the three segments, hateful tweets that have been labeled by both components as well as hateful tweets that were labeled by one component only. Note that the three segments of tweets are mutually exclusive from others. We can see that many tweets were labeled by both components and each component separately labeled some additional tweets as well. This demonstrates that hateful tweets often contain both explicit hate indicator phrases and implicit expressions. Therefore in our two-path bootstrapping system, the hateful tweets identified by slur matching are useful for improving the LSTM classifier, vice versa. This also explains why our two-path bootstrapping system learn well to identify varieties of hate speech expressions in practice.

We randomly sampled 1,000 tweets from each of the three segments of system predicted hateful tweets and have them annotated. We use the three sets of annotated tweet samples to calculate estimated number of truly hateful tweets and hence estimated precision and recall for each of the

two components in the bootstrapping system as well as the system overall. First, we estimate the number of truly hateful tweets in each segment based on the precision calculated using tweet samples for the same segment. Then we obtain the estimated number of truly hateful tweets labeled by each of the two hate detection components by summing up the estimated number of truly hateful tweets from the intersection segment and from one other segment corresponding to the component. Next, we obtain the number of predicted tweets by each of the two components by summing up the number of predicted tweets from the intersection segment and from the segment corresponding to the component. Then the estimated precision of each component is the ratio of the estimated number of truly hateful tweets over the predicted number of tweets by each component. In addition, we obtain these three metrics when applying the complete system (Union) by considering tweet samples across the three segments.

3.2.6.3 One-path Bootstrapping System Variants

In order to understand how necessary it is to maintain two learning paths for online hate speech detection, we also ran two experiments with one learning component removed from the loop each time. Therefore, the reduced bootstrapping systems can only repeatedly learn explicit hate speech (with the slur learner) or implicit hateful expressions (with the LSTM classifier).

The third section of Table 3.2 shows the evaluation results of the two single-path variants of the weakly supervised system. We can see that both the estimated precision, recall, F score and the estimated number of truly hateful tweets by the two systems are significantly lower than the complete two-path bootstrapping system, which suggests that our two-path learning system can effectively capture diverse descriptions of online hate speech, maintain learning momentums as well as effectively combat with noise in online texts.

3.3 Analysis

The fact that logistic regression baseline has very low precision is surprising. Despite the fact that character ngram feature have showed its effectiveness in hateful speech classification [1, 5], it seems to fail when used in unsupervised approach. The intuition of using character ngram

features is because that prefix and suffix of English words like 'ist', 'ism', 'xis' are indicative hatefulness. However, our experiment shows that character ngram feature does not work when applied to extremely unbalanced data(0.6% positive cases).

The performance of supervised baseline of LSTM shows the effectiveness of neural network in unbalanced classification. However, it is unable to discover large numbers of hate speech. We found that most of the hateful tweets found by LSTM baseline are related to Muslims and women while the hateful tweets in the 16k training set are mainly about anti-Muslim and sexism. This shows that supervised approaches suffer from biased data selection.

If we were to consider a lower confidence threshold for supervised LSTM baseline, take 0.5 for example, it would then label over 2 million tweets to be hateful, which will result in a very low precision.

Compared to supervised baselines, although our model only has the 'supervision' from 20 seed rules, it manages to find more than 4 times of hateful tweets. This shows that our weakly supervised approach outperforms supervised approach in large scale data.

In the following sections, we will discuss and analysis on the behavior and predictions of two learning components, the slur learner and LSTM classifier.

3.3.1 Analysis of the Learned Hate Indicators

berk	chavs	degenerates	douches
facist	hag	heretics	jihadists
lesbo	pendejo	paedo	pinche
retards	satanist	scum	scumbag
slutty	tards	unamerican	wench

Table 3.5: New Slurs Learned by Our Model

We have learned 306 unigram phrases using the slur term learning component. Among them, only 45 phrases were seen in existing hate slur databases while the other terms, 261 phrases in

total, were only identified in real-world tweets. Table 3.5 shows some of the newly discovered hate indicating phrases. Our analysis shows that 86 of the newly discovered hate indicators are strong hate slur terms which does not exist in existing list of slurs and the remaining 175 indicators are related to discussions of identity and politics such as 'supremacist' and 'Zionism'. Specifically, they are:

- A specific slur that is not named in our list of slurs such as unamerican or deplorables. Indicators in this category are either misspellings of common hateful words, neologisms, or words that have taken on a newly inflammatory meaning due to current events.
- A word/phrase that evaluates or draws attention to hateful speech. For example, bigot, facist, or racist.
- A word/phrase which is not relevant to hate speech such as the words noun and verb.

3.3.2 Analysis of LSTM Identified Hateful Tweets

The LSTM labeled 483,298 tweets as hateful, and 172,137 of them do not contain any of the original seed slurs or our learned indicator phrases. The following are example hateful tweets that have no explicit hate indicator phrase:

(1) *@janh2h The issue is that internationalists keep telling outsiders that they're just as entitled to the privileges of the tribe as insiders.*

(2) *@VikingsFan1964 nauseating traitor to America that's Hillary when more Americans are raped & murdered by illegals+ sharia, you'll B Sorry*

(3) *@YeIIowbang like it ain't shows like 16 & pregnant showing off the wild Caucasian*

(4) *@tavleen_singh Shut muezzin calls, abolish Triple talaq, FGM, Madrassas, Road Namaaz & goat slaughter.Equal rights, pay & free edu for women*

(5) *This is disgusting! Christians are very tolerant people but Muslims are looking to wipe us our and dominate us! Sen https://t.co/7DMTIrOLyw*

We can see that the hatefulness of these tweets is determined by their overall compositional

meanings rather than a hate-indicating slur which supports our assumption that our system can identify implicit hateful speech.

3.3.3 Error Analysis

The error of our model comes from semantic drift in bootstrapping learning, which partially results from the complexity and dynamics of language. Specifically, we found dynamic word sense of slurs and natural drifting of word semantic. Many slur terms are ambiguous and have multiple word senses. For instance, “Chink”, an anti-Asian epithet, can also refer to a patch of light from a small aperture. Similarly, “Negro” is a toponym in addition to a racial slur. Further, certain communities have reclaimed slur words. Though the word “dyke” is derogatory towards lesbians, for example, some use it self-referentially to destigmatize it, a phenomenon we sometimes encountered.

We also found that the slur learner began to pick up terms about football games and lyrics from popular music. The former is because some football fans use offensive language in trash-talking the opponents of their home teams. As for the latter, we found that racial slurs sometimes appear in popular music. For example, the rule-based slur learner started to find terms about football or some lyrics of pop singers like Beyonce after three rounds of bootstrapping. We found out that this is because some football fans use a lot of offensive in trash talking against the opponent of their home team. Also, racial slurs in popular music can result in the learning of bad indicators, an issue we came across when the classifier chose a number of rules from Beyonce’s recent hit "Formation." These terms make the slur learner start to collect terms about football and pop songs.

To improve the precision of our model as well as deadling with semantic drift, we mention some useful alternatives in improving the precision of our co-training model. Applying some of these alternatives will be part of our future work.

- Use a larger training set. A smaller training set will suffer from occasional noise.
- Use more seed slurs or use annotated data to start bootstrapping. The
- Make the bootstrapping slower by setting higher thresholds.

- Manually inspect those slur after each iteration. This will need some supervision but the precision of learned slur will improve a lot.
- Use more unlabeled data in training the classifier.
- Use a weighted loss function to get higher precision when training the neural network.
- Replace LSTM with convolution neural network or any other classifier.

3.3.4 Temporal Distributions of Tagged Hateful Tweets

By applying our co-training model on the 62 million tweets corpus, we found around 510 thousand tweets labeled as hateful in total. We analyzed these hateful tweets by their content and spacial/temporal distributions.

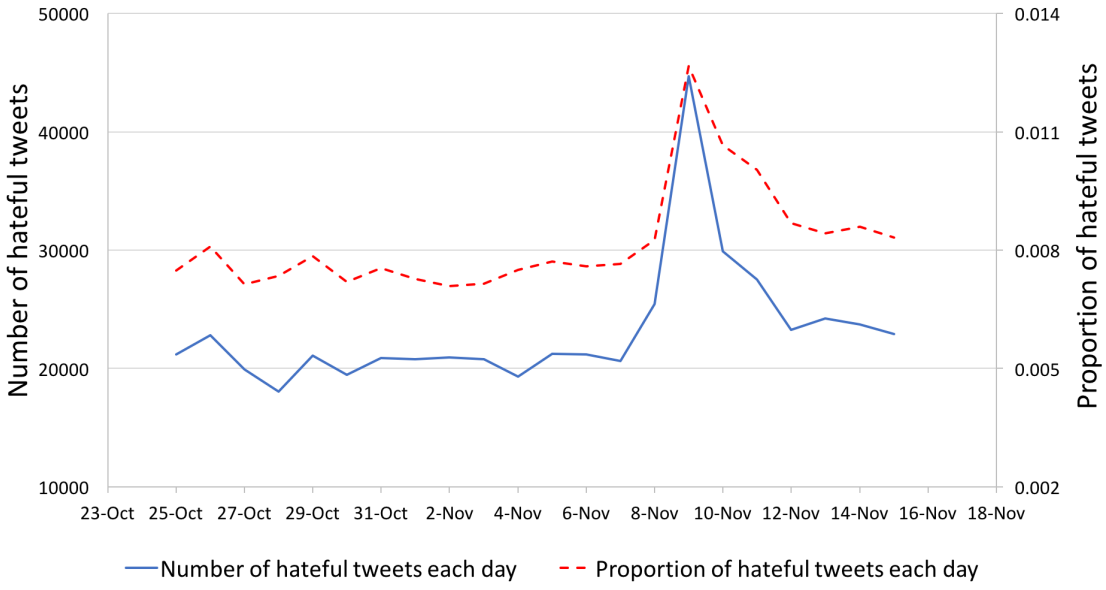

Figure 3.2: Temporal Distribution of Hateful Tweets

The figure 3.2 displays the temporal distribution of hateful tweets. There is a spike in hateful tweets from Nov.7th to Nov.12th in terms of both number of hateful tweets and ratio of hateful tweets to total tweets.

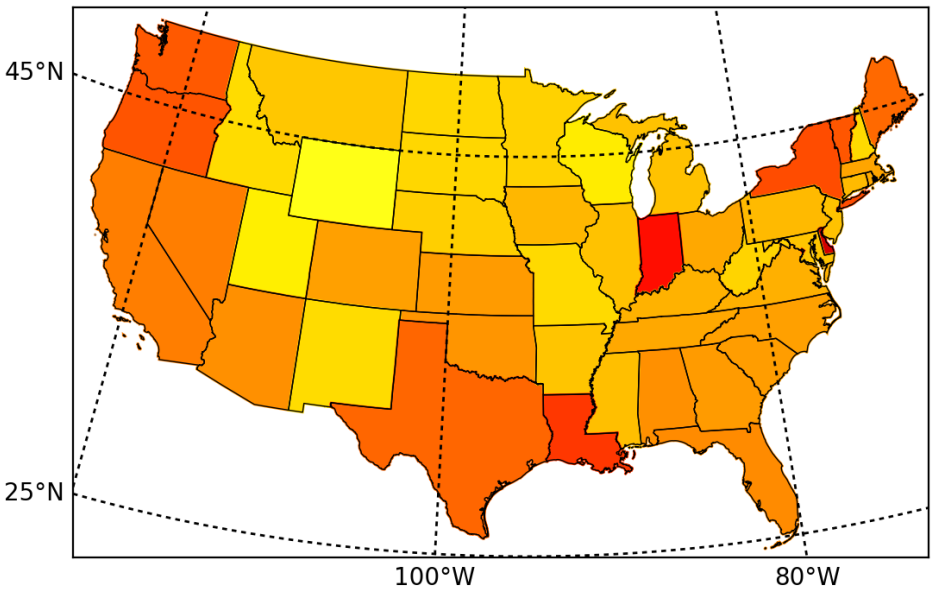

Figure 3.3: Geographical Distribution of Hateful Tweets

Figure 3.3 was produced using the geographic information of users who wrote inflammatory tweets and displays the number of hateful tweets per state† normalized by the population size of each state. A deeper color stands for a larger number of hateful tweets. Interestingly, states that strongly support one of the two parties (Democratic or Republican) and have a large population, such as Illinois, New York, and Washington, produced the greatest volume of hateful tweets. We find that the states with higher proportion of hateful tweets usually have larger population and they also strongly support one of the two parties. However, not all tweet users disclose their location. Therefore, the geographic distribution of hateful tweets may not be accurate.

@realDonaldTrump	@HillaryClinton	@megynkelly
@CNN	@FoxNews	@newtgingrich
@nytimes	@YouTube	@POTUS
@KellyannePolls	@MSNBC	@seanhannity
@washingtonpost	@narendramodi	@CNNPolitics
@PrisonPlanet	@guardian	@JoyAnnReid
@BarackObama	@thehill	@BreitbartNews
@politico	@ABC	@AnnCoulter
@jaketapper	@ArvindKejriwal	@FBI
@mitchellvii	@purplhaze42	@SpeakerRyan

Table 3.6: List of Top 30 Mentions in Hateful Tweets During Election Days

3.3.5 Most Frequent Mentions and Hashtags of Tagged Hateful Tweets

Table 3.6 shows the top 30 most frequent mentions in hateful tweets. They are ranked by frequency from left to right and from top to bottom.

It is clear that the majority of mentions found in tweets tagged as hateful address polarizing political figures (i.e. @realDonaldTrump and @HillaryClinton), indicating that hate speech is often fueled by partisan warfare. Other common mentions include news sources, such as Politico and MSNBC, which further support that "trigger" events in the news can generate inflammatory responses among Twitter users. Certain individual Twitter users also received a sizable number of mentions. @mitchellvii is a conservative activist whose tweets lend unyielding support to Donald Trump. Meanwhile, Twitter user @purplhaze42 is a self-proclaimed anti-racist and anti-Zionist. Both figured among the most popular recipients of inflammatory language. Van Jones, author and political commentator was also the recipient of many mentions.

Table 3.7 shows the top 30 most frequent mentions in hateful tweets. They are ranked by frequency from left to right and from top to bottom. It shows that the majority of hashtags also indicate the political impetus behind hate speech with hashtags such as #Trump and #MAGA (Make America Great Again, Trump's campaign slogan) among the most frequent. Second to politically charged hashtags are those that reference specific televised events such as reality shows and sports.

[†]By using the geographic information of users who wrote inflammatory tweets

#Trump	#ElectionNight	#Election2016
#MAGA	#trndnl	#photo
#nowplaying	#Vocab	#NotMyPresident
#ElectionDay	#trump	#ImWithHer
#halloween	#cdnpoli	#Latin
#Hillary	#WorldSeries	#1
#Brexit	#Spanish	#auspol
#notmypresident	#C51	#NeverTrump
#hiring	#bbcqt	#USElection2016
#tcot	#TrumpProtest	#XFactor

Table 3.7: List of Top 30 Hashtags in Hateful Tweets During Election Days

In regards to sports, a significant amount of gendered hateful language (such as the slur "cunt") is employed by sports fans while describing opposing teams or failures of their favored teams. The specific televised events also engender proportionally large amounts of hateful language as they can be commonly experienced by all television-owning Americans and therefore a widely available target for hateful messages. Some generic hashtags like #photo are evident of that hateful language can occur in mundane instances.

4. SUMMARY AND CONCLUSIONS

The first part of our work demonstrated the importance of utilizing context information for online hate speech detection. We first presented a corpus of hateful speech consisting of full threads of online discussion posts. It aims to call for attention on contextual information which has long been neglected by researchers in hate speech study community. In addition, we presented two types of models, feature based logistic regression models and neural network models, in order to incorporate context information for improving hate speech detection performance. Furthermore, we show that ensemble models leveraging strengths of both types of models achieve the best performance for automatic online hate speech detection.

The second part of our work focuses on the need to capture both explicit and implicit hate speech from an unbiased corpus. To address these issues, we proposed a weakly supervised two-path bootstrapping model to identify hateful language in randomly sampled tweets. Starting from 20 seed rules, we found 210 thousand hateful tweets from 62 million tweets collected during the election. Our analysis shows a strong correlation between temporal and geographical distributions of hateful tweets and the election time, as well as the partisan impetus behind large amounts of inflammatory language. In the future, we will look into linguistic phenomena that often occur in hate speech, such as sarcasm and humor, to further improve hate speech detection performance.

4.1 Further Study

We plan to explore unsupervised approaches in detecting online hate speech.

REFERENCES

[1] Z. Waseem and D. Hovy, "Hateful symbols or hateful people? predictive features for hate speech detection on twitter," in *Proceedings of NAACL-HLT*, pp. 88–93, Association for Computational Linguistics, 2016.

[2] Z. Waseem, "Are you a racist or am i seeing things? annotator influence on hate speech detection on twitter," in *Proceedings of the first workshop on NLP and computational social science*, pp. 138–142, 2016.

[3] E. Wulczyn, N. Thain, and L. Dixon, "Ex machina: Personal attacks seen at scale," *arXiv preprint arXiv:1610.08914*, 2016.

[4] B. Ross, M. Rist, G. Carbonell, B. Cabrera, N. Kurowsky, and M. Wojatzki, "Measuring the reliability of hate speech annotations: The case of the european refugee crisis," *arXiv preprint arXiv:1701.08118*, 2017.

[5] C. Nobata, J. Tetreault, A. Thomas, Y. Mehdad, and Y. Chang, "Abusive language detection in online user content," in *Proceedings of the 25th International Conference on World Wide Web*, pp. 145–153, International World Wide Web Conferences Steering Committee, 2016.

[6] I. Kwok and Y. Wang, "Locate the hate: Detecting tweets against blacks.," in *AAAI*, Association for the Advancement of Artificial Intelligence, 2013.

[7] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[8] N. Djuric, J. Zhou, R. Morris, M. Grbovic, V. Radosavljevic, and N. Bhamidipati, "Hate speech detection with comment embeddings," in *Proceedings of the 24th International Conference on World Wide Web*, pp. 29–30, ACM, 2015.

[9] S. Sood, J. Antin, and E. Churchill, "Profanity use in online communities," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pp. 1481–1490, ACM,

2012.

[10] P. Burnap and M. L. Williams, "Hate speech, machine classification and statistical modelling of information flows on twitter: Interpretation and communication for policy decision making," in *Proceedings of the Internet, Politics, and Policy conference*, Policy and Internet, 2014.

[11] W. Warner and J. Hirschberg, "Detecting hate speech on the world wide web," in *Proceedings of the Second Workshop on Language in Social Media*, pp. 19–26, Association for Computational Linguistics, 2012.

[12] H. Hosseinmardi, S. A. Mattson, R. I. Rafiq, R. Han, Q. Lv, and S. Mishra, "Detection of cyberbullying incidents on the instagram social network," *arXiv preprint arXiv:1503.03909*, 2015.

[13] A. Schmidt and M. Wiegand, "A survey on hate speech detection using natural language processing," *SocialNLP 2017*, p. 1, 2017.

[14] A. H. Razavi, D. Inkpen, S. Uritsky, and S. Matwin, "Offensive language detection using multi-level classification," in *Canadian Conference on Artificial Intelligence*, pp. 16–27, Springer, 2010.

[15] Y. Kim, "Convolutional neural networks for sentence classification," *arXiv preprint arXiv:1408.5882*, 2014.

[16] S. Lai, L. Xu, K. Liu, and J. Zhao, "Recurrent convolutional neural networks for text classification.," in *AAAI*, vol. 333, pp. 2267–2273, Association for the Advancement of Artificial Intelligence, 2015.

[17] C. Zhou, C. Sun, Z. Liu, and F. Lau, "A c-lstm neural network for text classification," *arXiv preprint arXiv:1511.08630*, 2015.

[18] G. Xiang, B. Fan, L. Wang, J. Hong, and C. Rose, "Detecting offensive tweets via topical feature discovery over a large scale twitter corpus," in *Proceedings of the 21st ACM international conference on Information and knowledge management*, pp. 1980–1984, ACM, 2012.

[19] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, *et al.*, "Scikit-learn: Machine learning in python," *Journal of machine learning research*, vol. 12, no. Oct, pp. 2825–2830, 2011.

[20] J. W. Pennebaker, M. E. Francis, and R. J. Booth, "Linguistic inquiry and word count: Liwc 2001," *Mahway: Lawrence Erlbaum Associates*, vol. 71, no. 2001, p. 2001, 2001.

[21] S. M. Mohammad and P. D. Turney, "Nrc emotion lexicon," tech. rep., NRC Technical Report, 2013.

[22] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," in *Advances in neural information processing systems*, pp. 3111–3119, 2013.

[23] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," *arXiv preprint arXiv:1409.0473*, 2014.

[24] Y. Chen, Y. Zhou, S. Zhu, and H. Xu, "Detecting offensive language in social media to protect adolescent online safety," in *Privacy, Security, Risk and Trust (PASSAT), 2012 International Conference on and 2012 International Confernece on Social Computing (SocialCom)*, pp. 71–80, IEEE, 2012.

[25] X. Zhang, J. Zhao, and Y. LeCun, "Character-level convolutional networks for text classification," in *Advances in neural information processing systems*, pp. 649–657, 2015.

[26] Z. Yang, D. Yang, C. Dyer, X. He, A. Smola, and E. Hovy, "Hierarchical attention networks for document classification," in *Proceedings of NAACL-HLT*, pp. 1480–1489, Association for Computational Linguistics, 2016.

[27] A. Conneau, H. Schwenk, L. Barrault, and Y. Lecun, "Very deep convolutional networks for text classification," in *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, vol. 1, pp. 1107–1116, Association for Computational Linguistics, 2017.

[28] S. Hingmire and S. Chakraborti, "Sprinkling topics for weakly supervised text classification," in *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, vol. 2, pp. 55–60, 2014.

[29] J. Ebrahimi, D. Dou, and D. Lowd, "Weakly supervised tweet stance classification by relational bootstrapping," in *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pp. 1012–1017, Association for Computational Linguistics, 2016.

[30] R. Johnson and T. Zhang, "Supervised and semi-supervised text categorization using lstm for region embeddings," in *International Conference on Machine Learning*, pp. 526–534, 2016.

[31] T. Miyato, A. M. Dai, and I. Goodfellow, "Adversarial training methods for semi-supervised text classification," *arXiv preprint arXiv:1605.07725*, 2016.

[32] Y. R. Tausczik and J. W. Pennebaker, "The psychological meaning of words: Liwc and computerized text analysis methods," *Journal of language and social psychology*, vol. 29, no. 1, pp. 24–54, 2010.

[33] J. Cohen, "A coefficient of agreement for nominal scales," *Educational and psychological measurement*, vol. 20, no. 1, pp. 37–46, 1960.

[34] L. Silva, M. Mondal, D. Correa, F. Benevenuto, and I. Weber, "Analyzing the targets of hate in online social media," *arXiv preprint arXiv:1603.07709*, 2016.

[35] M. Hasanuzzaman, G. Dias, and A. Way, "Demographic word embeddings for racism detection on twitter," in *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, vol. 1, pp. 926–936, Association for Computational Linguistics, 2017.

[36] J. Pennington, R. Socher, and C. D. Manning, "Glove: Global vectors for word representation.," in *EMNLP*, vol. 14, pp. 1532–1543, Association for Computational Linguistics, 2014.

[37] A. Blum and T. Mitchell, "Combining labeled and unlabeled data with co-training," in *Proceedings of the eleventh annual conference on Computational learning theory*, pp. 92–100, ACM, 1998.

APPENDIX A

ANNOTATION GUIDELINES

A.1 Annotation Guidelines For Fox News User Comment Corpus

Inflammatory language explicitly or implicitly threatens or demeans a person or a group based upon a facet of their identity such as gender, ethnicity, or sexual orientation. In our definition, we exclude insults towards other anonymous users and include insults of belief systems.

Here is a list of annotation examples:

■**stereotypes**: *EZinTexas: Hey woman, bring me a beer and go make me a ham sammich. Chop chop...!*

■**invokes violence**: *woetothegutless: So, how many sexual assaults would there be if Muslims were burned in ovens?*

■**Disparaging slurs**: *3865: Its failing has failed to remove any illegals period there all criminal*

■**Explicit statement of the inferiority of a group**: *FitRon21: Outlaw the savage cult of muzzieism, period!*

■**dehumanization**: *PantsSuit: What does NAACP stand for? Now Apes Are Called People*

■**xenophobia**: *Hairdic: Weve been invaded, folks. And now were being colonized by the invaders. SAYS IT ALL! And OUR own government made it happen!*

■**infantilization**: *ThomasK777: This just in.....radical feminists at liberal college accuse all male fraternity members of being Bill Cosby. Wow, first chalk was found to send liberals spiraling out of their happy place safe zone, and now bed sheets can apparently do it as well.*

■**sweeping generalization that is not a stereotype**: *UnbelievableGodhelpyou: Perhaps it is time to admit the real truth. That religion...ALL religion is cancer. Cults should be outlawed for the evil that they are.*

■**Implication of uncivilized nature of one group**: *Lookandlisten: Islamic countries never had a renaissance and Islam never had a reformation. They need to start over.*

■**Fear of power grab/inversion of political/social order**: *Mtgmike: It's a war on straight men with spines and common sense. Stop working with these feminist lunatics, if you're white and male they hate you no matter what you do.*

■**Suggestion that groups should be excluded**: *Davidisaacferguson; cant we just put feminists,black lives matters and other people and just put them on an island all to themselves*

■**Using they in lieu of naming a group**: *Karek40: THESE PEOPLE DO NOT BELIEVE THEY HAVE TO OBEY THE LAW.*

■**Dismissal of activism as irrational or crazy**: *MrCrabs2322: These whack jobs are anarchist! If you disagree with them they protest. If you agree with them they protest.(In reference to feminists)*