Towards Automatic Detection and Explanation of Hate Speech and Offensive Language

Wyatt Dorris*†
Clemson University
Clemson, SC
wyattadorris@gmail.com

Ruijia (Roger) Hu*†
Clemson University
Clemson, SC
roger.rj.hu@gmail.com

Nishant Vishwamitra*
Clemson University
Clemson, SC
nvishwa@clemson.edu

Feng Luo
Clemson University
Clemson, SC
luofeng@clemson.edu

Matthew Costello
Clemson University
Clemson, SC
mjcoste@clemson.edu

ABSTRACT

The use of hate speech and offensive language online has become widely recognized as a critical social problem plaguing today's Internet users. Previous research in the detection of hate speech and offensive language has primarily focused on using machine learning approaches to naively detect hate speech and offensive language, without explaining the reasons for their detection. In this work, we introduce a novel hate speech and offensive language defense system called `HateDefender`, which consists of a detection model based on deep Long Short-term Memory (LSTM) neural networks and an explanation model based on the gating signals of LSTMs. `HateDefender` effectively detects hate speech and offensive language (average accuracy of 90.82% and 89.10% on hate speech and offensive language, respectively) and explains their factors by pinpointing the exact words that are responsible for causing them. Our system uses these explanations for the effective intervention of such incidents online.

CCS CONCEPTS

• **Security and privacy → Social network security and privacy**; **Social aspects of security and privacy**; • **Social and professional topics → Hate speech**.

ACM Reference Format:
Wyatt Dorris, Ruijia (Roger) Hu, Nishant Vishwamitra, Feng Luo, and Matthew Costello. 2020. Towards Automatic Detection and Explanation of Hate Speech and Offensive Language. In *Proceedings of the Sixth International Workshop on Security and Privacy Analytics (IWSPA '20), March 18, 2020, New Orleans, LA, USA.* ACM, New York, NY, USA, 7 pages. https://doi.org/10.1145/3375708.3380312

*These authors contributed equally to this work.
†Intern from D.W. Daniel High School, Central, SC, USA.

IWSPA '20, March 18, 2020, New Orleans, LA, USA

ACM ISBN 978-1-4503-7115-5/20/03…$15.00
https://doi.org/10.1145/3375708.3380312

1 INTRODUCTION

The growing popularity of Online Social Networks (OSNs) has led to new paradigms of information sharing. Users of OSNs, such as Twitter, can share information instantaneously with a large number of people all over the world. However, one of the biggest issues of information sharing in OSNs is their inherent potential to engender *hate speech* and *offensive language*, which have been widely recognized as serious social problems. Recent research [2, 8] has shown that as much as 70% of Internet users are now being exposed to hate speech and offensive language on a regular basis.No formal definition of hate speech exists in the current scientific literature. However, there is a consensus that it is speech that targets disadvantaged social groups in a manner that is potentially harmful to them [9, 17]. In our work, we consider hate speech to be abusive speech directed towards a particular group of people [7]. Furthermore, we consider offensive language as the presence of offensive or curse words in online comments [19]. Hate speech laws vary among different nations. In the United States, hate speech is protected under the provisions of the First Amendment, but it has been extensively debated in the legal sphere. Countries such as the United Kingdom, Canada, and France have laws prohibiting hate speech, which tends to be defined as speech that targets minority groups in a way that could promote violence or social disorder. Instances of hate speech may appear in all kinds of OSN platforms. These instances can have severe negative impact on users worldwide.

To address the growing problem of hate speech, automatic hate speech detection models have been recently developed [5, 6, 11, 15, 16]. For example, a logistic regression-based model is introduced in [3], which performs reasonably well on offensive language detection, however, it may not be able to accurately detect hate speech (40% of hate speech mis-classified). The authors of [16] use a different approach, where sentence structure is used to detect hate speech (e.g., $< intensity >< user\ intent >< hate\ target >$). A major limitation of this approach is that it may not be able to capture hate speech that does not fit into this pre-defined sentence structure. The authors of [10] train an SVM-based classifier to classify comments as hate speech and non-hate speech. However, the limitation in performance of this classifier may not allow a practical use in OSNs. In addition, none of the existing works discuss any

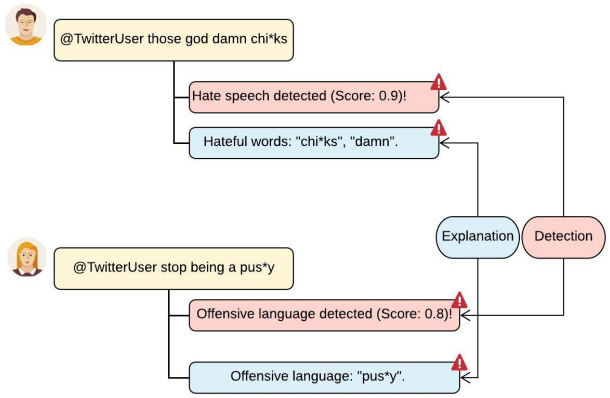


Figure 1: An illustration of hate speech and offensive language detection and intervention by `HateDefender`.

strategies to *explain* hate speech and offensive language or discuss factors that are responsible for them.

Our study reveals that there are two critical challenges that must be resolved to ensure a robust defense against hate speech and offensive language. First, the highly contextual and subjective nature of hate speech and offensive language have posed enormous difficulties for their accurate analysis. Therefore, the detection of hate speech and offensive language is a complex problem. Second, to understand the phenomenon of hate speech and offensive language and to enable effective intervention strategies to defend against them, new explanation and factor discovery strategies should be studied. These two critical challenges are depicted in Figure 1. In the first text, our system detects the text as hate speech (with a score of 0.9) using our detection model, and pinpoints the words responsible for causing hate speech ("ch*nks" and "d*mn") using our explanation methodology. In the second text, our system detects the text as offensive (with a score of 0.8) using the detection model, and pinpoints the offensive words ("pus*y") using our explanation methodology. The explanations generated by our methodology are used in intervention strategies, such as removing the hateful/offensive words or warning the senders.

Based on the observations and studies discussed above, we believe it is timely and important to systematically investigate online hate speech and offensive language, to understand these new phenomena, and to design approaches to accurately detect and explain them. In this work, we design a system called `HateDefender` for online hate speech and offensive language defense. `HateDefender` consists of a detection model that detects hate speech and offensive language based on deep Long Short-Term Memory (LSTM) neural network, and an explanation model based on word salience computed from the gating signals of the LSTM (for example, the words that are responsible for hate speech or offensive language may have high salience). `HateDefender` can detect hate speech and offensive language in an input text, and can also pinpoint the exact words in the text that are responsible for causing the hate speech or offensive language, which can be subsequently intervened by warning a user to reconsider sending the text. This feature of our system could be useful to obfuscate potentially offensive or hate speech text and warn users about not using such text.

Our contributions are summarized below.

- **Hate Speech and Offensive Language Detection.** Our system uses a deep LSTM-based neural network model to accurately detect hate speech and offensive language. Our detection model achieves an average accuracy of 90.82% and 89.10% on hate speech and offensive language, respectively, and outperforms the current baseline model [3] in terms of both precision and recall.
- **Explanation and Intervention.** In order to explain the predictions by our model, we use the gating signals of the trained LSTM model to compute salience of the words in the input text. The hate speech or offensive language incidents are then appropriately intervened by our system by issuing a warning to the sender about the high salience words computed by our explanation model.
- **Characterization of Hate Speech and Offensive Language.** We use our explanation technique to characterize hate speech and offensive language, by discovering factors of these phenomena. Our hate speech and offensive language factors could be an important contribution for social sciences and psychological research communities towards understanding these issues.

2 THREAT MODEL AND SCOPE

Threat Model. In this work, we consider two types of users: 1) a perpetrator is a user who sends a message with hate speech or offensive language towards a specific user or a group of users online; and 2) a victim who is a single user or a group of users. We consider the scenario where messages with hate speech or offensive content are sent by a perpetrator to a victim online. The affected users are the victims reading the message. In this work, we do not consider hate speech or offensive language cases with inside meaning that is only understandable to specific users. For example, a perpetrator Alice creates a new term to degrade users of a particular group and uses it to harass a victim Bob who belongs to this group.

Problem Scope. In this work, our goal is to detect hate speech and offensive language in text to a high degree of accuracy and explain the specific words that are responsible for causing them. Although there are multiple ways of defining what constitutes hate speech and offensive language, we have chosen to define hate speech as "speech that targets disadvantaged social groups in a manner that is potentially harmful to them" [9, 17] and offensive language as "offensive or curse words in online comments" [19]. Our work does not consider the impact of our system on the usability of existing social media systems.

3 OUR APPROACH

3.1 Overview

The goal of our work is to develop a system that can detect hate speech and also explain the reasons that are responsible for the hate speech or offensive language in an input sample. The overview of our system is depicted in Figure 2. In the first phase of our system (Figure 2, "Online Training Phase"), we use hate speech and offensive language data to train our detection and explanation models that are based on deep LSTM neural network. The detection model determines if an input sample is hate speech, offensive language, or

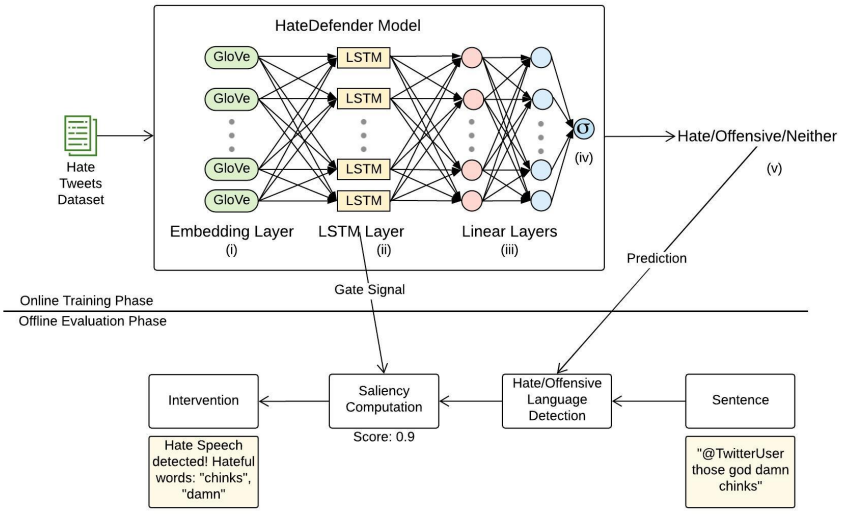


Figure 2: Overview of HateDefender System.

neither. The detection model's gating signals are used to compute a word salience score for each word in the input sample. In the second phase of our system (Figure 2, "Offline Evaluation Phase"), our system uses the salience score to report which words are responsible for causing hate speech or offensive language. This allows for users to be told what needs to be removed or changed from their messages to avoid being flagged as hate speech or offensive language.

3.2 Dataset

In our work, we used an existing annotated dataset [3] consisting of tweets labeled as hate speech, offensive language, or neither to train our models. We first preprocess our dataset using the following methodology. To reduce noise in the dataset, mentions to other Twitter users (such as "@DonaldJTrump") were replaced with the identifier "<TwitterUser>". Next, links in the text were replaced with an identifier "<link>". By replacing the terms with identifiers, the deep learning model can still learn important features in relation to people mentions and links without the random noise. To further filter out the noise, we lower cased all samples in the dataset. Finally, we tokenized all the samples in the dataset and converted them into GloVe embeddings [12].

Original Tweet	Filtered Tweet
"@SamJLayman : Holy sh*t, Freddie Highmore was in Charlie and the Chocolate Factory Me: *rolls on the floor, laughing*"	"<TwitterUser>: holy sh*t, freddie highmore was in charlie and the chocolate factory me: *rolls on the floor laughing*"
"Most hated but the Hoes favorite #2MW #SevenOne # http://t.co/BMdSVMc3rC"	"most hated but the hoes favorite #2mw #sevenone # <link>"
"@KyraNadiya: These hoes ain't loyal ; no they ain't http://t.co/h1UBsVbkGl"	"<TwitterUser>: these hoes ain't loyal ; no they ain't <link>"

Table 1: Examples of filtered tweets in the dataset.

Table 1 depicts three examples of original tweets from the dataset and the filtered tweets that have been preprocessed with our methodology. For example, it can be observed from Table 1 that user mentions such as "@SamJLayman" and "@KyraNadiya" have been replaced with the identifier "<TwitterUser>". Similarly, the Table 1 depicts that URLs in the dataset have been replaced with the identifier "<link>".

3.3 Background

In this section, we briefly discuss the LSTM neural network and focus our discussion on the operation of gating signals in LSTM. Recurrent units such as LSTMs are suitable in modelling sequence-based data, such as textual data. LSTMs specifically have been successfully applied to various sequence-based tasks such as sentiment analysis [18], machine translation [1] and caption generation [20]. We briefly discuss the LSTM unit for an input sample at a single time step t.

The LSTM unit takes as inputs, the current word x_t in the input sample, the activation of the previous word h_{t-1}, and the cell state from the previous time step c_{t-1}.

Three gate signals, input gate i_t, forget gate f_t, and output gate o_t are computed to decide whether the LSTM unit must consider the current word as relevant for the input sample. These gates are computed as follows:

$$i_t = \sigma(W_{ii}x_t + b_{ii} + W_{hi}h_{(t-1)} + b_{hi}) \quad (1)$$

$$f_t = \sigma(W_{if}x_t + b_{if} + W_{hf}h_{(t-1)} + b_{hf}) \quad (2)$$

$$o_t = \sigma(W_{io}x_t + b_{io} + W_{ho}h_{(t-1)} + b_{ho}) \quad (3)$$

The three gates of the LSTM unit are important in determining the way, in which the LSTM unit processes new information (we use a variant of the LSTM with an additional "cell gate", g_t). If the current word in the input sequence is important for the category (or the assigned label) of the input sample, the LSTM unit *remembers* the current word and consequently, the category is decided by the LSTM unit based on the gate signals using the Equations 1, 2 and 3.

In our explanation model, we utilize this feature of gating signals to compute word saliency (Section 3.4.2).

Next, a "candidate" activation is computed using the input and the hidden activations from the previous time step, h_{t-1}. If the LSTM unit determines that the current candidate is important for the input sample's category, the candidate is considered as the output of the LSTM unit. Otherwise, effectively the previous hidden activation is retained. The candidate is computed in Equation 4.

$$c_t = f_t * c_{(t-1)} + i_t * g_t \tag{4}$$

Finally, an output activation is computed for the current word given by Equation 5.

$$h_t = o_t * \tanh(c_t) \tag{5}$$

3.4 System Design

3.4.1 Hate Speech and Offensive Language Detection. The first goal of our `HateDefender` system is to accurately detect hate speech or offensive language in an input sample. To achieve this goal, `HateDefender` system contains a detection model based on LSTM units. Our hate speech detection model is depicted in Figure 2. Our model consists of a deep LSTM network consisting of LSTM units and Linear layers. Specifically, we use bi-directional LSTMs in our detection model. We begin by first training our model, depicted in the "Online Training Phase", in Figure 2. We first preprocess the input samples (Section 3.2) and transform the input samples into 100-dimensional GloVe representations [12] (Figure 2, Step (i)). This allows the LSMT network to associate similar words and make better predictions. Next, we train a bi-directional LSTM layer (Figure 2, Step (ii)), with the input sample embeddings generated in the GloVe layer.

From the LSTM layers, we get an output from each time step of the input sample. However, in the detection model, we only consider the output from the last time step (Equation 5, h_t). This output is then passed through linear (fully connected) layers (Figure 2, Step (iii)), so that the model can utilize the linear layer's trainable parameters to make more accurate predictions. Finally, we output a detection score according to Equation 6 (Figure 2, Steps (iv) and (v)).

$$Detection\ Score = Softmax(h_t) \tag{6}$$

Equation 6 outputs a probability for each category in our dataset (hate speech, offensive language, or neither). We consider the final category to be the category that has the highest probability.

In our work, we use a one-versus-rest, "ensemble" framework where a separate detection model is trained for each category and the category label with the highest predicted probability across all detection models is assigned to each input sample. We discuss the performance of our ensemble model in further detail in the Section 4.

3.4.2 Hate Speech and Offensive Language Explanation. Our explanation technique is based on the gate signals in a trained LSTM model, and comes into effect in the "Online Evaluation Phase". The LSTM gates control the flow of information in an LSTM and are therefore an important component of the LSTM for explanation. In our work, after our detection model detects hate speech or offensive language (Figure 2, "Hate Detection"), we first compute the gate signals for an input sample (Figure 2, "Gate Signal"), followed by computing a salience vector for each word in the input sample using the gating signals. Then, we normalize the resulting high dimensional vector into salience scores for the final explanation (Figure 2, "Saliency Computation"). This process is discussed in more detail below.

We first compute the gating signals with Equation 7 for each word in an input sample as follows. In this work, we use the output gate of the LSTM for explanation. Although, we found that the other gates of the LSTM can also be used in a similar manner.

$$o_t = \sigma(W_{io}x_t + b_{io} + W_{ho}h_{(t-1)} + b_{ho}) \tag{7}$$

In Equation 3, we use the weights and biases of the output signal of the LSTM. x_t and h_{t-1} are the current time step's input sequence and the previous time step's hidden activation, respectively. Next, we compute a "salience" score for the gate signal of each word in the input sequence. To compute the word salience, we compute the partial derivative of the output label of the input sample with respect to the gate signal.

$$\frac{\partial O_t}{\partial o_t} = \sigma(\frac{\partial O_t}{\partial W_{io}}x_t + \frac{\partial O_t}{\partial b_{io}} + \frac{\partial O_t}{\partial W_{ho}}h_{(t-1)} + \frac{\partial O_t}{\partial b_{ho}}) \tag{8}$$

In Equation 8, O_t represents the final label predicted by the model. In other words, the word salience represents the influence of the gating signal of specific word in the input sequence on the overall output label for that sequence.

The salience from Equation 8 is in the form of a high dimensional vector. To compute a salience score, we normalize the high dimensional vector using Min-Max normalization and output a score in the range of 0 to 1 (Equation 9).

$$salience_t = \frac{x_i - min(x)}{max(x) - min(x)} \tag{9}$$

The final salience scores computed in Equation 9 allows our system to pinpoint the exact words that are responsible for causing hate speech or offensive language. In our system, our explanations are used in intervention strategies (Figure 2, "Intervention") discussed in detail in the following section.

3.4.3 Intervention. Our detection model categorizes the input text according to three categories: hate speech, offensive language, and neither. If our detection model detects hate speech or offensive language in the input text, we use our intervention strategies to mitigate the presence of hate speech or offensive content in the input text. To achieve this, we need to know the exact words that are responsible for causing the hate speech or offensive language in the input sample. We use the explanation generated by our explanation model for this purpose, as our explanation model can pinpoint the exact words (high salience words) that are responsible for the hate speech or offensive language in the input sample.

In our work, we have used system generated warnings as an intervention strategy to mitigate the presence of hate speech or offensive language. We further explain this process with the following example.

A user *Alice* tweets the following text: "These sand ni**ers and porch monkeys need to stop fu**ing goats and get out of this country". This text is sent as an input sample to our detection model first. Our detection model detects hateful language in the input text. Next, we use our explanation model to pinpoint the exact words that may be responsible for causing the hate speech in the text. The explanation model pinpoints the words "sand ni**ers" and "fu**ing goats" as the highest salience words that are responsible for causing the hate speech. Finally, our intervention strategy generates a warning to the user, *Alice*, informing her that the text contains hate speech, and asking her to reconsider sending the words pinpointed by our explanation model.

4 IMPLEMENTATION AND EVALUATION

4.1 Implementation

Our detection model is implemented as a one layer, bi-directional LSTM. We use 100-dimensional GloVe [12] as word embedding model for our input samples for the purpose of transfer learning. We use one fully connected layer to improve prediction accuracy in our detection model. We train both our models jointly using 5-fold cross validation with 80% of the dataset for training and 20% for test. We use Adam optimizer with a learning rate of 0.0001 and we use Cross Entropy Loss as the loss function. For training the hate speech model, we use class weighting in the loss computation to mitigate the class imbalance in the dataset. We have used the PyTorch framework to train both our models. After training, we use the weights of the trained model for computing the partial derivatives involved in word salience computation.

4.2 Detection Model Evaluation

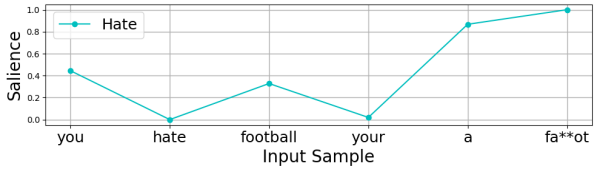
We use our test dataset to report our evaluation results of the detection model in `HateDefender`. In our model, we first discuss the accuracy, precision, and recall of the hate speech and offensive model. Then, we compare the performance of our detection model with an existing baseline [3]. The baseline model employs a logistic regression with L2 regularization and a one-versus-rest framework to determine the class label with the highest predicted probability. Table 2 depicts the performance of our detection model on the accuracy, precision, and recall metrics.

	Hate Speech	Offensive Language
Accuracy	90.82	89.10
Precision	60.56	83.82
Recall	64.71	84.23

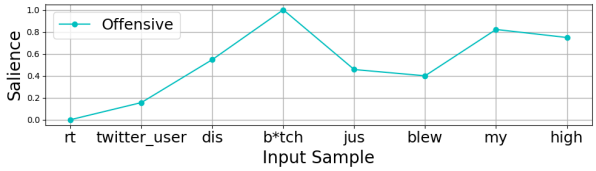
Table 2: Accuracy, precision, and recall of hate speech and offensive language detection models in `HateDefender`.

Overall, from Table 2, the model performs reasonably well on both hate speech and offensive language detection (average accuracy of 90.82% and 89.10% on hate speech and offensive language, respectively). Higher percentages of accuracy in both hate speech and offensive language may indicate that the cost of an incorrect prediction may be quite low.

From Table 2, it can be observed that the precision of the offensive language detection model is higher than the hate speech detection model. Precision indicates how many samples of hate speech and offensive language are predicted correctly when compared to how



(a) Word salience for hate speech sample.



(b) Word salience for offensive sample.

Figure 3: Word salience scores for hate speech and offensive language input samples generated by `HateDefender` explanation model.

many are actually hate speech or offensive language. The ratio of hate speech to offensive language to other words in the dataset was 35.2 percent hate speech, 53 percent offensive language, and 11.8 percent neither, which contributes to the higher value of offensive language because of the much larger number of offensive language samples in the dataset. We also observed a similar behavior for recall of the hate speech and offensive language models and we may similarly attribute this to the much larger number of offensive language samples in the dataset. However, our ensemble model (Table 3) improves both metrics, discussed in more detail below.

	HateDefender Ensemble Model	Baseline
Precision	65	44
Recall	63.43	61

Table 3: Precision and recall of `HateDefender` ensemble detection model in comparison with baseline model [3].

Next, we compare the performance of our ensemble detection model with an existing baseline [3]. In this test, we use a one-versus-rest, "ensemble" framework, where a separate detection model is trained for hate speech and offensive language and the category label with the highest predicted probability across the two detection models is assigned to the test sample. In this way, the ensemble model combines the hate speech and offensive language independent detection models and uses the results given by both to get a better prediction on whether or not text is hate speech or offensive language. Table 3 depicts the comparison results of the `HateDefender` ensemble detection model to the baseline model in [3]. Our ensemble model greatly improves on the precision score compared to the baseline model, while the recall score also improves on the baseline.

4.3 Explanation Model Evaluation

We evaluate the high salience words pinpointed by our explanation model by considering some randomly selected samples of hate speech and offensive tweets from our test dataset, depicted in Table 4. The Table 4 shows the hate speech and offensive samples with high salience words pinpointed by our model highlighted.

Label Type	Tweet	High salience words
Hate Speech	You hate football your a fa**ot .	fa**ot
	Ni**a just had them crackers on a chase.	Ni**a, crackers
	Told my dad to go buy cookies for the graduation this ni**a bought oreos.	this, ni**a, bought
	Happy first day of college ni**er twitter_user .	ni**er, twitter_user
	Them qu**r a** shorts got on .	qu**r
	twitter_user Studies show that you're a fa**ot .	a, fa**ot
	twitter_user Pu**y a** ni**a .	ni**a
	rt twitter_user Beyonce is trash .	trash
Offensive Language	I moved that ni**ah anyways	ni**ah
	rt twitter_user You bi*ches are so inconsiderate smh .	bi*ches
	rt twitter_user Dis bi*ch jus blew my high .	bi*ch
	Tryna hoe .	hoe

Table 4: Hate speech and offensive language samples with high salience words pinpointed by our model.

In Table 4, the more hateful or offensive our model considers the word, the higher salience rating it will receive, and also stronger highlighting. It can be observed from Table 4 that our explanation model is able to correctly pinpoint the words that are responsible for causing hate speech or offensive language (such as fa**ot, Ni**a, and bi*ches).

In order to demonstrate that hate speech and offensive words have high salience scores, we plot the salience scores of all words in two randomly selected input samples in Figure 3. Figure 3 (a) demonstrates the salience scores for a hate speech sample and Figure 3 (b) demonstrates the scores for an offensive sample. Our explanation model pinpoints the word "your" (Figure 3 (a)) as a word with low salience, because it is not responsible for causing hate speech or offensive language. However, our model indicates that the word "fa**ot" as high salience because it expresses the hate speech in the input sample.

4.4 Characterization of Hate Speech and Offensive Language

In this section, we discuss some interesting findings based on our explanation model (Section 3.4.2) about the hate speech and offensive language samples in the dataset and also characterize these two issues. In this test, we use our explanation technique to first determine all the salience words appearing in the hate speech and offensive input samples. Next, we exclude the words that are not high salience [1]. Table 5 depicts all the high salience words found in hate speech and offensive language input samples, respectively. We present our two findings below, based on the high salience words explained by our explanation model.

- **Hate Speech is Characterized by Specific Groups.** From Table 5, it can be seen that the high salience words in hate speech may primarily be aimed at specific groups of people. For example, words such as muz*ies, ni**er and fa**ot which are words that are attributed to certain groups of people were found to be high salience hate words. This result is consistent with a definition of hate speech that describes it as "speech that targets disadvantaged social groups in a manner that is potentially harmful to them" [9, 17]. However, we also note that some words that were found to be high salience hate words were not only limited to "disadvantaged social groups", but may also include words such as aryan, christian and anglo, which may not be attributed to disadvantaged groups. We surmise that such cases may be incidents of reverse hate speech.

Hate Speech	Offensive Language
muzzies, black, allah, ho*, asian, ni*,aryan, colored, eyes, nic*as, fa**ots, ni**as, mfka, old, coloreds, monkey, anglo, lux, slavery, hollywood, enjoy, saxon, stank, lightskin, illegal, queer, fa*s, girls, commies, white, wayans, american, ghetto, maidana, desert, kfc, retard, punk, spanish, ho*s, chava, fa*, bi**hes, rapper, darkie, ni**er, demons, trash, ni**a, israel, deported, conservative, christian, mustache, raghead, gay, fa**ot,garbage, african, losangeles	trippen, dumba**, fu*ks, peckerwood, dumpster, dog, fake, mf, breast, pimp, hungover, muthafu**in, as*, kicking, ti*s, redneck, co*k, bong, trash, mutha, fat, fool, bs, fu*k, shoot, queer, s*it, ni**uh, hairy, killed, dope, taliban, merica, girls-planks, lmfaooo, shyt, ni**ah, whore, pis*, drugs, sucking, idgaf, nasty, leak, homeboy, troll, spineless, fa*s, commie, juicy, crazy, scum, bi*chin, retard, rednecks, marijuana, fu**ed, fu**ing, bi*ch, sick, gurl, ho*, cunts, spit, bashed, midget, fu**in, di*k, rag, fu*, pisses, ni*lets, slut, af, beefing, dawg, ratchet, slutsquads, balls, butthurt, nipple, pus*y, cumming, ho*s, retards, tran*y, pus*ies, d*mn, stripper, motherfu**ers, boosie, lame, stinky, kush, skanks

Table 5: Characterization of hate speech and offensive words.

[1] In this experiment, we have excluded words below a salience threshold of 0.6, on a scale from 0 to 1.

- **Offensive Language is Characterized by Insults/Name Calling.** From Table 5, it can be seen that the high salience words in offensive language may primarily consist of insults and name calling. For example, words such as f**k, dumba** and wh*re that are typically insulting/curse words were found to be high salience offensive words. This result may be consistent with the definition of offensive language that describes it as "offensive or curse words in online comments" [19]. However, we also note that some words in offensive language samples may also be attributed to hate words, such as ni**ah and fa*s. We surmise that these words, although used for specific people groups, are also used as insults or curse words. Some words, like black or Allah, are characterized as hate speech because they pertain to the appearances or beliefs of a certain ethnic group. This is a form of targeting minorities and pertains to their culture. This is characterized as hate speech because they are sometimes derogatory terms against the ethnic group that is being targeted.

5 RELATED WORK

Several recent studies have emerged in the area of hate speech detection. A logistic regression based model was introduced in [3], which performs reasonably well on offensive language detection. However, this approach is unable to accurately detect hate speech (40% of hate speech mis-classified). To combat the rise of online hate speech, the authors of [10] trained an SVM classifier. This machine learning model classifies tweets as hateful and marks users who frequently use hate speech. However, their precision and recall scores (0.795 and 0.794) are low.

The authors of [4] made a distinction between types of hate speech (general vs specific) and special characteristics about each type. However, they did not show how this is specifically helpful in combating hate speech online. One problem of identifying hate speech online is that it is often obscure and only clear to human beings who can extract information behind symbols or acronyms. Authors of [14] offered two deep learning models to combat obscure meaning in hate speech. However, they found it hard to train machines to learn new hate symbols as the model cannot explain the hate words.

The authors of [11] used Gab (gab.com) to find out the diffusion of hate speech. For the dataset, they used a Lexicon based filter to identify racial slurs, and chose non-ambiguous words to increase accuracy. They also utilized DeGroot's model of information diffusion to identify hateful users. They focused on the diffusion characteristics of hateful users, but not how to pinpoint and remove hateful comments in general. In [13], the authors used a large dataset from Reddit and Gab and narrowed it down to hate speech by using human intervention, which is inefficient because it takes a long time to label so many tweets. It is also unreliable because there are some tweets that are incorrectly labeled. They used a survey and crowdsourcing to label all the tweets, which is not reliable, takes too much time, and adds cost. They created a dataset of hate speech and used programs like Seq2Seq and VAE. These are unreliable because it only uses an input and output tags, and does not go through multiple verifications. VAE is unreliable because it is just a probability distribution, and does not pinpoint certain hate words.

6 CONCLUSION

In this work, we have introduced a novel system called `HateDefender`, which can detect and explain hate speech and offensive text with high accuracy. Our system uses a deep LSTM neural network based detection model to accurately detect hate speech and offensive language and an explanation model based on LSTM gate signals to pinpoint and intervene hate speech and offensive words. Our detection model outperforms the baseline model and our explanation model allows us to intervene and to also provide new characterizations into the nature of hate speech and offensive language.

REFERENCES

[1] Kyunghyun Cho, Bart Van Merriënboer, Dzmitry Bahdanau, and Yoshua Bengio. 2014. On the properties of neural machine translation: Encoder-decoder approaches. *arXiv preprint arXiv:1409.1259* (2014).

[2] Matthew Costello, Rebecca Barrett-Fox, Colin Bernatzky, James Hawdon, and Kelly Mendes. 2018. Predictors of viewing online extremism among AmericaâĂŹs youth. *Youth & Society* (2018), 0044118X18768115.

[3] Thomas Davidson, Dana Warmsley, Michael Macy, and Ingmar Weber. 2017. Automated hate speech detection and the problem of offensive language. In *Eleventh international aaai conference on web and social media.*

[4] Mai ElSherief, Vivek Kulkarni, Dana Nguyen, William Yang Wang, and Elizabeth Belding. 2018. Hate Lingo: A Target-based Linguistic Analysis of Hate Speech in Social Media. arXiv:cs.CL/1804.04257

[5] Mai ElSherief, Shirin Nilizadeh, Dana Nguyen, Giovanni Vigna, and Elizabeth Belding. 2018. Peer to peer hate: Hate speech instigators and their targets. In *Twelfth International AAAI Conference on Web and Social Media.*

[6] Claudia I Flores-Saviaga, Brian C Keegan, and Saiph Savage. 2018. Mobilizing the trump train: Understanding collective action in a political trolling community. In *Twelfth International AAAI Conference on Web and Social Media.*

[7] Antigoni Maria Founta, Constantinos Djouvas, Despoina Chatzakou, Ilias Leontiadis, Jeremy Blackburn, Gianluca Stringhini, Athena Vakali, Michael Sirivianos, and Nicolas Kourtellis. 2018. Large scale crowdsourcing and characterization of twitter abusive behavior. In *Twelfth International AAAI Conference on Web and Social Media.*

[8] James Hawdon, Atte Oksanen, and Pekka Räsänen. 2017. Exposure to online hate in four nations: A cross-national consideration. *Deviant Behavior* 38, 3 (2017), 254–266.

[9] James B Jacobs, Kimberly Potter, et al. 1998. *Hate crimes: Criminal law & identity politics.* Oxford University Press on Demand.

[10] Rijul Magu, Kshitij Joshi, and Jiebo Luo. 2017. Detecting the hate code on social media. In *Eleventh International AAAI Conference on Web and Social Media.*

[11] Binny Mathew, Ritam Dutt, Pawan Goyal, and Animesh Mukherjee. 2019. Spread of hate speech in online social media. In *Proceedings of the 10th ACM Conference on Web Science.* ACM, 173–182.

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

[13] Jing Qian, Anna Bethke, Yinyin Liu, Elizabeth Belding, and William Yang Wang. 2019. A Benchmark Dataset for Learning to Intervene in Online Hate Speech. *arXiv preprint arXiv:1909.04251* (2019).

[14] Jing Qian, Mai ElSherief, Elizabeth Belding, and William Yang Wang. 2019. Learning to Decipher Hate Symbols. *Proceedings of the 2019 Conference of the North* (2019). https://doi.org/10.18653/v1/n19-1305

[15] Manoel Horta Ribeiro, Pedro H Calais, Yuri A Santos, Virgílio AF Almeida, and Wagner Meira Jr. 2018. Characterizing and detecting hateful users on twitter. In *Twelfth International AAAI Conference on Web and Social Media.*

[16] Leandro Silva, Mainack Mondal, Denzil Correa, Fabrício Benevenuto, and Ingmar Weber. 2016. Analyzing the targets of hate in online social media. In *Tenth International AAAI Conference on Web and Social Media.*

[17] Samuel Walker. 1994. *Hate speech: The history of an American controversy.* U of Nebraska Press.

[18] Yequan Wang, Minlie Huang, Li Zhao, et al. 2016. Attention-based LSTM for aspect-level sentiment classification. In *Proceedings of the 2016 conference on empirical methods in natural language processing.* 606–615.

[19] Wikipedia. 2019 (accessed November 18, 2019). *Profanity.* https://en.wikipedia.org/wiki/Profanity.

[20] Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhudinov, Rich Zemel, and Yoshua Bengio. 2015. Show, attend and tell: Neural image caption generation with visual attention. In *International conference on machine learning.* 2048–2057.