Revisiting Contextual toxicity Detection In Conversations

Vamsi Krishna Gaddamadugu
University Of Texas At Arlington
Arlington, Texas, USA

Akhil Gajulavarti
University Of Texas At Arlington
Arlington, Texas, USA

Jaswanth Vemulapalli
University Of Texas At Arlington
Arlington, Texas, USA

Abstract

There is little dispute that gaining an understanding of the poisonous nature of user conversations is a crucial topic. Context is essential for toxicity situations that are deemed "covert" or "implicit," as addressing these cases can be particularly challenging. There has only been a relatively small amount of research done in the past that has studied the effect that the context of a conversation has on human perception or automated detection models. We start by doing an analysis of previously collected contextual data-sets and find that the conversational structure, polarity, and topic of the context tend to have an effect on how humans label the level of toxicity of a substance. We next suggest incorporating these findings into computer detection models by introducing and analyzing neural architectures for contextual toxicity detection that are aware of conversational structure. This would be done in order to include these findings in computational detection models. The results of our study illustrate the potentially fruitful potential of neural architectures that are aware of the structure of conversations. We also demonstrate how such models might benefit from the use of synthetic data, notably in the realm of social media.

CCS Concepts: • **Computer systems organization** → **Embedded systems**; *Redundancy*; Robotics; • **Networks** → Network reliability.

Keywords: Toxicity Detection, Conversational Analysis, Machine Learning Algorithms

ACM Reference Format:
Vamsi Krishna Gaddamadugu, Akhil Gajulavarti, and Jaswanth Vemulapalli. 2023. Revisiting Contextual toxicity Detection In Conversations . In *Proceedings of jxv9221@mavs.uta.edu (ACM Conference).* ACM, New York, NY, USA, 7 pages. https://doi.org/XXXXXXX.XXXXXXX

ACM Conference, May 03–05, 2023, Arlington, TX

ACM ISBN 978-1-4503-XXXX-X/23/05…$15.00
https://doi.org/XXXXXXX.XXXXXXX

1 Introduction

1.1 HATE SPEECH DETECTION

Analyzing the content of a communication—including text, audio, and other forms of media—is one method of detecting hate speech. This method is used to determine whether a communication—including text, audio, and other forms of media—contains hatred or urges violence against an individual or a group of people. Discrimination against people based on "protected attributes" such as a person's race, gender, sexual orientation, religion, age, etc. is typically the primary motivating factor behind this. As an illustration, take a look at the benchmarks ETHOS and HateXplain. Metrics such as the F-measure and the F-score can be utilized in the process of evaluating model performance. The purpose of the process known as "hate speech detection" is to ascertain whether or not a form of communication, such as a written statement, an audio recording, or any other type of media, advocates for the physical harm of a person or a group of people. Prejudice against "protected qualities" like a person's color, gender, sexual orientation, religion, age, etc. is typically the cause of this. Two such instances are the benchmarks known as ETHOS and HateXplain. Metrics like the F-measure and the F-score can be utilized in the process of model evaluation. The website uses methods of machine learning to identify offensive comments. The models assign a score to a sentence based on their perception of how it will affect a conversation. Developers and publishers can use this rating to give feedback on comments, to make it simpler for comment moderators to review remarks, or to help readers filter out "toxic" language. Perspective models generate scores for a wide variety of different criteria. In addition to the flagship toxicity feature, Perspective can also provide scores for the following attributes: serious toxicology, insulting profanity, theft by identity, and sexually explicit threats.

1.2 Contextual Hate Speech Detection

Hate speech refers to any comment that is directed toward a certain group of individuals based on their sexual orientation, gender identity, race, religion, ethnicity, national origin, or national origin. It is common practice to disseminate bigotry and prejudice through the use of hate speech. It is also a tool that can be used to intimidate and threaten other individuals. It has the potential to cause people to experience feelings of isolation, anxiety, and fear. Hate crimes are another possible

consequence of it. A discourse based on hatred can likewise wreak havoc on the connections between people. The identification of hate speech is critical because it can contribute to the avoidance of these unfavorable outcomes. The purpose of monitoring communications (including text, audio, and other forms of media) with the intent of identifying instances of hate speech or content that incites violence against a specific person or group of people is known as "hate speech detection." Several scholarly articles have been published that explore toxicity and various facets of how it impacts members of our community. In addition, several studies that have presented numerous algorithms for the identification of toxicity have concluded that the most advanced techniques available today are worthless. One of the most widely used APIs for toxicity identification, Google's Perspective API, fails to function well when the context of an online chat is taken into consideration. This application programming interface (API) takes text as its input and returns a toxicity score between 0 and 1, where 1 is the most toxic and 0 is the least toxic. The score indicates the possibility that the person who viewed the comment would withdraw from the conversation. The website uses methods of machine learning to identify offensive comments. The models assign a score to a sentence based on their perception of how it will affect a conversation. But when it comes to interacting with people in the real world, they don't work very well. Take, for instance, a debate taking place on Twitter, in which several users publish comments or their perspectives on a specific subject.

Figure 1. This image depicts toxicity of 2nd comment depending on context

Take a look at the above picture, someone wrote a post about how bad one of the leading food industry's services or food taste is, and they responded to it. The second comment cannot be classified as toxic on its own. Therefore, what exactly is the problem with that? The word "context" really shines when it's applied in this situation. When it comes to the identification of hate speech, the surrounding environment is an extremely important factor in establishing the level of poison. For instance, if you look at figure1 the 2nd comment can never be classified as a toxic comment if it was processed alone. But if you take the context into the limelight we can see what's wrong with the comment.

Many research articles concentrated on the detection of hate speech. Take, for example, the research published in the article "Deep Learning for Hate Speech Detection: A Comparative Study," which examined the benefits and drawbacks of the most advanced detecting systems available today. However, their strategies do not take into account the surrounding environment. Among the earlier works that we looked at, "Revisiting Contextual Toxicity Detection in Conversations" was one of them. In this particular investigation, the researchers incorporated context into their machine-learning models by including context-RNN into their model to make toxicity predictions. Which one has more successful outcomes? This report will serve as the basis for our strategy moving forward. To begin, we collected information from a wide variety of sources. Hateful Quain Reddit, CAD (Contextual Abuse Dataset), BAD (Bot Adversarial Dialogue), and HQG (Hateful Quain Gab) were the types of data sets that we investigated. Target data and context data are both labels that are included in this data set. Our approach begins with utterances, in which the utterances (un) represent the data that have been labeled with the target, and the utterances (u1) represent the data that have been labeled with the context. while the context data is being maintained by two BERT models that each have two CLS tokens contained within them. Each CLS token is given a training session with the RNN model, and the results are saved in the "History" folder. While the target-labeled data also goes through the same process, a different RNN model is employed in this stage of the analysis. Please take note that we utilized two separate RNN models. "Target" is where the data from "Target" is saved. After that, the history and the target are combined, and this new information is utilized to make a toxicity prediction. The references contain clickable links that will take you to the various data sets.

2 Background and Related Work

"Revisiting Contextual Toxicity Detection in Conversations" is the primary prior work that we have done. The following is a list of their primary contributions:

- They present a comprehensive study of a data set on the detection of contextual toxicity, which sheds light on the peculiarities associated with the perception of contextual toxicity.
- The researchers investigated a variety of topologies for contextual toxicity identification. These structures are more suited to the context of a conversation and lead to enhanced performance.
- They investigated various data augmentation methods that are useful for modeling toxicity.

They looked at five different data sets: BBF, FBK, BAD, and CAD, in addition to HQG and HQR. There are two stages to the FBK data set: phase 1 and phase 2. Whereas phase one refers to the situation without context and phase two refers to the situation with context. Therefore, the labels in that data set have been switched so that 0 > 1 and 1 > 0. They also investigated the causes of why the data set is inverted, which they denoted with the letters C1, C2, C3, and C4, respectively. Take note that they came up with their original data augmentation strategies for their solution. They tested two data augmentation options for both the objective and its environment due to the inconsistency and limited size of the data sets: a generative approach and a transformative approach. Both approaches were meant to improve the data in some way. The former modifies previously collected training data to generate new content based on the original distribution in the training data, whereas the latter modifies collected training data to develop a variant of the original training data. They used (a) a fine-tuned GPT-2 language model and (b) the Style Transformer model so that they could put these strategies into action. To better handle the conversational structure and come to terms with their methodology, we develop and experiment with three different architectures, each of which encodes the context independently from the target comment and then combines the two using techniques such as late and model fusion. Specifically. Their strategy entails encoding the context as a string of sentences (U1...Un1), each of which has an independent BERT sentence (CLS) embedding. After that, these embeddings are merged to produce a history representation (H). They look into the use of the BERT summary token for the concatenated context utterances (ContextSingle), the sum of the context representations (ContextSum), and the last hidden state representation from a GRU model (ContextRNN) as possible summary alternatives. The historical representation is concatenated with the representation of the target post (Un), which is then input into a final classification layer, followed by sigmoid activation, to determine whether or not the sample is harmful.

3 Data-Set Collection

In this section, we will discuss all of the data sets that were used for this study. These data sets originated from a wide number of places and were sampled and labeled using a wide variety of techniques. Binary labels that differentiate harmful from non-toxic situations are taken into account across all of our data sets.

3.1 HQG and HQR

The HQR and HQG data sets were both developed with the help of data taken from conversations on Reddit and Gab, respectively. A method that is centered on keywords was utilized in the collection of the data. In other words, using a list of hateful keywords, the first step of the process identified potentially offensive remarks. After that, the conversational context of each comment was recreated by taking into consideration all of the comments that came before it and all of the comments that came after it in the thread. After the comments were retrieved, workers from the crowd went through and annotated each one to evaluate whether or not it contained hate speech. In the same way that we did with the BAD data set, we break each conversation thread into many samples, with each conversation becoming a separate training sample. In all data sets, instances have anywhere from one to twenty comments attached to them.

They propose a unique job of generative hate speech intervention coupled with two fully labeled data sets obtained from Gab and Reddit to stimulate the development of techniques for combating online hate speech. These data sets can be found here. This data set is distinct from others that focus on hate speech since it preserves the conversational context of the statements and includes human-written answers to the interventions. Due to the way, they gather their data, employees on Mechanical Turk review every post that is a part of their data sets to see if it contains hate speech. This ensures that the data sets are suitable for the task of identifying hate speech.

These data sets provide discussion segments, hate speech labels, and intervention answers written by Mechanical Turk workers. The two CSV files gab.csv and reddit.csv may be found within the data directory.

3.2 CAD

Reddit postings, specifically in subreddits that had the potential to have abuse levels that were higher than average, served as the source of the data for the CAD data set [**?**].

For each individual comment and post title, crowd workers completed annotations. Workers in the crowd were given the instruction to classify each individual's comment into one or more of the following categories: identity-directed abuse, affiliation-directed abuse, person-directed abuse, counter-speech, non-hateful insults, and neutral. When annotating each statement in the toxic situations, annotators were also asked to note whether or not they needed context. As part of our experiment, we have assigned the label "toxic" to each of the three abuse-related categories and have determined the context of each comment to be either all of the prior comments or the post titles of the same article.

4 Methodology

4.1 Neural Network

A recurrent neural network is one that retains the activations that the network's output at one or more of its layers caused. These are often activated at a later time and are hidden. After that, the next time an example of input is fed into the network, we include the previously recorded outputs as additional inputs. This continues until we have exhausted

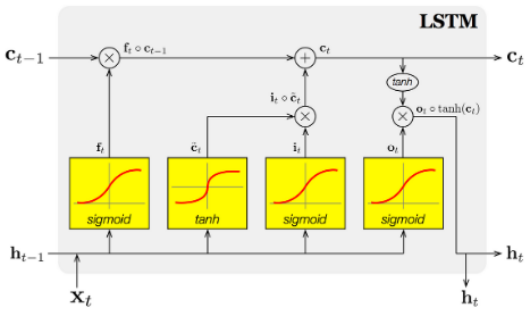

Figure 2. General LSTM Model

all possible combinations. Consider the additional inputs to be an addition to the "normal" inputs that the layer before them provided. For instance, if a hidden layer has 10 ordinary input nodes and 128 hidden nodes, it has 138 total inputs (this is assuming that the layer's outputs are fed into itself rather than into another layer, as Elman advises). When you first attempt to compute the output of the network, you will, of course, need to fill in those additional 128 inputs with 0s or some other value.

Long-term short-term memory units (or blocks), often known as LSTM, are the fundamental components of recurrent neural network (RNN) layers. An RNN that is made up of LSTM units is known as an LSTM network. The components of a typical LSTM unit are referred to as a cell, an input gate, an output gate, and a forget gate. Because the cell is responsible for "remembering" values across arbitrary time intervals, the term "memory" is employed in LSTM. This is one of the reasons why. Each of the three gates can be thought of as a "conventional" artificial neuron that is a component of a multi-layer or feed-forward neural network. This is because each of the gates computes the activation of a weighted sum by using an activation function. Because they can be conceptualized in an intuitive manner as gatekeepers for the passage of values across the LSTM connections, the term "gate" is used to refer to them. Both of these gates and the cell itself are connected to one another.

Figure 2 is a diagram depicting a general LSTM model.

RNNs are unable to use long-term information because they are plagued by an issue known as the vanishing gradient, despite the fact that they have a great deal of computational power. For instance, they are useful for storing memory in three to four instances of the preceding iteration, but because using conventional RNNs with larger numbers of instances does not produce satisfactory results, we do not merely rely on those. Long Short Term Networks, or LSTM for short, is an improved form of recurrent neural networks (RNNs) that are used here.

4.2 Implementation

They constructed and tested three different architectures in order to better handle the conversational structure. Each of these designs encodes the context independently from the target comment and then combines the two using late and model fusion approaches. They represent the context as a series of sentences (U1...Un1), each of which has its own BERT sentence (CLS) embedding, as shown in Figure 3. After that, these embeddings are merged to produce a history representation (H). They looked into three different ways to summarize information: utilizing the BERT summary token for the concatenated context utterances (ContextSingle), taking the total of the context representations (ContextSum), and using the context RNN model's last hidden state representation (ContextRNN). The historical representation is concatenated with the representation of the target post (Un), which is then input into a final classification layer, followed by sigmoid activation, in order to determine whether or not the sample is harmful. However, we focused solely on context-specific RNNs that have demonstrated superior performance in their results.

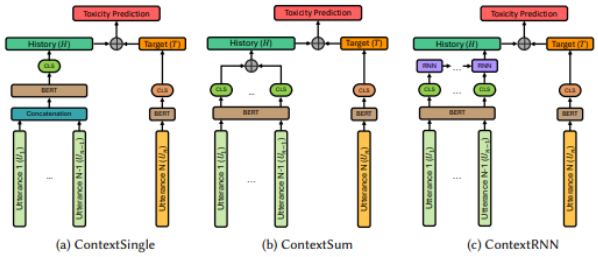

Figure 3. ContextSingle, ContextSum, and ContextRNN architectures. Each utterance U is represented by its BERT CLS token embedding. The context history (H) representation is taken as the summary CLS BERT representation for the concatenated context utterances (ContextSingle), the sum of the context representations (ContextSum), or the last hidden state representation from a GRU model (ContextRNN). The history representation is concatenated with the representation of the target post (Un) to produce toxicity predictions

Before moving on to the model, we want to make sure that the libraries that we employed are included. For this particular implementation, we relied on TensorFlow. Keras was incorporated into the layers as well. For the purpose of data processing, we incorporated libraries such as Numpy and Pandas. A natural language toolkit, often known as an nltk, was incorporated into the sentence analysis. After constructing a machine learning model with two thick layers and a compiler that uses binary cross-entropy, the next step is to construct an optimizer called Adam. Through the use of the sigmoid and soft-max functions, we activate two dense layers. After that, we combine that model with the LSTM

model. We utilized a dropout layer (spatial dropout 1d) so that we would not over-fit the data. The epoch size we used was 25, and the batch size we used was 64. After that, we look through the data sets for any null values and replace them with 0. After that, we perform some final adjustments to the data by removing any texts or labels that were included in the data sets. Finally, we change all of the data to lowercase. Before the tokenization stage, any text that uses the BERT-based uncased model must first be converted to lowercase. The data sets should then have any punctuation and spaces removed, followed by the removal of any stop-words. After that, we vectorize the texts in order to speed up the process of training the algorithm. Following that, we separated the data into training and testing sets. Because we are working on developing a toxicity predictor, we only separated the data for the very final data set (CAD). After that, we will need to train the model, and this process will be repeated for each data set. The accuracy graph for each data set is then plotted after this step. Then, for the user prediction, we take a second model with the LSTM algorithm that was used in the first model. After that, we put the evaluation models, such as the accuracy score, f1-score, and the ROC curve, into action. Then, as a last step, we plot the graphs to determine their accuracy.

5 Evaluation

Recently, the identification of multilingual bigotry has been a major concern, particularly in contexts where code-mixing or the usage of more than one language in online discussion has occurred. The performance of hate speech detection models is typically tested by utilizing measures such as accuracy and F1-score to examine how well the models perform on data that has been held out for testing. Despite the fact that these measures are helpful, they make it harder to determine where the model is failing and how to correct the problem.

5.1 Metrics

The F1 score is a machine learning evaluation statistic that quantifies the level of accuracy a model has attained. It does it by combining the scores for a model's precision and recall. The accuracy metric determines how many instances out of all the data in the data set a model accurately predicts something. We also compute the ROC curve by making use of the Sklearn ROC curve module, in addition to the AUC(Area Under Curve), which assists us in obtaining the tpr(True Positive Rates and FPR (False Positive Rates) values.

A receiver operating characteristic curve, more commonly referred to as a ROC curve, is a graph that displays the performance of a classification model's overall classification levels. The following two parameters are plotted on this curve:

- True Positive Rate
- False Positive Rate

A true positive outcome is one in which the model correctly predicts the positive class. A true negative, on the other hand, is an outcome in which the model correctly predicts the negative class. A false positive is an outcome in which the model predicts the positive class incorrectly. A false negative is an outcome in which the model predicts the negative class incorrectly.

AUC is an abbreviation that stands for "area under the receiver operating characteristic curve." More specifically, AUC is a measurement that determines the whole two-dimensional area that lies below the ROC curve in its entirety.

Accuracy is one metric for evaluating classification models. Informally, accuracy is the fraction of predictions our model got right.

Sklearn's evaluation modules were the source of our import. The ROC curve, AUC, f1-score, and accuracy score were both imported. Following that, we defined functions for the ROC, f1, and accuracy variables. We get the numbers for TPR (true positive rates) and for FPR (false positive rates) by calculating the area under the curve (AUC). After that, we obtained the tpr, for, and f1 scores by computing the ROC curve with the Sklearn ROC curve module in order to get the results. The results are shown in the next section.

Additionally, we attempted to converge the epoch sizes of the models in an effort to obtain more accurate results using an epoch size of 25, whereas the paper utilized an epoch size of 4. The findings of the accuracies were anywhere from 4 to 25.

6 Results

First, we want to talk about the F1 scores. It combines the precision and recall score of the model. Our scores are slightly less compared to the Context models that they have performed. Due to low precision and low recall our model's f1 score was affected.

Table 1. F1 Scores for each data set of our model and their model

Data sets	Their model performance	Our model performance
CAD	0.525	0.29942
Reddit	0.809	0.06061
GAB	0.908	0.05573

Generally an f1 score with

- >0.9 is an excellent interpretation

Next, we want to talk about the ROC curve. It shows us the performance of a classification model at all classification thresholds. In our case due to highly imbalanced data sets, Our ROC AUC curve was optimistic. we got a ROC score of 0.5 for each data set. Figure 4 shows the ROC AUC curve of our model.

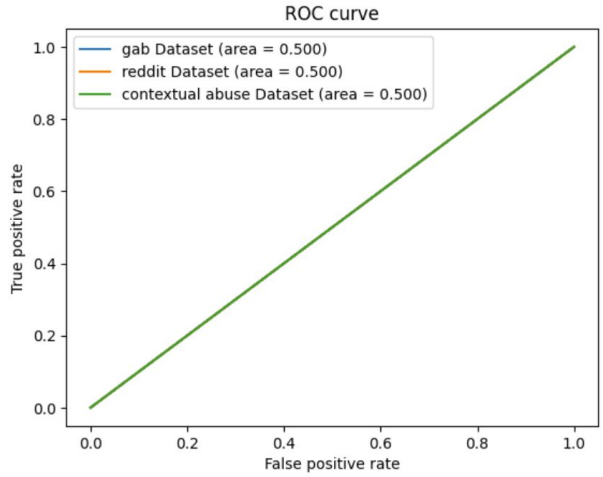

Figure 4. ROC Curve for each data set

Finally coming to the accuracies of each data set. we plotted graphs for each data set using the matplot library. Figure5,6,7 shows the accuracies for each data set. we accuracy of 81 for the CAD data set, 38.5 for the GAB data set, 43 for the Reddit data set

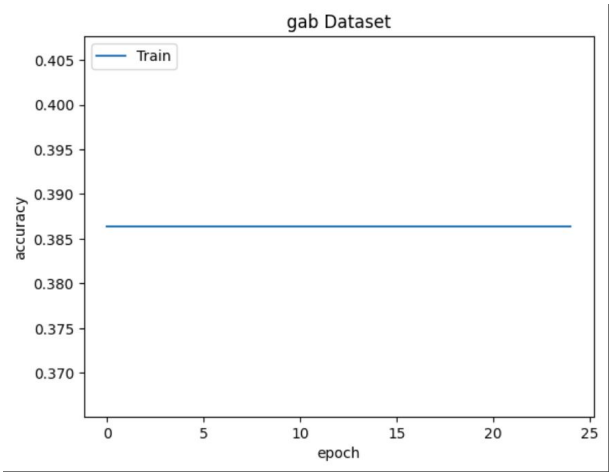

Figure 5. Accuracy For GAB data set

7 Limitations and Future work

To begin, let's talk about some of the restrictions. The methodologies that are discussed in this paper have some restrictions attached to them. Because of the data-driven nature of these techniques, their ability to detect toxicity is limited by the many forms of toxicity that were observed throughout the training process. Consequently, the models may be unable to recognize the toxicity associated with subjects or cultures that were not discussed throughout the training process. In addition, some data sets may introduce a bias into model prediction with regard to the use of particular phrases or

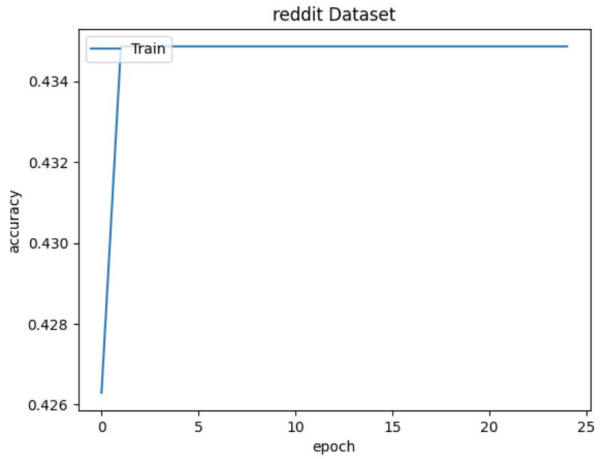

Figure 6. Accuracy for Reddit data set

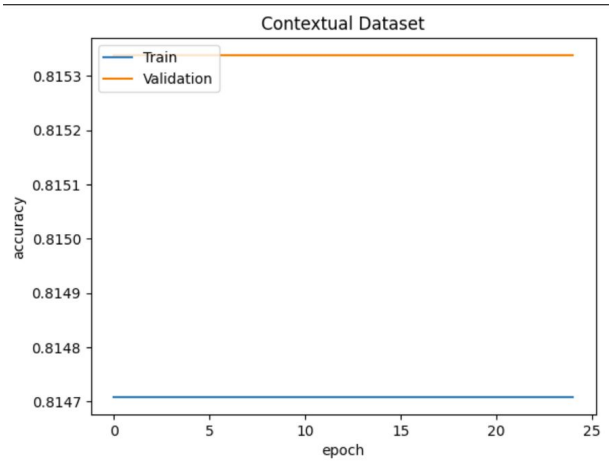

Figure 7. Accuracy for CAD data set

protected categories (for example, black people). As a consequence of this, a particular emphasis should be placed on the curation of data sets in order to guarantee that the models are free from bias and can identify a variety of forms of toxicity.

There is a wealth of model-related territory to cover for work that will be done in the future, but the primary deficiency is the availability of data sets. It is necessary for us to take into consideration data sets that are free of bias and include the maximum number of chats possible. One of the thoughts that came to mind was that if you consider a platform like Twitter, you can see that each post has comments and retweets attached to it. This is one of the ideas that we came up with. If you think of a post as the main variable represented by "x" and a comment as 'y' then you can easily solve the problem of contextual hate speech by simply assigning some mathematical scores to it. If x is positive and y is negative, then you can categorize 'y' as negative, and if

'x' is negative and 'y' is positive, then you can categorize 'y' as negative. This holds true in the opposite situation as well. Obviously, the architecture of Twitter has to incorporate an Auto-bot in order to recognize these comments. Despite the fact that we will need to construct a SOTA(State Of The Art) detector and train a model on effective data sets in order to accomplish this, even if that is conceivable, it is still a violation of privacy and should not be done.

References

https://github.com/dhfbk/twitter-abusive-context-dataset https://github.com/jing-qian/A-Benchmark-Dataset-for-Learning-to-Intervene-in-Online-Hate-Speech https://github.com/dongpng/cad_naacl2021

This project Report was done by using Overleaf https://www.overleaf.com/project/6451c56d0263368877454032