SMU Data Science Review

Volume 3 | Number 1 Article 13

2020

Toxic Comment Classification

Sara Zaheri
Southern Methodist University, szaheri@mail.smu.edu

Jeff Leath
Southern Methodist University, jleath@smu.edu

David Stroud
Southern Methodist University, jdstroud@smu.edu

Toxic Comment Classification

Sara Zaheri, Jeff Leath, David Stroud

Southern Methodist University, 6425 Boaz Lane,
Dallas, Texas 75205
{szaheri, jleath, jdstroud} @smu.edu

Abstract. This paper presents a novel application of Natural Language Processing techniques to classify unstructured text into toxic and non-toxic categories. In the current century, social media has created many job opportunities and, at the same time, it has become a unique place for people to freely express their opinions. Meanwhile, among these users, there are some groups that are taking advantage of this framework and misuse this freedom to implement their toxic mindset (i.e insulting, verbal sexual harassment, threads, Obscene, etc.). The 2017 Youth Risk Behavior Surveillance System (Centers for Disease Control and Prevention) estimated that 14.9% of high school students were electronically bullied in the 12 months, prior to the survey. The primary result could be an Open Source model used by app developers in support of anti-bullying efforts. The analysis of the results showed that LSTM has a 20% higher true positive rate than well-known Naïve Bayes method and this can be a big game changer in the field of comment classification. These results indicated that smart use of data science is able to form a healthier environment for virtual societies. Additionally, we improved our working pipeline and incorporated Amazon Web Service (AWS) as a fast, reliable and online platform to be able to run our classification algorithm. Our result showed a very promising accuracy of more than 70% performance by LSTM among all algorithms.

1 Introduction

The advances in IT technologies and generalizing virtualization all over the world has led to an unprecedented participation in social media; and there is no doubt that social media is one of the biggest hallmarks of the 21^{st} century. According to de Bruijn, Muhonen [1] ; social media has been growing exponentially since 2004. Meanwhile, social media is a place to express individual opinions and share thoughts in line with a constructive contribution to develop a safe place for everybody practicing their rights accordingly [2]. Based on the report by Birkland [3], 'Twitter users generate 500 million tweets per day, and in 2019 they had a 14% year-over-year growth of daily usage'. However, behind the shield of computers as virtual walls, some individuals also think they can abuse and harass other people's opinions and characters. Accordingly, a jargon word has been coined recently to address such behaviors as "cyberbullying". Based on U.S. Department of Health and Human Services, cyberbullying could be introduced as mistreatments that occurs all over digital instruments and devices such as computers, tablets, and mobile phones. Cyberbullying may take place via short massage system (SMS), general apps with the possibility of communication between users, online social media forums, or even online gaming where individuals can virtually participate in. Cyberbullying includes posting, sending, or sharing harmful, negative, mean, or false content about someone else either directly sent to the person or post as a general comment where other could observe it. It also includes sharing private or personal info about someone else in order to humiliation or embarrassment [4]. Such online harassment suppresses so many of our fellow citizens from expressing their opinions. According to the US Government, two separate recent studies have estimated that over 15% of teenagers have been victim to cyberbullying in the last 12 months. This includes common 'cyberbullying' tactics such as threatening to hurt someone, posting mean, hurtful, or embarrassing comments, and name calling. Related to significant impact of cyberbullying on people lives, Hinduja and Patchin [5] stated that their empirical study on some high-profile subjects have confirmed a link between suicidal ideation and experiences with cyberbullying offending or victimization. These studies and results related to this kind of online harassment leads to an important area of data science in order to be able to separate and distinguish harassment comments and cyberbullying, we call them toxic comments, from normal comments. According to HuffPost, such misbehavior's leaded to a reduction in digital activities. Many studies have also demonstrated that negative online interactions in social media can increase depression, lack of self-esteem, and even increase suicidal ideation [6,7,8,9,10,11,12].A research initiative founded by Jigsaw, and Google are currently working on tools to help improve online conversations. One significant aspect of their efforts is aiming to identify the toxic comments and lunch online toxicity monitoring system on various online social platforms. In a joint effort with Kaggle, they defined the project as a contest toxic comment classification challenge. The main goal of the challenge is developing a multi-label classifier, not only to identify the toxic comments but also detect the type of toxicity such as threats, obscenity, insults, and identity-based hate; while, there is still no practical tools that is able to distinguish these types of comments from each other with a very low rate of errors and a high accuracy. Automatic recognition of toxic contents in online forums, and social media is a useful provision for moderators of public platforms as well as users who could receive warnings and filter unwanted contents [2]. The need of advanced methods and techniques to improve identification of different types of comment posted online motivated the current study. As mentioned above, distinguishing improper comment that brings harassment to privacy and morality is one of the most crucial subjects related to the current extension of social media. With all the progress and improvement in IT and data science, the world is in requirement of a properly designed technique to find and isolate these kinds of comments that we call it toxic. Naïve Bayes is the most advanced method and algorithm currently used in this area; yet, it is not able to distinguish improper comments efficiently enough. Accordingly, aiming to find and develop an efficient algorithm to identify toxic comments, the current study implemented several algorithms, including Naïve Bayes (as a benchmark), Recurrent Neural Network (RNN), and Long Short Term Memory (LSTM). In order to achieve our goal in this study, related comments data of Crowd-sourcing (159,000 text comments) that was formerly labeled as seven categories were used.

Our contributions in this paper demonstrate that a practical data science solution is more than an algorithm, and that the scalability of the architecture must be considerate of the task at hand. First, we simplified the challenge to a binary classification whether the comment is toxic or not toxic. We also demonstrate that utilizing elastic cloud computing resources provides the scalability necessary to provide a practical solution. Finally, within this framework, we employed an advanced deep learning algorithm that provided a significant improvement in classification, versus our baseline modeling.

The remainder of the paper is organized as follows. Section 2 describes related work. Section 3 outlines the methodologies, and models. Section 4 describes data preparation. Section 5 discusses results and workflow. Section 6 provides a conclusion and discusses future works.

2 Related Work

Several studies have formerly investigated hate speech using neural network techniques; Badjatiya et al., used extensive experiments with multiple deep learning architectures to learn semantic word embedding to handle toxic comments identification [13]. In another study, sentiment analysis model of YouTube video comments, using a deep neural network was proposed that leaded to 70-80% accuracy [14]. Also, general use of different types of neural network methods for comment classification have been extensively used in recently published literature [6,15,16,17]; however, these approaches only addressed some of the task's challenges while others still remain unsolved. Furthermore, Farag, El-Seoud [18] reported that extensive numbers of literature have shown that supervised learning techniques have been the most frequently used methods for cyber-bullying detection. Nevertheless, other non-supervised techniques and methods have recognized to be operative on cyber-bullying recognition. Also, Karlekar and Bansal [19] reported an increased number of personal sexual harassment and abuse that are shared and posted online. In this study, authors presented the task of automatically categorizing and analyzing various forms of sexual harassment, based on stories shared on the online forum SafeCity and used labeling levels of groping, ogling, and commenting; their results indicated that single-label CNN-RNN model achieves an accuracy of 86.5.

One of the main undiscovered issues is how to identify algorithms that are able to implement high sensitivity in detection of toxic comments. Of course, identifying comment that is not toxic as a toxic can be frustrating for the users and there should be a lot of effort to form an algorithm with highest degree of sensitivities. In this paper, we implemented deep learning networks trained and tuned with the full capability of current Elastic Cloud Computing (EC2) infrastructure, supporting the MXNet framework deployment model. We applied LSTM/RNN (Long-short term memory/Recurrent neural networks), optimized with MXNet to harness the capability of multiple GPUs, as one of the very recent advances method in this field in order to learn the sequential relationship between choice of vocabulary; that causes the current paper to be categorized as a novel and practical work in the area of toxic comments.

3 Basic Description of the Solution

Based on the identified problems in the area of distinguishing toxic comments, the solutions that are assessed in the current study involved three primary components. First, a baseline classification solution using a Naïve Bayes [NB] approach was created. Subsequently; NB, as a powerful and widely recognized classifier, provided a firm benchmark for our additional work. Following this, we developed a classification solution framed around an LSTM/RNN algorithm. Since challenges with computational expense of a finely tuned LSTM/RNN solution was expected, the solution was scaled with an Elastic Compute Cloud [EC2] component. The AWS Sagemaker platform provided access to GPU capability which enables the full capability of Deep Neural Networks, such as an LSTM/RNN model.

3.1 Benchmark Model: Naive Bayes Approach

Naïve Bayesian classification [20] is a probabilistic approach to machine learning. It is based in the Bayes Theorem (Equation 1).

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)} \tag{1}$$

The probability of A happening knowing that B has occurred could be calculate. Event B represents the evidence and A the hypothesis to be approved. The theorem runs on the assumption that all predictors/features are independent and the presence of one would not affect the other. This is the Bayes approach naive simplification. The probability of one event, B, is independent from another B event occurring. The approach to classify an Internet comment as offensive or toxic would begin by study our collection of training data labeled as toxic and non-toxic. From the knowledge gained in the initial assessment of used data, the probability of whether that new comment is toxic or non-toxic was calculated. As an example,

$$P(\text{message is toxic}|\text{message content : ''Hi! I am back again!''}) \tag{2}$$

Applying the Bayes Theorem to calculate the probability that a new message is toxic according to the content of the message was then estimated by:

$$P(\text{message is toxic}|\text{message content}) = \frac{P(\text{message content}|\text{toxic})P(\text{Toxic})}{P(\text{message content})} \tag{3}$$

The probability that a message is toxic, P(Toxic), was then calculated based on the proportion of toxic messages occurred in training data set. As described in "Data Science in R" [20], there is no requirement to calculate the probability of the message content, P(message content), the probability that the new message's content is found in words pool that are toxic is required, based on the naive simplification of the Bayes Theorem. The naive assumption leads to calculate the likelihood of the new message being toxic as the product of the word appearing or not appearing in the toxic words pool. Likewise, the likelihood of the new message being non-toxic can be calculated, and the classification of the new message would depend on the likelihood ratio (Equation 4).

$$\frac{P(\text{message is toxic}|\text{message content})}{P(\text{message is non-toxic}|\text{message content})} \tag{4}$$

Additionally, further simplifying for this ratio could be achieved by a series of summations taking the log scale:

$$Log(\frac{P(\text{message is toxic}|\text{message content})}{P(\text{message is non-toxic}|\text{message content})}) \tag{5}$$

To avoid an obvious division by zero, a 0.5 scalar was added to the numerator and denominator.

$$Log(\frac{P(\text{message is toxic}|\text{message content}) + 0.5}{P(\text{message is non-toxic}|\text{message content}) + 0.5}) \tag{6}$$

3.2 LSTM/RNN Approach

Long Short Term Memory Recurrent Neural Network [LSTM-RNN], which is a class of deep, artificial neural networks [ANN/DNN], was applied to identify the toxic comments. DNN and its applications in NLP were found to be highly capable in higher order parameters which are very vital for a specific classification [21,22]. In this application, the higher order parameter can be a sequence of words which were labeled as a specific class. LSTM-RNN treats the comment as a set of vectorized words similar to a time series trying to learn how the words are aligned in a time series attributed to a specific label [23]. The performance of the models was tested against the benchmark model.

If a sentence or comment was treated like a time-series, the natural choice for would use RNN (Recurrent neural network). However, for long time series and more, its applications in NLP, LSTM is highly preferred over RNN. Due to chain rule and the fact that the inverse of activation functions is usually less than one, the early layers are pruned to be treated with very small portion of gradient descents. As a result, those early layers or deeper layers from the solution, were not trained. RNN (NLP tasks) treated the first, second, etc. words as earlier nodes as a result of Vanishing Gradient; those beginning words did not contribute as much as later words for NLP tasks. In order to solve this problem, LSTM was designed in a way to have a more capacity for remembering long term memories. The idea behind LSTM is in fact simple. Rather than each hidden node being simply a node with a single activation function, each node is a memory cell that can store other information. Specifically, it maintains its own cell state. Normal RNNs take in their previous hidden state and the current input, and output a new hidden state. An LSTM acts similarly, except it also takes in its old cell state and outputs its new cell state. LSTM has a way to control the flow of information through its hidden nodes. In other words, standard RNNs (Recurrent Neural Networks) suffer from vanishing and exploding gradient problems. LSTMs deal with these problems by introducing new gates, such as input and forget gates, which allows a better control over the gradient flow and enables a better preservation of long-range dependencies.

3.3 Metrics

In order to be able to evaluate the performance of each algorithm, several metrics are defined accordingly, and are discussed briefly in this section.

- **Confusion Matrix**: It is very informative performance measures for classification tasks. Ci,j an element of matrix tells how many of items with label i are classified as label j. Ideally we are looking for diagonal Confusion matrix where no item is miss-classified. The matrix in Figure 1 is a good representation for our binary classification. Positive (P) represents toxic label and n (negative) represents non-toxic label.
- **Accuracy**: This metric measures how many of the comments are labeled correctly. However, in our data set, where most of comments are not toxic, regardless of performance of model, a high accuracy was achieved (Equation 7.

$$Precision := \frac{TP + TN}{N' + P'} \tag{7}$$

- **Precision and Recall**: Precision and recall in were designed to measure the model performance in its ability to correctly classify the toxic comments. Precision explains what fraction of toxic classified comments are truly toxic, and Recall measures what fraction of toxic comments are labeled correctly (Equation 8).

$$Precision := \frac{TP}{P} \quad Recall := \frac{TP}{P'} \tag{8}$$

- **F Score**: Both Precision and Recall are important for checking the performance of the model. However, implementing a more advanced metric that combines both Precision and Recall together is quite informative and applicable (Equation 9). In this equation, setting $\beta = 1$ leads equation to return harmonic mean of Precision and Recall.

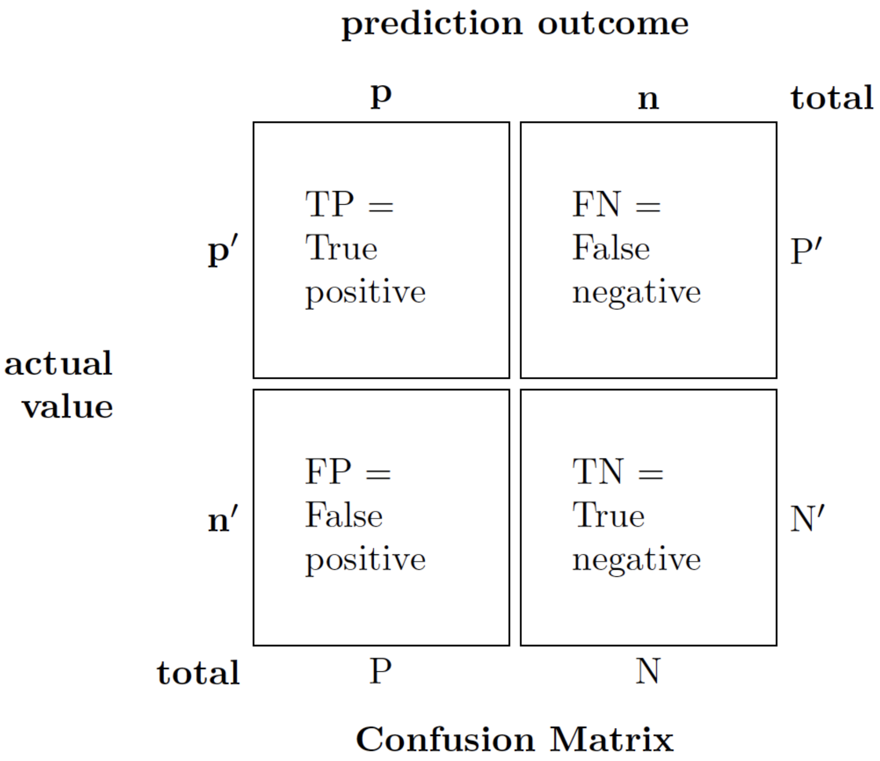


Fig. 1. Elements of confusion matrix; P (positive) represents toxic label and n (negative) represents non-toxic label.

$$F_{beta} = (1+\beta)^2 . \frac{Precision.recall}{(\beta^2.precision)+recall} \quad Recall := \frac{TP}{P'} \tag{9}$$

4 Data Preparation

One of the primary challenges in classification cases is having appropriately labeled data, in which a representative training set could be extracted for modeling. For text based or Natural Language Processing [NLP] problems, this quandary is even more pronounced [24]. As the sentiment inference of written communication is very subjective, there is little choice but to leverage human labeling in lieu of a proper analytical labeling model (hence the need for the classification algorithm). Although many large unlabeled text corpora are readily available, human labeled data is much more infrequent [25].

Fortunately, for various altruistic reasons, there are several organizations providing open sourced labeled data sets[1]. One such source is the WikiMedia Foundation[2] that offers access to text comment snippets from Wikipedia talk pages. Crowd-sourcing was used to manually label over 159,000 text comments, flagging each as one of seven following options: 1) Non-Toxic, 2) Toxic, 3) Severe Toxic, 4) Obscene, 5) Threat, 6) Insult, and 7) Identity Hate.

The data set is accessible via a closed Kaggle competition[3]. Each record in the data set is a textual user comment with six binary fields for each of the six toxicity levels, with null being non-toxic. The initial task was to reduce the seven labels down to two (toxic, non-toxic) categories. This generalization allowed us to focus on a more accurate binary classifier. In this sense, Non-toxic was taken as a category and all other toxic comments categorized as Toxic, according to Table 1.

Table 1. Transforming structure of different types of comments into a binary type for being used in the analyses accordingly.

Types of comments	Binary types of comments
Toxic Severe Toxic Obscene Threat Insult Identity Hate	Toxic
Other types of comments	Non-toxic

After finding and providing data for this study, preprocessing and cleaning the data were the first barriers which challenged this project. It is very vital to any NLP model in real world challenges. For example, ‘dont’, ‘do not’ and ’don’t’ have all similar meaning but without pre-processing they are considered as different tokens. The design of the pre-processing section is really task specific. In the current study, we assume that the verb tenses or similar words were not vital for being identified. Therefore, lemmatization was applied to reduce vocabulary to four forms of noun, verb, adjective, and adverb. Using the very simple intuitive methods our architectures were quite useful in NLP tasks. Regardless of final performance, they provide a proper intuition about the task.

5 Results

This section discusses our initial study, the results of the analysis, and a comparison between the algorithms. Below are the six subsections:

[1] www.wired.com/2016/09/inside-googles-internet-justice-league-ai-powered-war-trolls/

[2] meta.wikimedia.org/wiki/Research:Detox

[3] www.kaggle.com/c/jigsaw-toxic-comment-classification-challenge/data

• Data Exploration explains the structure of the data and how it's applied to our analysis.
• Workflow explains vectorization, embedding, and quantification of the words, along with their results.
• Scalable Architecture discusses the capacity of the platform used in the study and how it helped to imply the comparison between assessed algorithms.
Finally, three other essential subsections, including Benchmark Model, Naive Bayes Results, LSTM/RNN, and LSTM/RNN Versus Benchmark Model, are prepared. These three subsections, as the most parts of the result section, are presenting the final results of the algorithms implemented in the current study and make a proper comparison with reasons related to the algorithm that might be more appropriate for the goal of distinguishing toxic comments.

5.1 Data Exploration

Among 159,000 comments in the main database, there were about 8% of the cases that were labeled as toxic, while the remaining 92% were labeled as non-toxic. Since normally fewer comments are toxic in every general social media related data set [26], the distribution of classes of the data set used in this study is expected. Therefore, because the main goal here is to develop an algorithm that is more highly accurate than others using the same data set (or similarly structured data sets), this issue does not create a problem in comparing results to similar methods and algorithms.

In order to assess the frequency of words size being repeated in two different categories, the overall frequency over number of words was plotted Fig. 2 and Fig. 3. Interestingly, the toxic group were usually shorter than non-toxic and therefore it helped us to focus more on shorter sentences to improve algorithm performance.

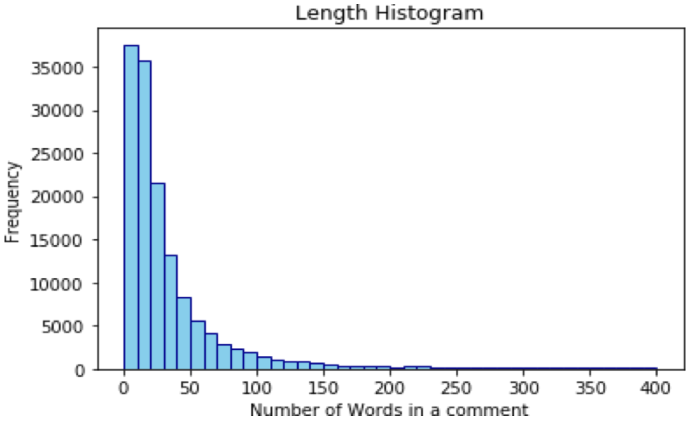
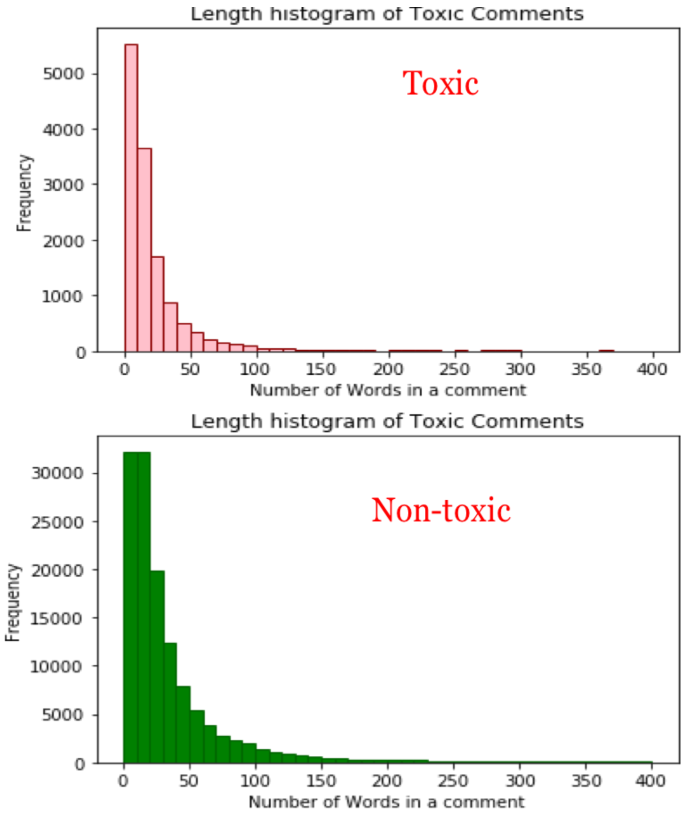


Fig. 2. Histogram representation the number of total words in comments.



Length histogram of Toxic Comments

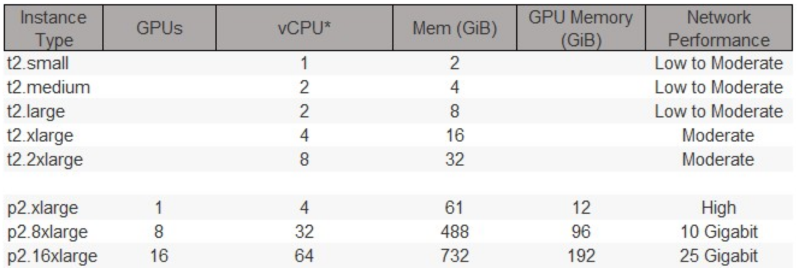
Length histogram of Toxic Comments

Fig. 3. Histogram representation the number of words in comments stratified by toxic and non-toxic.

5.2 Work Flow

Subsequent to the initial data preprocessing, the next step was vectorization and embedding the explanatory data. Here, every sentence was transformed into vectors of the same size. We heavily utilized the Natural Language Toolkit library (NLTK) in Python as this is one of the most well regarded packages for symbolic and statistical natural language processing[27]. After the final processing of the data, we applied several supervised learning algorithms including LSTM, RNN, and Naive Bayes. Their respective performances on the classification of the holdout data were subsequently compared to identify the best algorithm.

In order to make a proper comparison between the results obtained by both the Naive Bayes baseline and the LSTM-RNN models, the performance of each supervised learning method was represented as a confusion matrix for elucidating accuracy, sensitivity, specificity, and other statistical metrics.



Instance Type	GPUs	vCPU*	Mem (GiB)	GPU Memory (GiB)	Network Performance
t2.small		1	2		Low to Moderate
t2.medium		2	4		Low to Moderate
t2.large		2	8		Low to Moderate
t2.xlarge		4	16		Moderate
t2.2xlarge		8	32		Moderate
p2.xlarge	1	4	61	12	High
p2.8xlarge	8	32	488	96	10 Gigabit
p2.16xlarge	16	64	732	192	25 Gigabit

Fig. 4. This is a select representation of available instance types in AWS Sagemaker that allow selection of and access to varying levels of computational resources.

5.3 Scalable Architecture

From a processing standpoint, any data science solution must also be considerate of the processing load capacity of the platform being used. In support of our novel approach, with the intent of being able to provide a scalable and deployable solution, we also focused on utilizing an EC2 component (or Elastic Cloud Computing). EC2 is the ability to use cloud based computer resources, that can scale from the equivalent of single desktop CPU capability to multiple instances of GPUs portioning the workload for rapid, efficient and scaled processing[28]. In our case, even with a relatively small training data set, we struggled to process this intensive text vectorization workload, on a single CPU instance. Out of necessity, driven by computational expense, we moved our work to Amazon's AWS Sagemaker environment and leveraged the available GPU processing. For our solution, we deployed our models via AWS Sagemaker, using an instance type of p2.8xlarge. In any deployed solution, which could involve large, continuously evolving datasets, an EC2 component would absolutely be required[29].

Figure 4 is a select representation of available instance types in AWS Sagemaker that allow selection of and access to varying levels of computational resources[4].

5.4 Benchmark Model: Naive Bayes Results

As it was mentioned earlier in this study, Naive Bayes was selected the benchmark solution. Naive Bayes embedded in scikit-learn 0.20.0 in python was used. In order to achieve a precise comparison regarding models' performance on testing data-set, an extra confusion matrix along with F1 metric, as an optimum parameter for evaluation of the data-set, was finally implemented. according to the results of this section, accuracy was always above 90%, no matter what algorithm was used. This result may be due to the imbalance in the frequency of the data towards non-toxic comments. If one algorithm predicts all comments as non-toxic, since the numbers of toxic comments are only 8% of the data, accuracy still would be above 90%. It is worth mentioning that because of the same

[4] https://aws.amazon.com/ec2/instance-types/

data and the proportion in the data set used for all different algorithms, the imbalances in data would not affect the comparison results of these algorithms. In this case, little differences between accuracy of these algorithm would make a significant effect toward selection of the compared models. Fig. 5 presents the confusion matrix of Naive Bayes methods. It is clear that only 742 toxic comments (true positive) out of 1,543 (roughly 50%) toxic comments (total number of toxic comments) were identifiable by this algorithm.

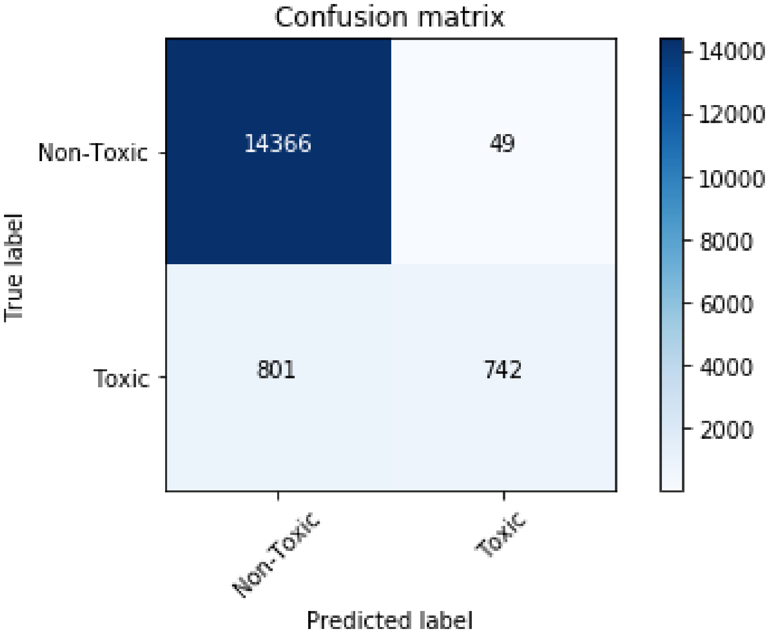


Fig. 5. Evaluation of Naïve Bayes algorithm using confusion metrics.

5.5 LSTM/RNN

LSTM, as a novel and new approach, was implemented for categorizing toxic data-set. similar to NB, the LSTM method was carried out by scikit-learn 0.20.0 in Python. For hyper-parameter, learning rate of 0.1, 128 vector dimensions, 5 epochs, and relu as activation function were implemented. The reason for implementing hyper-parameter was to make sure that hyper-tuning would not intensively limit the precise application of LSTM in this study. Fig. 6 represents the fact that 1,026 toxic comments (True positive) achieved by using LSTM approach, and only 517 comments was being missed by this method.

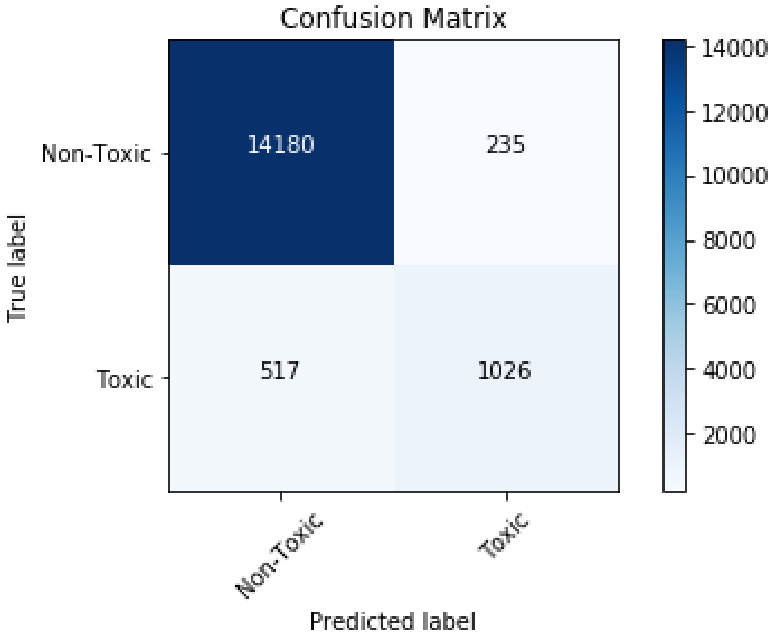


Fig. 6. Evaluation of LSTM algorithm using confusion metrics.

5.6 LSTM/RNN Versus Benchmark Model

Comparing the well-known method with LSTM approach (Table 2) reveals the fact that LSTM has an almost 20% increase in True Positive Rate (TRP). This represent the fact that among all identified toxic comment, LSTM has 20% more sensitivity in correct assignment of the comments. Also, Recall and F1 score also increases in LSTM comparing to Naive Bayes method. This data clearly explain the fact that LSTM can be a game changer in the field and it has significantly better performance over most accepted method in this field (Naive Bayes method).

Table 2. Summary figure that compares performance of each method (values are in percent).

Metric	Naïve Bayes (NB)	Long Short Term Memory (LSTM)
Total Positive Rate (TPR)	48	67
F1	64	73
Precision	94	81
Recall	48	66

6 Conclusions and Future Work

Our research has shown that harmful or toxic comments in the social media space have many negative impacts to society. The ability to readily and accurately identify comments as toxic could provide many benefits while mitigating the harm. Also, our research has shown the capability of readily available algorithms to be employed in such a way to address this challenge. In our specific study, it was demonstrated that an LSTM solution provides substantial improvement in classification versus a baseline Naïve Bayes based solution. To recall, the True Positive Rate of LSTM was almost 20% higher than Naïve Bayes method. Additionally, the followings are some suggested studies to be considered as future work in this area:

a) We suggest a plan to improve the NLP classifiers: first by using other algorithms which such as Support Vector Clustering (SVC) and Convolutional Neural Networks (CNN); secondly, extend the classifiers to the overall goal of Kaggle competition which is multi-label classifiers. in the current study, the problem simplified into two classes but it worth to pursue a main goal which is 7 classes of comments.

b) We also suggest using **SVM** for text processing and text classification. It requires a grid search for hyper-parameter tuning to get the best results.

c) using **Other DNN techniques (CNN))** because some recently published papers such as [34] have shown that CNN proves to have a very high performance for various NLP tasks.

References

1. Alissa de Bruijn, Vesa Muhonen, Tommaso Albinonistraat, Wan Fokkink, Peter Bloem, and Business Analytics. Detecting offensive language using transfer learning. 2019.
2. Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le. Xlnet: Generalized autoregressive pretraining for language understanding. In *Advances in neural information processing systems*, pages 5754–5764, 2019.
3. Thomas A Birkland. *An introduction to the policy process: Theories, concepts, and models of public policy making.* Routledge, 2019.
4. Estela Saquete, David Tomas, Paloma Moreda, Patricio Martinez-Barco, and Manuel Palomar. Fighting post-truth using natural language processing: A review and open challenges. *Expert Systems with Applications*, 141:112943, 2020.
5. Sameer Hinduja and Justin W Patchin. Bullying, cyberbullying, and suicide. *Archives of suicide research*, 14(3):206–221, 2010.
6. Saleem Alhabash, Anna R. McAlister, Chen Lou, and Amy Hagerstrom. From clicks to behaviors: The mediating effect of intentions to like, share, and comment

on the relationship between message evaluations and offline behavioral intentions. *Journal of Interactive Advertising*, 15(2):82–96, 2015.

7. Shannon D Bailey and Lina A Ricciardelli. Social comparisons, appearance related comments, contingent self-esteem and their relationships with body dissatisfaction and eating disturbance among women. *Eating behaviors*, 11(2):107–112, 2010.
8. Mark Hsueh, Kumar Yogeeswaran, and Sanna Malinen. Leave your comment below: Can biased online comments influence our own prejudicial attitudes and behaviors? *Human communication research*, 41(4):557–576, 2015.
9. Maria Koutamanis, Helen GM Vossen, and Patti M Valkenburg. Adolescents' comments in social media: Why do adolescents receive negative feedback and who is most at risk? *Computers in Human Behavior*, 53:486–494, 2015.
10. Moon J Lee and Jung Won Chun. Reading others' comments and public opinion poll results on social media: Social judgment and spiral of empowerment. *Computers in Human Behavior*, 65:479–487, 2016.
11. Hyejoon Rim and Doori Song. How negative becomes less negative: Understanding the effects of comment valence and response sidedness in social media. *Journal of Communication*, 66(3):475–495, 2016.
12. Leonie Rösner, Stephan Winter, and Nicole C Krämer. Dangerous minds? effects of uncivil online comments on aggressive cognitions, emotions, and behavior. *Computers in Human Behavior*, 58:461–470, 2016.
13. Pinkesh Badjatiya, Shashank Gupta, Manish Gupta, and Vasudeva Varma. Deep learning for hate speech detection in tweets. In *Proceedings of the 26th International Conference on World Wide Web Companion*, pages 759–760, 2017.
14. Alexandre Ashade Lassance Cunha, Melissa Carvalho Costa, and Marco Aurélio C Pacheco. Sentiment analysis of youtube video comments using deep neural networks. In *International Conference on Artificial Intelligence and Soft Computing*, pages 561–570. Springer, 2019.
15. Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, 5:135–146, 2017.
16. Pete Burnap and Matthew L Williams. Us and them: identifying cyber hate on twitter across multiple protected characteristics. *EPJ Data science*, 5(1):11, 2016.
17. Keita Kurita, Anna Belova, and Antonios Anastasopoulos. Towards robust toxic content classification. *arXiv preprint arXiv:1912.06872*, 2019.
18. Nadine Farag, Samir Abou El-Seoud, Gerard McKee, and Ghada Hassan. Bullying hurts: A survey on non-supervised techniques for cyber-bullying detection. In *Proceedings of the 2019 8th International Conference on Software and Information Engineering*, pages 85–90, 2019.
19. Sweta Karlekar and Mohit Bansal. Safecity: Understanding diverse forms of sexual harassment personal stories. *arXiv preprint arXiv:1809.04739*, 2018.
20. Deborah Nolan and Duncan Temple Lang. *Data Science in R: A Case Studies Approach to Computational Reasoning and Problem Solving*. CRC Press, 2015.
21. Nicola Michielli, U Rajendra Acharya, and Filippo Molinari. Cascaded lstm recurrent neural network for automated sleep stage classification using single-channel eeg signals. *Computers in biology and medicine*, 106:71–81, 2019.
22. Weicong Kong, Zhao Yang Dong, Youwei Jia, David J Hill, Yan Xu, and Yuan Zhang. Short-term residential load forecasting based on lstm recurrent neural network. *IEEE Transactions on Smart Grid*, 10(1):841–851, 2017.
23. Rui Yan, Yiping Song, and Hua Wu. Learning to respond with deep neural networks for retrieval-based human-computer conversation system. In *Proceedings of*

the 39th International ACM SIGIR conference on Research and Development in Information Retrieval, pages 55–64, 2016.

24. Nemanja Djuric, Jing Zhou, Robin Morris, Mihajlo Grbovic, Vladan Radosavljevic, and Narayan Bhamidipati. Hate speech detection with comment embeddings. In *Proceedings of the 24th international conference on world wide web*, pages 29–30, 2015.
25. Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. Improving language understanding by generative pre-training. *Amazonaws*, 2018.
26. Poorna Krishnamoorthy, Rory MacQueen, and Sebastian Schuster. Detecting insults in online comments. Technical report, Technical report, Stanford University. http://www. rorymacqueen. org/wp . . . , 2017.
27. Jacob Perkins. *Python text processing with NLTK 2.0 cookbook*. Packt Publishing Ltd, 2010.
28. Gustavo Portella, Genaina N Rodrigues, Eduardo Nakano, and Alba CMA Melo. Statistical analysis of amazon ec2 cloud pricing models. *Concurrency and Computation: Practice and Experience*, 31(18):e4451, 2019.
29. Edo Liberty, Zohar Karnin, Bing Xiang, Laurence Rouesnel, Baris Coskun, Ramesh Nallapati, Julio Delgado, Amir Sadoughi, Yury Astashonok, Piali Das, et al. Elastic machine learning algorithms in amazon sagemaker. Technical report, https://ssc.io/publication/ . . . , 2020.