Toxicity Detection in Texts

Muhammad Hamza, Christian Tabbah, Henry Lin, Travis Shao
University of Toronto

Abstract

Toxicity on the internet has become a growing issue with the rise of social media and online gaming. As a result, this project was created to combat this issue by detecting toxicity in such online chats. In the context of this project, toxicity refers to the overall negative connotation and sentiment that a sentence has. Moderation of negative comments can be expensive and time consuming, which this model hopes to help address. The approach taken will revolve around the Long Short-Term Memory (LSTM) Recurrent Neural Network (RNN). The architecture of this model utilizes LSTMs, which enable effective optimization of the loss and is further explored in this report. Due to the different meanings words have in different contexts, it captures contextual information before making a prediction. Finally, the model will determine a toxicity level that moderation teams can use to filter out messages and improve user experiences.

1 Introduction

The average teen now spends 4.1 to 5.8 hours on social media[1]. At the same time, teen suicide rates have risen by 29% over the last decade. Recently in Myanmar, hate groups were able to push out a regional minority thanks to Facebook's echo chamber algorithm and poor moderation[2]. Negativity in social media has invaded the daily lives of people, raised political tensions, and heightened a modern mental health crisis. The internet is flowing with toxicity, and scalable tools are needed to help mitigate this toxicity.

Given the current state of content moderation at social media and gaming companies, it is easy to understand why the systems either seem lax or overdone. Automated content moderation is employed, but is usually too lax or too aggressive to effectively moderate. Many companies outsource moderation as low paying-jobs, which realistically can never keep up with the current output of content. There are 510,000 Facebook comments made per minute alone presenting a need for effective automated moderation. This high financial barrier to entry also makes it harder for new companies to enter the market, making a less competitive market overall.

To combat this problem, a deep learning model will be designed to classify different types of toxic comments. While the model can not fully capture the full nuances of human conversation, the expectation is to cover a majority of blatant cases with a quick and flexible solution. Allowing human moderators to deal with the more complicated cases. Furthermore, the model uses a toxicity score ranging from zero to six this allows users to filter out comments based on toxicity. Oftentimes, in competitive games for instance, the discourse may have some messages of only a slightly negative connotation but contain important information. By having a toxicity level users can choose what level is acceptable and what needs to be removed. For instance, users may choose to ban levels that exceed four that contain profanity or target a user.

Submitted to 37th Conference on Neural Information Processing Systems (NeurIPS 2023). Do not distribute.

2 Background and Related Work

The background material for this model has been covered in CSC413, as it is building off of the idea of Sentiment analysis. Sentiment analysis refers to taking a sentence and classifying which feelings are conveyed in them or through them. The model takes the overall sentiment and classifies different levels of toxicity. The idea behind how the model will work is fundamentally the same as sentiment analysis, however the outcome is completely different, as the outcome is used differently. For example, measures of toxicity through the means of machine learning can be useful in most games as a way to censor toxic behavior in chats.

The data comes from a 2018 competition - the Kaggle Toxic Comment Classification Challenge. The models that came out of the competition (mostly RNNs) have done fairly well, with some projects having an F1 score as high as 0.973026 out of 1 (see: https://github.com/tianqwang/Toxic-Comment-Classification-Challenge). However the project obtained this accuracy using ensembling, which they themselves admitted was "extremely complicated." Additionally, not many papers tackled the problem using Long short-term memory (LSTM) models. With these factors in mind, the model will experiment with LSTMs to determine if there is a better accuracy and not just classification accuracy but also analyzing the true and false positives and negatives.

While previous studies have proposed detection models, they have not explored the innovative applications to pre-existing software. Something that is not being used on most websites, like Wikipedia, is comment censoring based on filtering. Some people would rather not see harmful/hateful comments on their favorite websites, which is why this paper will aim to create a tool to censor comments out based on filtering mechanisms, that actually use the model created. A tool like this could be used on any websites where people can publicly make comments. Furthermore, this project will add an in-depth analysis of what it means for the model to have predicted a comment to be "toxic", using analysis of false positives as well as gradients.

3 Data

3.1 Dataset

The dataset of choice is the following:

https://www.kaggle.com/datasets/fizzbuzz/cleaned-toxic-comments

This dataset provides 55.18 MB of Wikipedia comments which have been labeled by human raters for toxic behaviours. Note that this amounts to 300000 entries, 150000 training comments and 150000 test comments. They have been classified into the following categories: toxic, severe-toxic, obscene, threat, insult, identity-hate. This dataset was chosen for its variety of toxicity categories and size. The dataset is already split into a training set and a test set. Figure 1 shows how sample comments are classified. The main takeaway here is toxicity column which produces a score from zero to seven.

comment_	id	identity_h	insult	obscene	set	severe_to	threat	toxic	toxicity
explanati	000099793	0	0	0	train	0	0	0	0
d aww he	000103f0d	0	0	0	train	0	0	0	0
hey man i	000113f07	0	0	0	train	0	0	0	0
more i ca	0001b41b1	0	0	0	train	0	0	0	0
you sir ar	0001d958c	0	0	0	train	0	0	0	0
congratul	00025465c	0	0	0	train	0	0	0	0
[redacted]	0002bcb3c	0	1	1	train	1	0	1	4
your vand	00031b1e9	0	0	0	train	0	0	0	0
sorry if th	00037261f	0	0	0	train	0	0	0	0
alignment	00040093b	0	0	0	train	0	0	0	0
fair use r	000530008	0	0	0	train	0	0	0	0
bbq be a r	00054a5e1	0	0	0	train	0	0	0	0
hey what	0005c987b	0	0	0	train	0	0	1	1

Figure 1: Sample data entries

Figure 1 shows a table that has been classified into 1 hot vectors defining whether a comment is toxic. The blurred out comment was seen to be toxic and rather extreme, which is why it was blurred out in the screenshot. Having several categories of toxicity as well as a large amount of data is helpful, as it will hopefully allow the model to be further tuned to a high degree of precision, without overfitting.

3.2 Splitting Data

As for the data split, the dataset provided has already been split into 50% training and 50% testing. However, since the data is taken from kaggle the test set has no labels. As a result, the training set will be split as 70% training, 15% validation and 15% testing. Initially the chosen split for the model was 40% training, 10% validation and 50% testing. However, due to the hidden labels the provided test set was unable to be used to determine a test accuracy.

3.3 Other datasets

Several other datasets were considered in the research phase data, but they either had too little data, or far too many classes, to the point of raising ethical concerns (like having to deem something racist or homophobic).

The datasets that were explored:

1. https://www.kaggle.com/datasets/ashwiniyer176/toxic-tweets-dataset
2. https://www.kaggle.com/datasets/reihanenamdari/youtube-toxicity-data

4 Model Architecture

4.1 LSTM

The proposed architecture for this problem is an Long Short-Term Memory (LSTM) Recurrent Neural network (RNN) as shown in Figures 2 and 3. It will receive a sequence of inputs representing the toxic texts. Then output a classification for the level of toxicity as outlined by the data section. Before the sentence can be processed it needs to be tokenized into individual words and the corresponding GLoVe embeddings are computed. The GLoVe embeddings ensure that the dimension of the words is reduced to speed up training time. A context vector is inputted into the LSTM and carried through until the final context vector is computed, which is used to determine the overall toxicity of the text. This is done through an MLP layer which will output the final prediction.

The RNN design choice was made due to the sequential nature of the data. Each word on its own describes only part of the context however in sequence formulate the overall sentiment. This is much more effective compared to MLP models which are unable to capture these sequential dependencies. Furthermore, RNNs can assess the sentiment after each word as such can help detect which words contribute the most to the toxicity. These words specifically can be filtered out or shown to the sender to make them more aware. Moreover, the LSTM design choice was made to avoid exploding and vanishing gradients typically observed in vanilla RNN. The LSTM block adds additional gates that ensure that the gradient is being summed instead of being multiplied reducing the size of the gradient for large sequences (Figure 3). Lastly, the LSTM structure was used over GRU due to its greater capacity. It has more gates than GRU and is able to train longer sequences as a result.

Some hyper parameters that will need to be considered are the sequence length, batch size and any optimization hyper-parameters. For the sequence length the size of a typical sentence, which is around 20 words is a starting point. As for the batch size and the optimization parameters they vary heavily depending on the data that is being trained. As a result, all these values will be tuned on the validation set to ensure the most optimal accuracy.

4.2 Loss Function

Cross Entropy will be the loss function of choice for this model. The additive property of the target labels (larger classifications equates to a more toxic comment) allows for concrete classification of "distance" between a correct and incorrect classification. This means the typical numeric loss

functions can be used (mean square error, mean absolute error, etc.) of which Cross Entropy has traditionally performed the most problem-free.

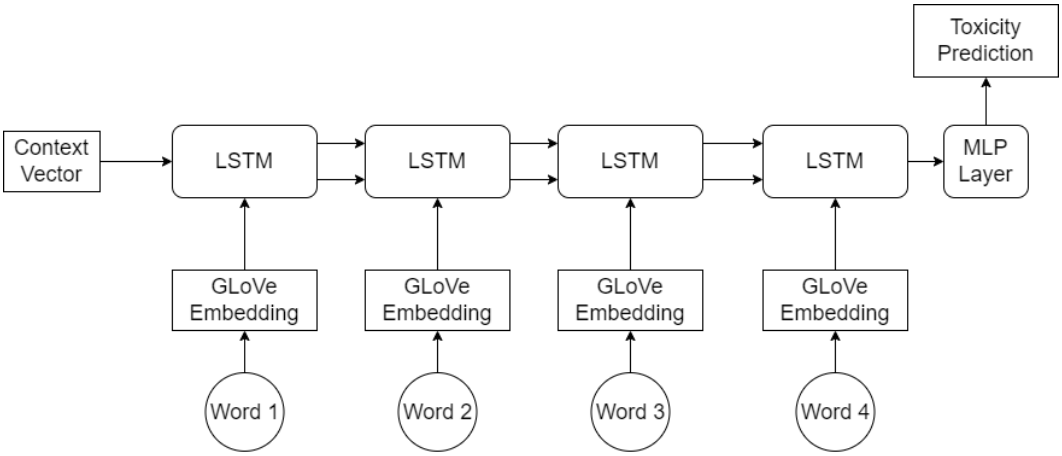

Figure 2: RNN Predication Architecture

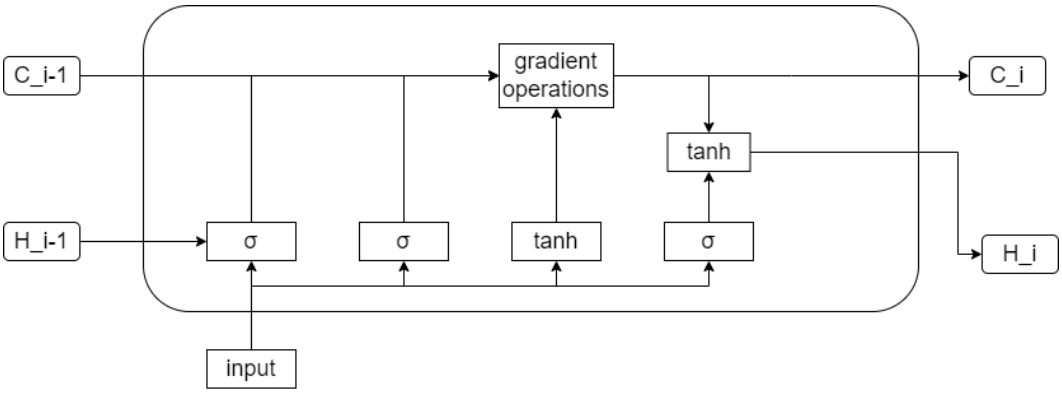

Figure 3: LSTM Block

5 Results

The LSTM model achieved a test accuracy of 88% on the data set. After training the model, the following confusion matrices were generated in Figures 5 and 6. The matrix in Figure 6 was generated by removing most of the non-toxic to give a better understanding of the toxic labels.

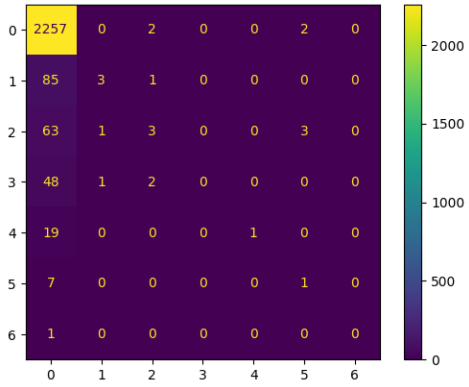

Figure 5: Confusion matrix of the first 2500 samples

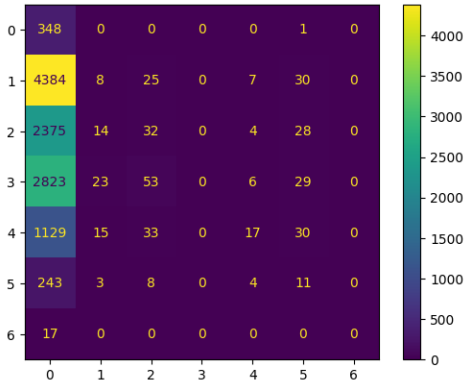

Figure 6: Confusion matrix of mostly toxic comments

After analyzing these matrices it becomes clear that many of the labels are simply zero, which enables the model to predict zero for all the inputs with seemingly reasonable accuracy. This leads to many false positives, severely impacting the model performance. As a way to prevent this data augmentation was used. Within the train data the toxic comments were slightly modified to generate similar sentences with the same toxicity level. This was achieved through determining the closest word from the GLoVe embeddings. After retraining the model the following confusion matrices were generated as seen in Figures 7 and 8.

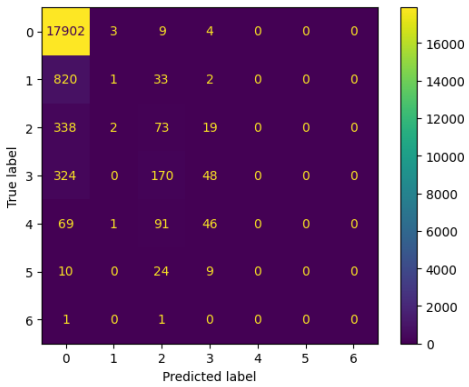

Figure 7: Confusion matrix of the first 2500 samples with augmentation

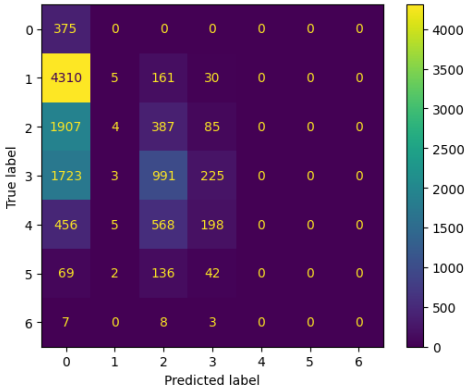

Figure 8: Confusion matrix of mostly toxic comments with augmentation

In addition, the training and loss curves with and without data augmentation can be analyzed. Figures 9 and 10 depict the training curves before the data augmentation. From Figure 9 it is evident that the LSTM model's validation accuracy starts decreasing as the training accuracy increases. This is an indication of overfitting, which may come as a result of having many more positive labels than toxic labels.

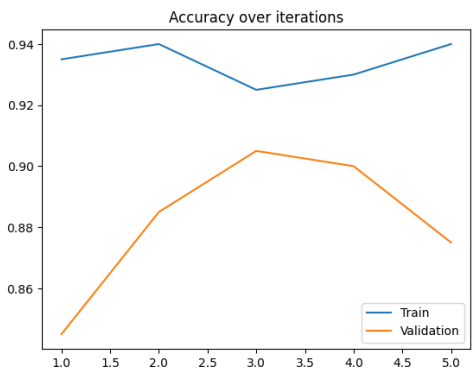

Figure 9: X-axis is iterations, Y-axis is Accuracy

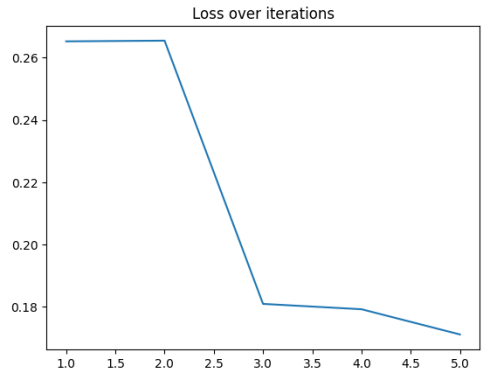

Figure 10: X-axis is iterations, Y-axis is Loss

After data augmentation, as seen in Figure 11 and 12, the validation curve matches the training much more closely. This is indication that the model is generalizing to patterns in the training set much more on the new augmented dataset. Augmenting the data provided some improvement as the accuracy increased by 1-2%. However, due to the limitations of the GLoVe embeddings the increase was subtle. Since GLoVe embeddings pick the closest word without any context the augmented data may not reflect actual texts that are sent. However, the model accuracy is still feasible when detecting toxicity in texts.

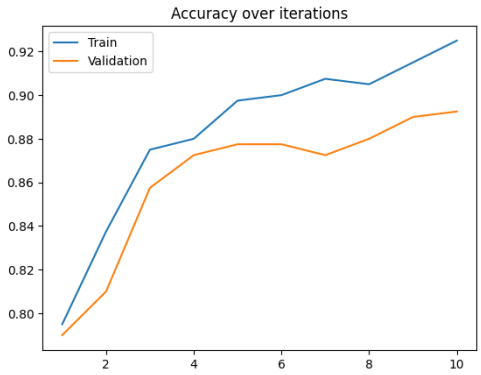

Figure 11: Accuracy with data augmentation. X-axis is iterations, Y-axis is Accuracy

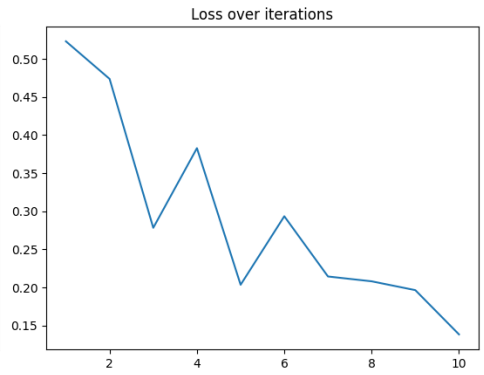

Figure 12: Loss with data augmentation. X-axis is iterations, Y-axis is Loss

6 Discussions

6.1 Practicality

The LSTM model is successfully able to determine the toxicity severity level of sentences it is provided with. Compared to the model provided in the CSC413 labs used to determine generically positive or negative Twitter comments, the LSTM model is able to differentiate the severity of the comments, significantly increasing its utility and real world applications. Compared to a model which has a binary classification, applications implementing the LSTM model are capable of discerning the severity. For example, users on some chat forum can be moderated relative to their message toxicity level, applying larger punishments for more toxic users. Although toxicity is still manually labelled within the toxic comments data set, this is a significant improvement over binary toxicity indicators.

The LSTM model is also able to successfully classify sentences based on its toxicity subcategories, such as being able to distinguish between a threat, an insult, identity hate, and obscenities or any combination of the above. This allows for filtering of just some combination of these subcategories. Users of this model will be able to filter specific a subcategory or subcategories of toxicity in comments, further increasing the usability.

6.2 Data Augmentation

Data augmentation proved vital in increasing the accuracy to 90+%. A large reason for this was the disproportionate amount of non-toxic to toxic comments (almost 9 to 1). While removing non-toxic comments was one way to balance out this ratio, the results depicted that the accuracy was severely impacted. This is due to the LSTM model requiring large amounts of data to make meaningful predictions and generalize.

Differing levels or types of data augmentation may be especially useful for improving the robustness of the model, as wikipedia talk page discussions are usually more structured and topical compared to youtube comments or tweets from twitter. Discourse on those sites are often less grammatically correct and harder to gauge context from.

6.3 Further Avenues for Improvement

Beyond accuracy, there are other potential ways to improve the robustness of this LSTM model. Identifying which sections of a text are toxic is a great way to improve the transparency of the model. It could also open up avenues of better feedback for generative adversarial networks. Theoretically, this could be done with the same dataset used, but would likely require a more complex model. This is out of the scope of this experiment, which focused on using rudimentary techniques to obtain the best results possible.

7 Limitations

7.1 GloVE and Other Pre-Trained Embeddings

The LSTM model achieved a final validation accuracy of over 90%. This puts it on par with other submissions to the Toxic Comment Classification Challenge in 2018 (where the dataset originates from). However, the top models largely depend on pre-trained embeddings to do the bulk of the work. The winner of the competition even claimed that off the pre-trained embeddings alone their model was able to have a validation accuracy of 98.77%[5]. They used several other techniques that improved the LSTM model's accuracy as well, including translations as train/test-time augmentation, rough-bore pseudo-labelling, and a robust CV and stacking framework. The techniques mentioned, along with training a new embedding, are out of the scope of this project. But given the performance of the LSTM model despite the rudimentary techniques used to prepare the data for the LSTM, it is a promising sign of its power and adaptability.

7.2 Subcategory Severity

The LSTM model used pretrained GloVE embedding weights, and was thus able to achieve a training and validation accuracy within a small number of training iterations even with a small batch size. Although it is able to discern severity levels of toxicity, due to the dataset's collection of binary toxicity subcategory binary classifications, the LSTM model is unable to discern the severity of these subcategories. For example, if a sentence has a personal threat, the severity of that threat can determine whether one sentence is more toxic than the other.

However, for this purpose the model's capacity is limited by the chosen data set. The dataset only has binary indicators for these subcategories, thus it becomes difficult to train a model on this data set for the purpose of making these indicators no longer binary. This could theoretically be done by initially training a binary classifier on each subcategory of toxicity and using such a model to artificially label each sentence giving it a toxicity level with respect to the subcategory. However, such a model would be severely dependent on each sub model for each subcategory.

8 Ethical Considerations

8.1 Freedom of Expression

There are many potential ethical issues posed by the use or misuse of the model such as over-reliance on the model for judgement and biases in determining toxicity. Using this model to judge and punish

user toxicity on social media and gaming platforms may be considered a restriction on free speech. Another issue is if this model is intentionally maliciously retrained to specifically target certain groups, organizations, and ideas for censorship. Such a model used to monitor online activity on forums, chat, and discussion pages could be considered harmful depending on the context of its usage.

8.2 Overreliance and Independence

The threshold for determining toxicity within an isolated statement is also arbitrary and society and culture dependent, thus classification labels from any data set are inherently biased towards and against certain cultures and groups due to the nature of toxicity. The attributes which determine a toxic statement may be biased socially, politically, and culturally. There also lies an issue in over-reliance on these tools. As stated in the introduction, the purpose of this model is to assist but not replace human moderators. In the case of bad decisions or difficult cases, there needs to be a person that is accountable, to look over and rectify these decisions.

8.3 Keeping models up to date

It is worth noting that human languages change over time. Phrases that were once acceptable can be perceived as being toxic moving forward. In contrast some types of behaviour may become more accepted in society. As a result, the model must be fine tuned as time progresses due to the shifting notions of what it means to be toxic.

9 Conclusion

The LSTM model created in this paper is successfully able to determine the relative toxicity level of comments and in which manner they are toxic in. While not the most accurate model, given the rudimentary techniques used, it provides quick auto moderation. The model was tested on a few sample words and sentences and produced a reasonable toxicity score. Phrases such as "I am doing great" produce the expected score of zero, whereas the word "ugly" produces the score of three. Furthermore, adding profanity further increases the score to five or six. As a result, the detector is able to filter out toxicity in layers, where the user can choose how much toxicity is acceptable and how much needs to be filtered out. This shows the promising power and flexibility of LSTMs in the moderation and sentence classification space that ought to be further explored.

References

[1] Rothwell, J. (2023, October 31). Teens spend average of 4.8 hours on social media per day. Gallup.com. https://news.gallup.com/poll/512576/teens-spend-average-hours-social-media-per-day.aspx: :text=This

[2] Amnesty International. (2022, September). Myanmar: Facebook's systems promoted violence against Rohingya - Meta owes reparations, new report. Retrieved from https://www.amnesty.org/en/latest/news/2022/09/myanmar-facebooks-systems-promoted-violence-against-rohingya-meta-owes-reparations-new-report/

[3] Patel, Alpna and Tiwari, Arvind Kumar, Sentiment Analysis by using Recurrent Neural Network (February 8, 2019). Proceedings of 2nd International Conference on Advanced Computing and Software Engineering (ICACSE) 2019, Available at SSRN: https://ssrn.com/abstract=3349572 or http://dx.doi.org/10.2139/ssrn.3349572

[4] S. Yang, X. Yu and Y. Zhou, "LSTM and GRU Neural Network Performance Comparison Study: Taking Yelp Review Dataset as an Example," 2020 International Workshop on Electronic Communication and Artificial Intelligence (IWECAI), Shanghai, China, 2020, pp. 98-101, doi: 10.1109/IWECAI50956.2020.00027.

[5] Lee, Chun Ming. "Toxic Comment Classification Challenge." Kaggle, www.kaggle.com/competitions/jigsaw-toxic-comment-classification-challenge/discussion/52557. Accessed 12 Dec. 2023.