Toxic Comment Classification
ECE 324 Term Project Final Report

Word count: 1995

Team Members:
Anton Liu 1005102407
Yue Zhuang 1005087781
Grace Wu 1005183356

Permission to post video: yes
Permission to post final report: yes
Permission to post source code: yes

Introduction

With the ongoing pandemic, there are more people online than ever. In fact, 84% of young Canadians actively rely on social media such as Facebook to communicate with others [1]. However, the internet also gives negative individuals "a safe space to explore extreme ideologies and intensify their hate without consequence ... until it explodes in the real world", said Shannon Martinez from the Free Radicals Project [2].

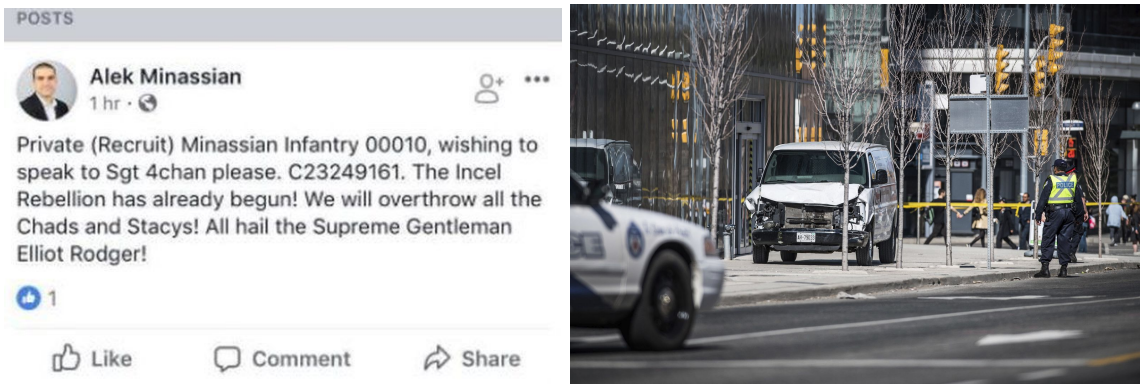


Figure 1: Alek Minassian, accused in the Toronto van attack, praised “incel rebellion.” on Facebook [3]

The goal of this project is to create a safer environment online. By detecting toxic comments on social media, they can be easily reported and removed. In the long term, this would allow people to better connect with each other in this increasingly digital world.

This problem can be tackled best with a neural network because it is hard for hard-coded algorithms to understand and detect human speech, and costly for companies to hire humans to identify these comments. Therefore, NLP with neural networks has been found to be the most effective way to do so in the industry, with an estimated market size of US$ 26.4 billion by 2024 [4].

Background

Before deep learning (NLP), companies resorted to ineffective methods of identifying hate speech, such as simple keyword searches (bag of word). This method has “high recall but leads to high rates of false positives” [5], mistakenly removing normal conversation.

Recently, research has already been conducted in the deep learning field to identify hate speech. A paper published in August 2019 used multiple-view stacked Support Vector Machine (mSVM) to achieve approximately 80% accuracy with data from various social media companies [6]. Another paper published in 2018 utilizes various word embeddings to train a CNN_GRU model, achieving 90% accuracy on 3 different classes [7]

In addition, many social media companies have invested in methods to eliminate online hate speech. In July 2020, Facebook Canada announced that it is "teaming up with Ontario Tech University's Centre on

Hate, Bias and Extremism to create what it calls the Global Network Against Hate" [8], for which Facebook will invest $500,000 to spot online extremism and countering methods.

Illustration

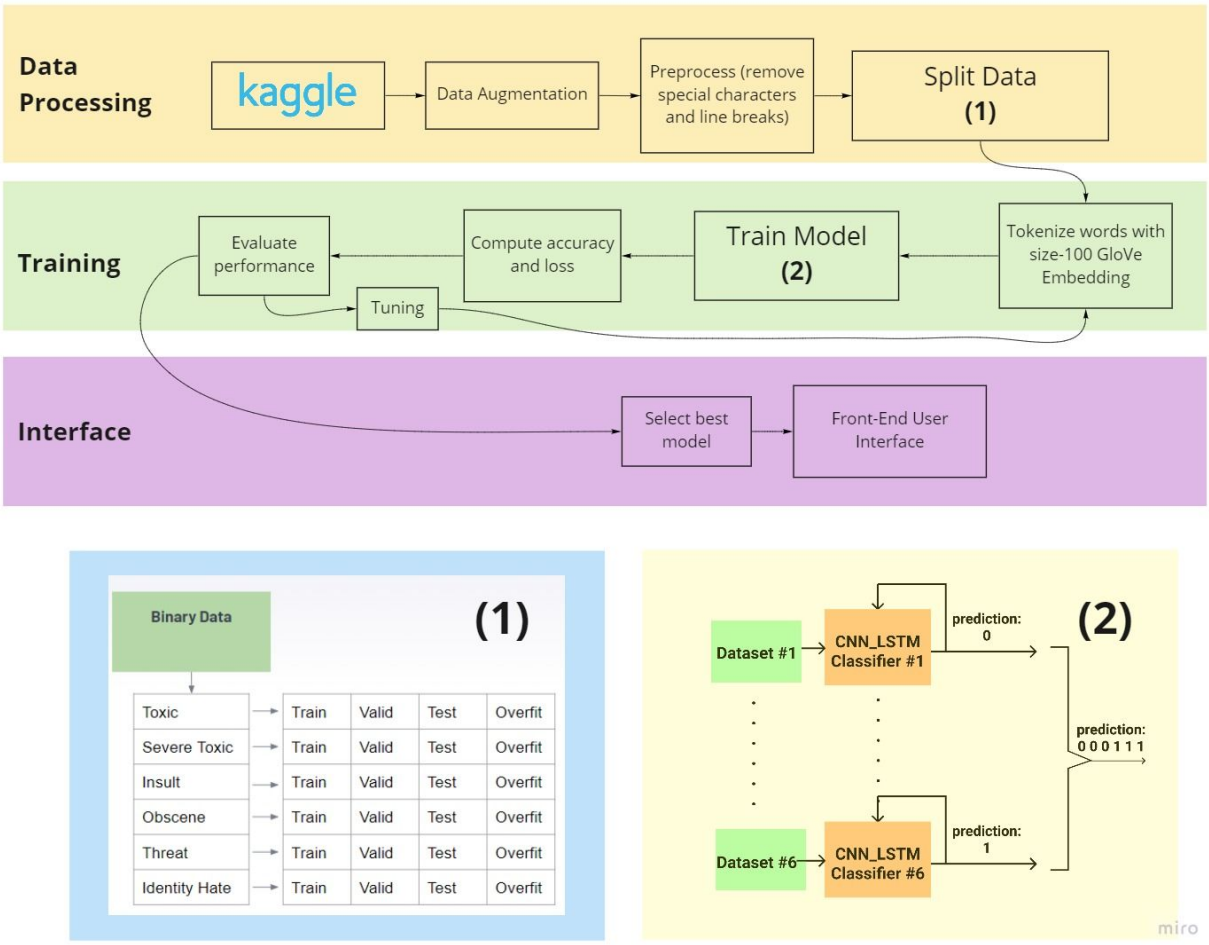


Figure 2: Model Production Pipeline

Data source, labelling, and processing

Data Source

We used data from the "Toxic Comment Classification Challenge" on Kaggle, which contains comments from Wikipedia editors labelled by human volunteers individually [9]. Comments are labelled in 6 categories: toxic, severe toxic, obscene, threat, insult, and identity hate. For example, if a comment is labelled as 100100, it is toxic and threat. However, we found that the labels are not accurate for some of the comments, which we manually relabelled.

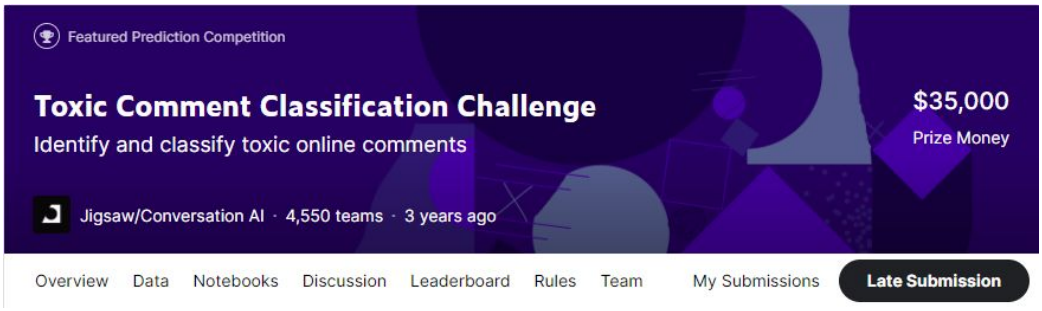


Figure 3: Title page of Kaggle Challenge

Examine data

Next, we examined the data using pandas. From **Figure 4**, we can see that there are 150000+ comments total. However, every class has a very low mean, which means that most of the comments are good comments. Therefore we need to balance the good and bad comments in the processing step.

	toxic	severe_toxic	...	insult	identity_hate
count	159571.000000	159571.000000	...	159571.000000	159571.000000
mean	0.095844	0.009996	...	0.049364	0.008805

Figure 4: the count and mean of the classes as output of .describe()

We also realized that some classes do not have enough comments. There are 15000+ comments labelled as toxic but less than 500 are threat. With such low numbers of comments in certain classes, it is easy for the models to overfit and generate a low accuracy for the validation and test set.

obscene	8449
insult	7877
toxic	15294
severe_toxic	1595
identity_hate	1405
threat	478

Figure 5: Number of comments in each class

Data Augmentation

To solve the issue above, we used the package nlpaug to artificially increase the amount of data in the minority classes by substituting synonyms [10]. Because the comments are multi-labelled, the amount of comments in the majority classes also increased.

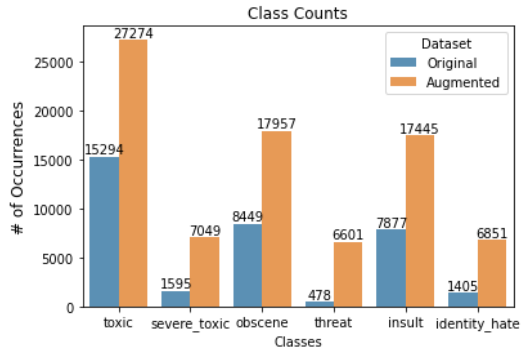


Figure 6: Amount of comments in each class before and after augmentation

```
[82] aug1 = naw.ContextualWordEmbsAug(
         model_path='distilbert-base-uncased', action="substitute")
     aug2 = naw.ContextualWordEmbsAug(
         model_path='roberta-base', action="substitute")
     aug3 = naw.SynonymAug(aug_src='wordnet')

     augmented_text1 = aug1.augment(text)
     augmented_text2 = aug2.augment(text)
     augmented_text3 = aug3.augment(text)
     print("Original: " + text)
     print("\nAugmented Text 1: " + augmented_text1)
     print("\nAugmented Text 2: " + augmented_text2)
     print("\nAugmented Text 3: " + augmented_text3)

Original: The quick brown fox jumps over the lazy dog

Augmented Text 1:the striped brown fox jumps over the muddy grass

Augmented Text 2:The quick brown fox jumps Into the bull dog

Augmented Text 3:The quick wild fox hurdles over the lazy dog
```

Figure 7: Example of augmented sentence

Data Cleaning

We cleaned the data using regex, matching patterns in the comments and replacing them with more organized counterparts. We removed any spaces, line breaks, contractions, etc. Cleaner data leads to a more efficient model and higher accuracy.

```
"id","comment_text","toxic","severe_toxic","obscene","threat","insult","identity_hate"
"0000997932d777bf","Explanation
Why the edits made under my username Hardcore Metallica Fan were reverted? They weren't vandalisms, just closur
"000103f0d9cfb60f","D'aww! He matches this background colour I'm seemingly stuck with. Thanks.  (talk) 21:51, J
"000113f07ec002fd","Hey man, I'm really not trying to edit war. It's just that this guy is constantly removing
"0001b41b1c6bb37e","""
More
I can't make any real suggestions on improvement - I wondered if the section statistics should be later on, or

There appears to be a backlog on articles for review so I guess there may be a delay until a reviewer turns up.
"0001d958c54c6e35","You, sir, are my hero. Any chance you remember what page that's on?",0,0,0,0,0,0
"00025465d4725e87","""

Congratulations from me as well, use the tools well.  · talk """,0,0,0,0,0,0
"0002bcb3da6cb337","COCKSUCKER BEFORE YOU PISS AROUND ON MY WORK",1,1,1,0,1,0
```

```
id,comment_text,toxic,severe_toxic,obscene,threat,insult,identity_hate
0000997932d777bf,explanation why the edits made under my username hardcore metallica fan were reverted they were
000103f0d9cfb60f,d aww he matches this background colour i am seemingly stuck with thanks talk 21 51 january 11 2
000113f07ec002fd,hey man i am really not trying to edit war it just that this guy is constantly removing relevant
0001b41b1c6bb37e,more i cannot make any real suggestions on improvement i wondered if the section statistics shou
0001d958c54c6e35,you sir are my hero any chance you remember what page that on,0,0,0,0,0,0
00025465d4725e87,congratulations from me as well use the tools well talk,0,0,0,0,0,0
0002bcb3da6cb337,cocksucker before you piss around on my work,1,1,1,0,1,0
```

Figure 8: The same comments before (top) and after (bottom) cleaning

Data Processing

For prototyping and testing models, 10+ datasets were created. In the final dataset, we created 6 different sub-datasets (one for each class) and each sub-dataset is split into training, validation, testing, and overfit sets. Each sub-dataset takes all the bad comments labelled as a certain class, and balances them with an equal number of good comments.

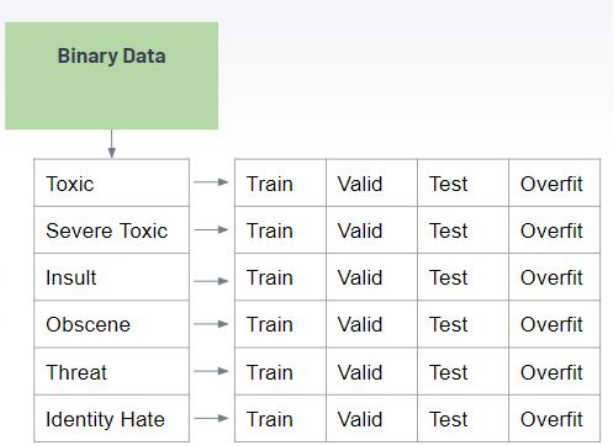


Figure 9: Visualization of dataset structure

Architecture

To achieve multi-label classification, the team used 6 CNN_LSTM Binary Classification Model. The input data was first converted into word vectors with GloVe Embedding. We identified that bad comments share similar sets of vocabularies. Thus two convolution layers were used to identify word patterns in sentences regardless of their position. To handle the change in sentence length, each convolution layer has one maxpool layer that gives one output feature for each sentence from each kernel. Then the outputs were concatenated to a vector and were fed to a LSTM (Long Short Term Memory network), which is a special kind of RNN capable of learning long dependencies [11]. The output of the LSTM was decoded using a fully connected layer.

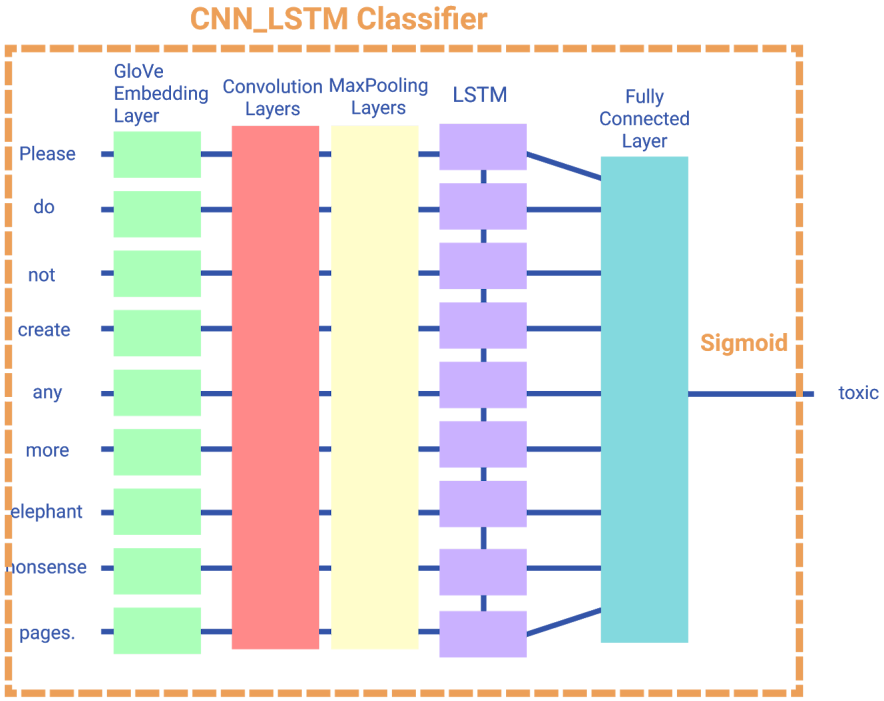


Figure 10: Model Architecture

This project focuses on multi-labelling classification since the comment labels are multi-hot encoded. We adopted the training approach shown in **Fig 10** to ensure each classifier is fully trained. 6 balanced binary datasets each having data from a class were used to train a CNN_LSTM binary classifier. The binary classifier outputs were combined to produce a multi-hot encoded final prediction on each comment.

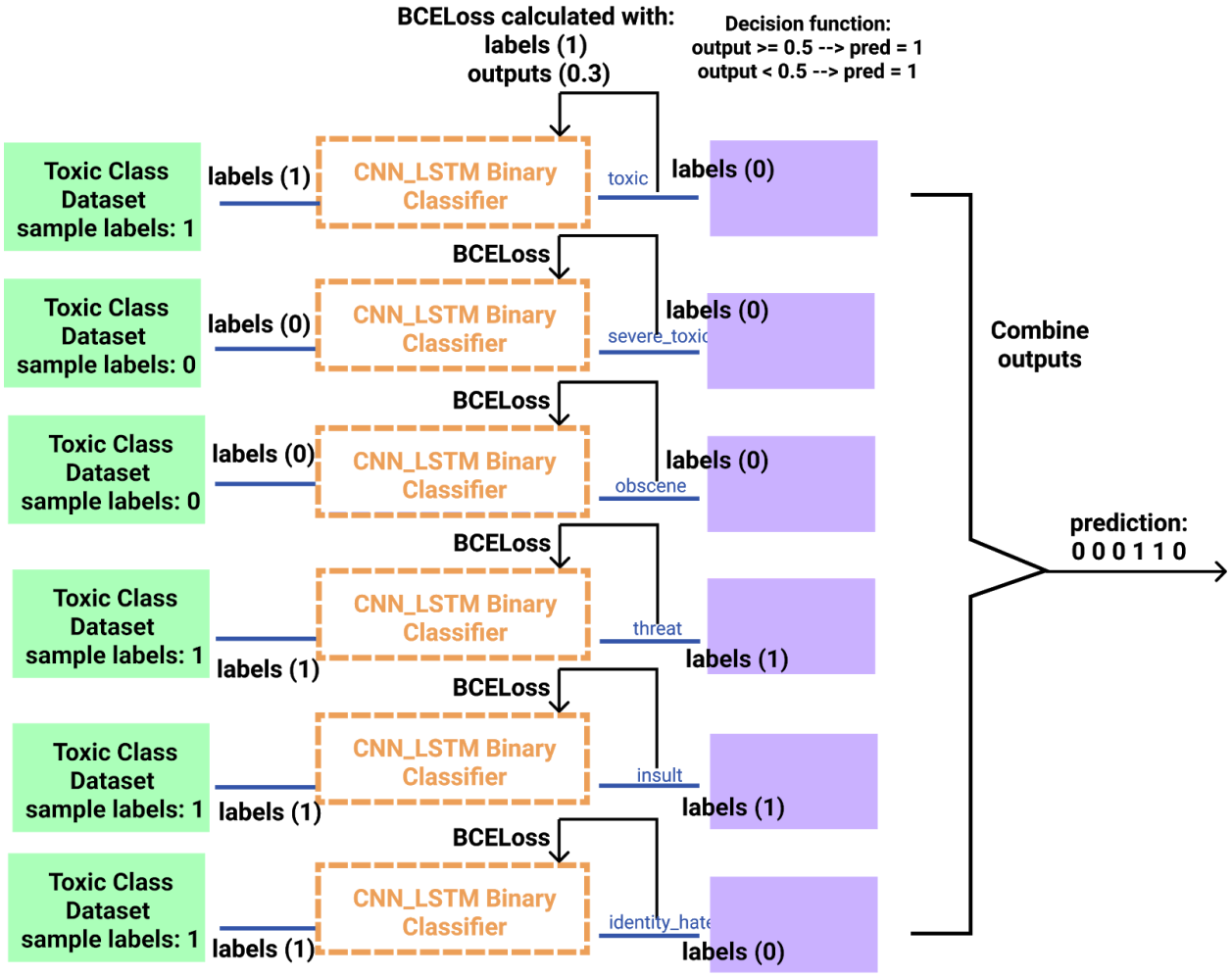


Figure 11: Best Training Approach

Baseline Model

For our baseline model, we obtain the glove embedding vectors of each word in each comment, average their values, and pass this average vector through a fully connected layer and generate an output prediction. As comments are generally short and are composed of few sentences, an average vector should be sufficient for the model to abstract some sense of toxicity.

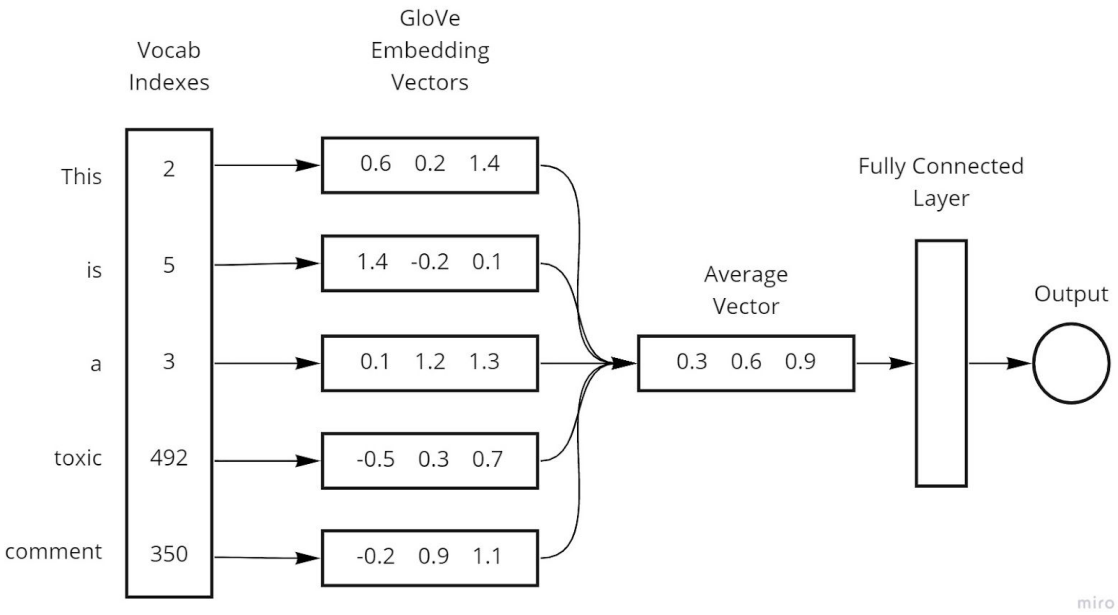


Figure 12: Baseline Model Pipeline

Quantitative Results

The best performing baseline model has a training loss of 0.617, accuracy of 79.7%, testing loss of 0.619, and accuracy of 79.4%. This suggested that our project idea is feasible.

The best model was trained with hyperparameters in **Table 1**. The combined loss and accuracy are calculated as the average loss and accuracy among the 6 binary classifiers. The best model has a training loss of 0.510, accuracy of 98.6%, testing loss of 0.528, and accuracy of 95.3%. As **Fig 13** shows, the curves are smooth and show exponential trends. No noticeable sign of overfitting is observed. The model performance reached a relatively steady state at epoch 10, however, the best training and testing accuracy were obtained at epoch 30.

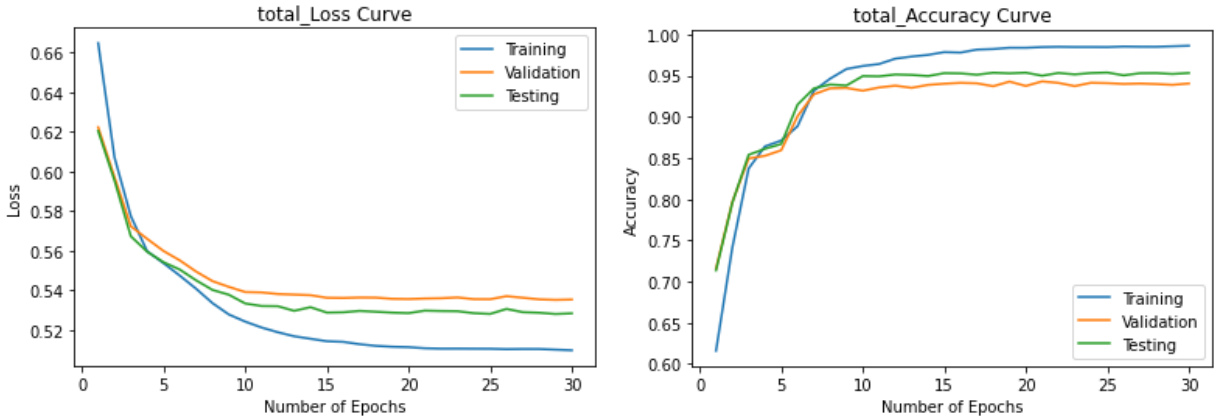


Figure 13. Loss and Accuracy Curve

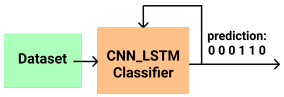
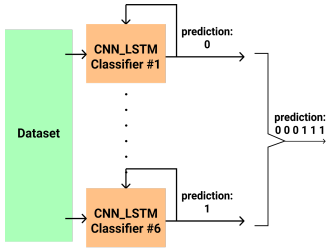
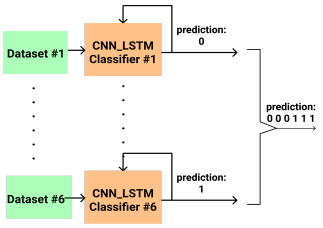
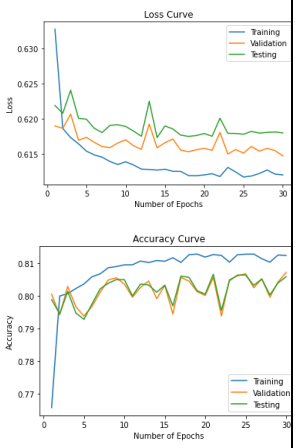
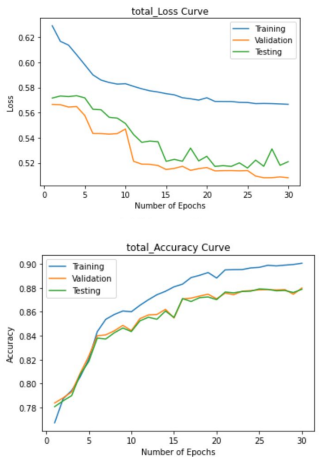
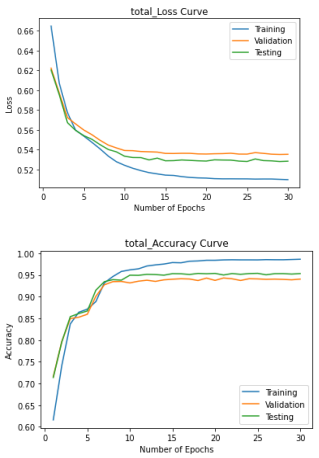
Table 1. Best Model Hyperparameters

CNN_LSTM Binary Classifier on Best Training Approach			
hidden_dim	100	# maxpool layer	2
embedding_dim	100	# conv2d layer	2
kernel_size	(2,100)	# LSTM	1
# kernels	50	# fc layer	1
batch_size	128	learning_rate	0.001
epochs	30	decision function	If output >= 0.5, prediction = 1 If output < 0.5, Prediction = 0

To ensure the best training approach, we trained the same CNN_LSTM classifier in the three approaches in **Table 2.** The first two approaches were trained using a general and unbalanced dataset. In approach 3, models were trained with 6 separated balanced datasets. As the accuracy and loss data show, the third approach, which is the approach used in the best model, performs the best. The curves in the other two approaches fluctuate and have evidence of overfitting.

Table 2. Three Training Approaches

	Approach 1	**Approach 2**	**Approach 3 (best)**
# classes in each dataset	**6**	**6**	**1**
Is # positive/negative labels in each class balanced?	**No**	**No**	**Yes**
# CNN_LSTM model	**1**	**6**	**6**
# Model Output	**6**	**1**	**1**
Training Loss	**0.612**	**0.612**	**0.510**
Training Accuracy	**81.2%**	**90.0%**	**98.6%**
Testing Loss	**0.618**	**0.537**	**0.528**

Testing Accuracy	**80.6%**	**87.9%**	**95.3%**
Model Architecture			
Loss and Accuracy Graphs			

In the multi-label classification problem, the number of FP and the F1 score of the model should also be evaluated. **Table 3** shows the confusion matrices of the 6 model classes tested with the balanced binary datasets. There are much more TP, TN than FP and FN. Moreover, the fact that most classes have fewer FP than FN proves that our best model gives desirable results. The F1 score for each class is higher than 90% thus showing that our classifying model has high precision and recall.

By comparing the F1 scores and the results shown in the confusion matrices, we notice that class toxic, has a relatively lower F1 score. It also gives more false predictions. This is due to inconsistency on the labelling of toxic comments. Besides, most bad comments in the dataset are labelled as toxic, so our classifiers have difficulty picking up the features specific to the toxic comments.

Table 3. Confusion Matrices and F1 Scores

Class name	**Toxic**	**Severe_toxic**	**Obscene**
TP / FP FN / TN	2454 / 173 274 / 2555	686 / 11 19 / 694	794 / 33 51 / 812

F1 score	0.9165	0.9785	0.9486
Class name	**Threat**	**Insult**	**Identity_hate**
TP: 642, FP: 20 FN: 19, TN: 641		TP: 1652, FP: 57 FN: 93, TN: 1688	TP: 649, FP: 25 FN: 37, TN: 661
F1 score	0.9705	0.9566	0.9544

Qualitative Results

Table 4. Sample Model Outputs, Predictions, and Actual Labels

Case	**Comment**	**Value Type**	**Toxic**	**Severe_toxic**	**Obscene**	**Threat**	**Insult**	**Identity_hate**
1	"Well that explains it thanks for clearing it up"	**Model Output**	3.3293e-06	0.0003	2.5801e-05	0.0004	1.1351e-05	0.0002
		Prediction	0	0	0	0	0	0
		Actual Label	0	0	0	0	0	0
2	"You are ugly I hate you."	**Model Output**	1.0000	0.9993	0.9999	0.9970	1.0000	0.9991
		Prediction	1	1	1	1	1	1
		Actual Label	1	1	1	1	1	1
3	"Hey loser get a life."	**Model Output**	0.9977	0.4768	0.9580	0.7029	0.9898	0.4662
		Prediction	1	0	1	1	1	0
		Actual Label	0	0	0	0	1	0
4	"You are just so freaking beautiful."	**Model Output**	1.0000	0.9928	0.9999	0.9553	0.9999	0.9949
		Prediction	1	1	1	1	1	1
		Actual Label	0	0	0	0	0	0

To obtain more insight on the model performance, we picked 4 representative comments from the dataset shown in **Table 4**. Case 1 only contains neutral vocabularies and the model was able to give correct

predictions. Case 2 has words “ugly” and “hate”, which are words commonly used in the bad comments. The model picked up the words and gave the correct prediction. However the model accuracy was only 50% in Case #3. Part of this is due to the inherent subjectivity of the labelling in the dataset. The model prediction in case #4 is completely wrong. The model recognized the word “freaking” but misunderstood the meaning of the sentence. This suggests that the model is good at picking up word patterns but lacks understanding of context.

Discussion and Learnings

Classifying comments is a difficult task because there is little structure and a large variety of words used. Considering the challenges of multi-label classification and working with a large quantity of messy, unbalanced data, we are satisfied with our testing results. Our model is able to successfully flag comments that exemplify some characteristics of toxicity, though there are sometimes errors when pinpointing the specific classes. Due to the subjective nature of toxicity, some of our errors can also be attributed to mislabeling of the training data.

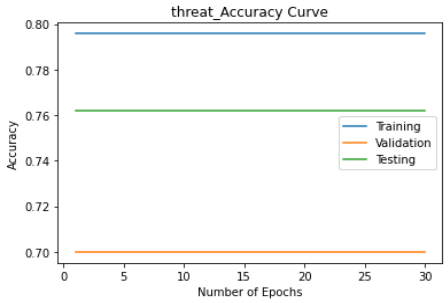
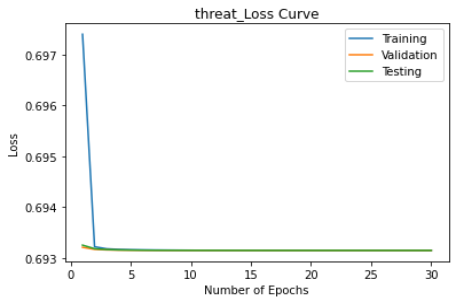
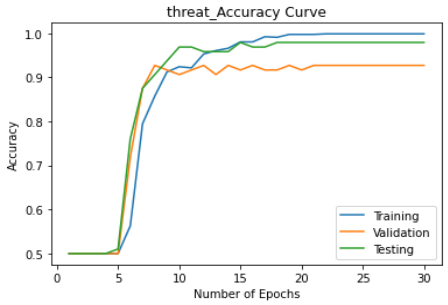
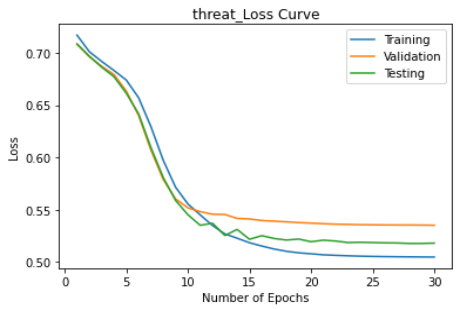
By implementing CNN and LSTM in our architecture and passing in the comments word by word, our final model performed much better than our baseline model and models we cited in literature, indicating that extracting sequence and word patterns plays a role in determining the toxicity of a sentence. However, our model still has room for improvement, being outperformed by the top Kaggle model with 98.9% accuracy.

Our results emphasized the importance of maintaining class balance with binary classifiers. After we trained our models with balanced data, we were able to mitigate a lot of false positives shown in **Table 5**, which was our primary concern. After data augmentation, our model was able to generalize from the increased training examples and move towards a lower loss shown in **Table 6**. Since augmentation was proven to be effective, more sophisticated methods might yield an even more diverse and robust training set, perhaps through back-translation.

Table 5. Comparison of confusion matrices before and after class balancing

Reference		**Severe Toxic Confusion Matrix Before**		**Severe Toxic Confusion Matrix After**	
TP	FP	897	785	686	11
FN	TN	13	1350	19	694

Table 6: Loss and accuracy curves of “threat” class before and after data augmentation

Without data augmentation		
With data augmentation		

During data examining, we saw significant correlation between some classes (e.g. 74% between obscene and insult). As a next step, we could investigate leveraging this pattern through implementing chained classifiers to improve our model accuracy.

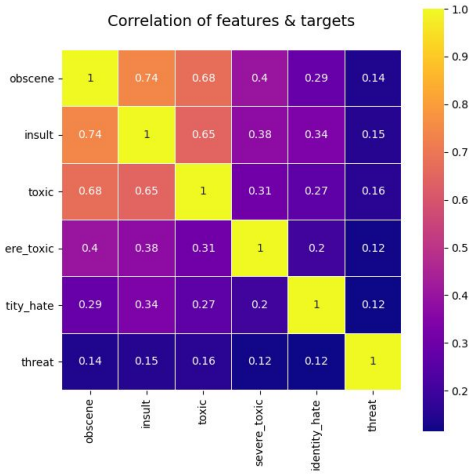


Figure 14: Plot of feature correlation

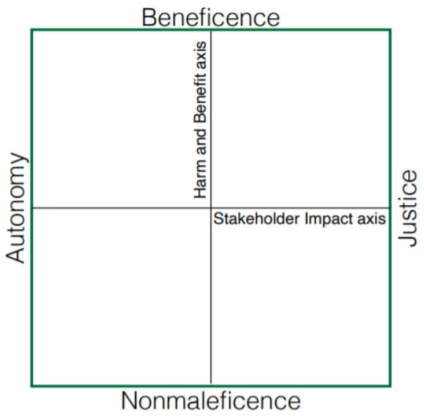


Figure 15: An Imagined Plane of Ethical Reasoning [11]

Ethical framework

The motivating principle behind our project is promoting nonmaleficence within online communities by identifying harmful comments and taking action against them. This is primarily experienced by those who prefer a safe and productive environment without negative distractions.

For the people who post toxic comments, this would reduce their autonomy by limiting their freedom of speech but may also end up limiting the variety of perspectives represented within the forums. In the long term, this could contribute to "political correctness culture" in a destructive way [12]. Therefore, it is important to minimize the amount of false positives our model produces, to encourage all constructive conversation.

Our model also provides beneficence for the platform hosts as it replaces the need to manually moderate discussions, saving time and resources. Employing a machine learning model to filter comments promotes justice, because all comments will be processed on an equal footing.

References

[1] Macleans.ca. 2020. *Online Hate Speech In Canada Is Up 600 Percent. What Can Be Done? - Macleans.Ca*. [online] Available at: <https://www.macleans.ca/politics/online-hate-speech-in-canada-is-up-600-percent-what-can-be-done/> [Accessed 28 October 2020].

[2] R. Hatzipanagos, "Perspective | How online hate turns into real-life violence," *The Washington Post*, 30-Nov-2018. [Online]. Available: https://www.washingtonpost.com/nation/2018/11/30/how-online-hate-speech-is-fueling-real-life-violence/. [Accessed: 28-Oct-2020].

[3] "Court documents identify 13 injured in deadly van attack | CBC News," *CBCnews*, 25-Apr-2018. [Online]. Available: https://www.cbc.ca/news/canada/toronto/injured-van-attack-1.4633308. [Accessed: 28-Oct-2020].

[4] "Potentials of NLP: Techniques, Industry-Implementation and Global Market Outline," *Analytics Insight*, 26-Jan-2020. [Online]. Available: https://www.analyticsinsight.net/potentials-of-nlp-techniques-industry-implementation-and-global-market-outline/. [Accessed: 28-Oct-2020].

[5] T. Davidson, D. Warmsley, M. Macy, and I. Weber, "Automated Hate Speech Detection and the Problem of Offensive Language," dissertation, 2017.

[6] S. MacAvaney, H.-R. Yao, E. Yang, K. Russell, N. Goharian, and O. Frieder, "Hate speech detection: Challenges and solutions," *PLOS ONE*. [Online]. Available: https://journals.plos.org/plosone/article?id=10.1371%2Fjournal.pone.0221152. [Accessed: 28-Oct-2020].

[7] Z. Zhang and L. Luo, "Hate speech detection: A solved problem? The challenging case of long tail on Twitter," *Semantic Web*, vol. 10, no. 5, pp. 925–945, 2019.

[8] E. Thompson, "Facebook partners with Ontario university on 'global network' to counter rise in online hate | CBC News," *CBCnews*, 28-Jul-2020. [Online]. Available: https://www.cbc.ca/news/politics/facebook-hate-online-extremism-1.5664832. [Accessed: 28-Oct-2020].

[9]"Toxic Comment Classification Challenge," *Kaggle*, 2017. [Online]. Available: https://www.kaggle.com/c/jigsaw-toxic-comment-classification-challenge/overview. [Accessed: 28-Oct-2020].

[10] Makcedward, "makcedward/nlpaug," *GitHub*. [Online]. Available: https://github.com/makcedward/nlpaug. [Accessed: 06-Dec-2020].

[11] R. Irish, "Ethics for MI," in *ECE324 Tutorial*, 20-Oct-2020.

[12] J. T. Moss and P. J. O'Connor, "Political correctness and the alt-right: The development of extreme political attitudes," *Plos One*, vol. 15, no. 10, 2020.