IMT575 Final Project
Automated Content Moderation: NLP Against Toxic Comments

Team 1: Jiashu Chen, Qianqian Liu, Irene Yang, Chesie Yu
Advisor: Prof. Aylin Caliskan

Information School
University of Washington
June 2, 2023

Abstract

Efforts to mitigate the impact of toxic content have sparked significant interest in automated content moderation. The dissemination of social media content acts as a catalyst for the perpetuation and amplification of toxicity, reaching beyond the digital sphere. While existing research predominately focuses on identifying toxic content as a unified concept, this study investigates subcategories of toxicity using the Jigsaw toxic comment dataset. Employing pre-processed text and GloVe word embeddings, the work compared different variants and hybrids of deep learning models. Our findings reveal that 2-Layer BLSTM model outperforms other methods including DNN and CNN in F1 score. This approach achieved competitive results leveraging state-of-the-art RNN-based techniques.

1 Introduction

1.1 Motivation

Toxic content is rampant on social media, posing far-reaching harm that extends beyond virtual boundaries. Such toxicity often perpetuates and amplifies as social media platforms broadcast their content to a wider audience. Content moderation thus holds significant implications in shaping the online community; social media companies like Meta or Twitter, despite their limited accountability under Section 230 of the Internet, need to act responsibly to provide a forum for free speech while ensuring proper use.

Manual moderation, however, is infeasible due to the large scale and detrimental effect on human moderators. Automated content moderation, powered by deep learning and natural language processing, can scale these efforts in detection and prevention, fostering a safer and more inclusive online environment. Extensive research has thus been undertaken to develop automated text-based methods to mitigate offensive content, yet challenges remain in identifying the nuances of toxicity within user-generated content.

1.2 Research Objective

This study aims to tackle the challenge of multilabel toxic content detection by focusing on six subtypes of toxicity: toxic, severe toxic, obscenity, threat, insult, and identity hate. As toxicity can manifest in different forms, we hope to devise targeted interventions tailored to specific subtypes leveraging deep learning techniques. Through investigating and understanding the patterns of abusive language, we hope to combat cyberbullying and online hate speech.

2 Related Work

The increasing propagation of hate speech sparked numerous efforts in finding effective countermeasures. Using various datasets and approaches, researchers have investigated the techniques for automated harmful content detection under various contexts. Waseem and Hovy (2016) examined character n-gram features for hate speech detection based on manually labeled tweets. Vigna et al. (2017) implemented SVM and LSTM for hate speech recognition on Facebook comments. Schmitt et al. (2017) investigated aspect-based sentiment analysis using LSTM and CNN. Kraus et al. (2019) proposed a tensor-based, tree-structured deep neural network, Discourse-LSTM, to learn the sentiment from rhetorical structures. While most of the existing automated solutions focus on detecting toxicity as a unified concept, our study will focus on the semantic detection of different subtypes of offensive content, as the next step in tackling the proliferation of online hate speech.

3 Data

3.1 Data Source

This study uses the Jigsaw toxic comment data from Kaggle. Our training corpus consists of over 150,000 Wikipedia comments, pre-labeled to identify 6 subtypes of toxic behavior.

Commonly observed in content moderation challenges, the data exhibits class imbalance with toxic comments representing the minority, particularly for threat and identity hate. Algorithms tend to overfit for majority class on such data, presenting a major challenge in model development. We will discuss the strategies to mitigate this bias and address potential limitations in subsequent sections.

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

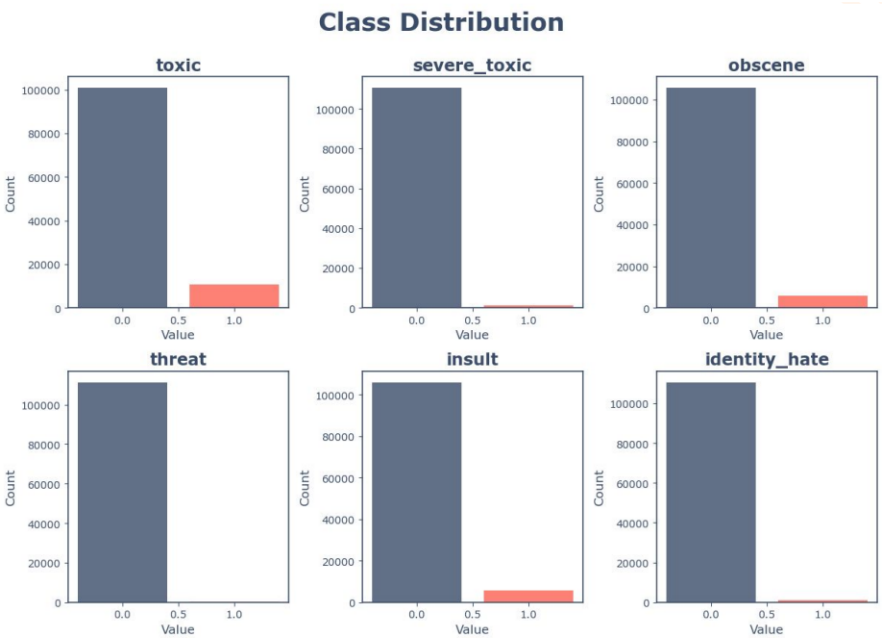

Figure 1: Data Imbalance

3.2 Data Processing

3.2.1 Train-Dev-Test split

We obtain separate training dataset and testing dataset from Kaggle website. We further splitted 20% of the training data to be the development dataset, which is used for parameter tuning and model comparison. The testing dataset is used to give us a final unbiased estimation of model performance.

3.2.2 Text preprocessing

We followed the standard process of text processing in nlp. Steps in text preprocessing include removing space and tab, lowercase sentence, remove non-alphabets, pos tagging and lemmatization, and remove stopwords. We especially chose lemmatization instead of stemming as we noticed some 'over-stemming' problems. Below is an example of text cleaning on one comment in our dataset.

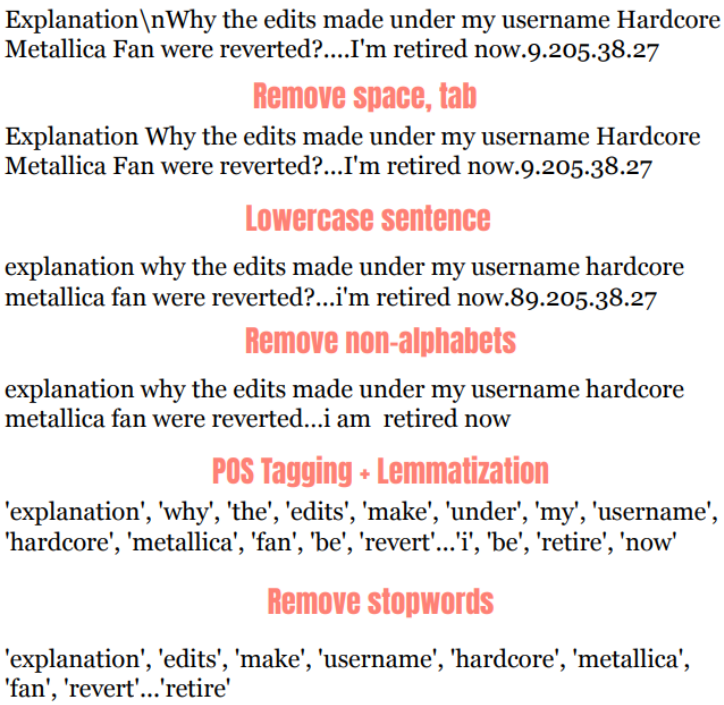

Figure 2: Text Preprocessing Example

3.2.3 Word embeddings

We used Keras tokenizer to extract 20,000 unique words from our preprocessed texts based on frequency. Each word is given a unique index. At the same time, we downloaded GloVe word embeddings, and extracted the 100-dimension word embeddings for each of these words. Thus, we built our embedding matrix which has the dimension of 20,000 * 100. Afterwards,

the normalized embedding matrix becomes our embedding layer, which is the first layer in all of our neural network models.

4 Methodology

4.1 Approach

Commonly used evaluation metrics such as accuracy are misleading for imbalanced data. To avoid over-optimistic assessments, we adopt a more robust evaluation approach that incorporates precision, recall, and F1 score. Precision and recall inform decision-making in targeted aspects: low precision restricts information access and free speech by systematically flagging harmless comments; low recall leads to proliferation of toxic hubs due to inability to capture toxicity. To strike the balance between the two, we therefore prioritize F1 score, the harmonic mean, as the key metric. We hope to achieve the optimal equilibrium between freedom of speech and safe online environment for users.

4.2 Experiments

Using a comparative setting, we examine the performance of different neural network architectures on identical training, dev, and test sets in Google Colab. Each proposed model incorporates 100-dimensional embeddings as input and features a 6-unit dense layer with sigmoid activation for output. To account for class imbalance, we adjust the decision threshold for each subtype to optimize the F1 score, instead of using the default binary classification threshold 0.5. Models are validated on the dev set to determine the best-performing model in each category; hyperparameter tuning is performed using Keras random search tuner. Final evaluation of model effectiveness is carried out using the test set, consisting of 63,978 comments.

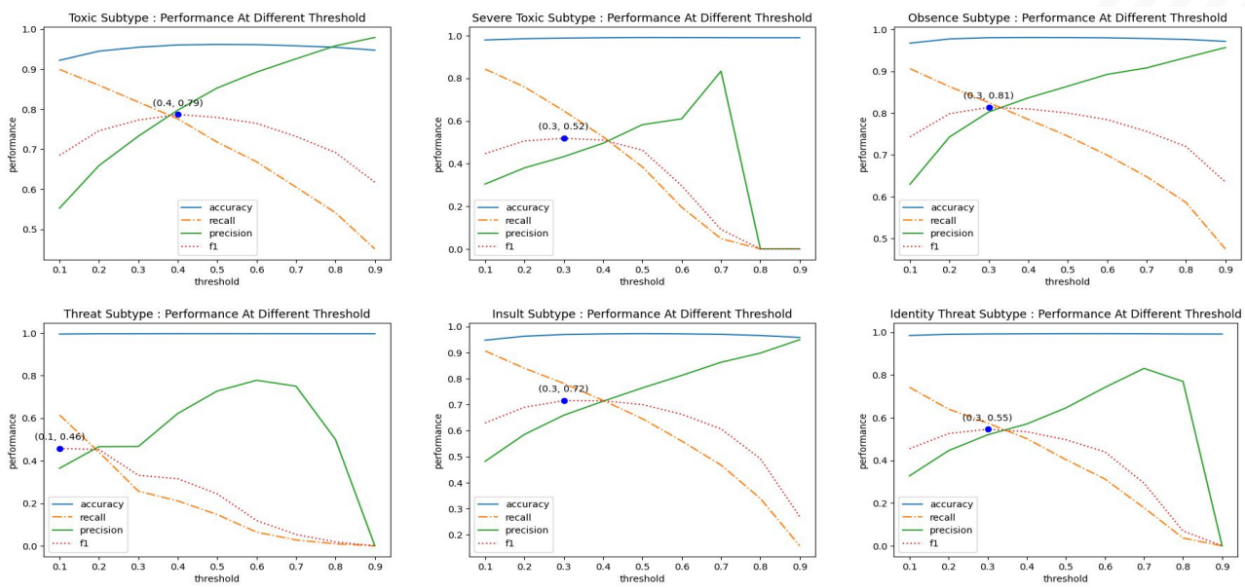

Figure 3: Classification Threshold

Following this framework, we conduct six experiments as follows: DNN, CNN, CNN + RNN (LSTM), RNN (1-Layer BLSTM), RNN (2-Layer BLSTM), RNN (3-Layer BLSTM).

5 Results

5.1 Experiments

5.1.1 Experiment 1

In Experiment 1, a simple DNN with 3 dense layers has been trained. Batch normalization and dropout layers are added in between each dense layer. This DNN can already have an extremely high accuracy rate on test data. However, the F1 scores, especially those for minority classes like severe toxicity, threat, and identity hate, are relatively low.

```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 embedding (Embedding)       (None, 50, 100)           2000000

 flatten (Flatten)           (None, 5000)              0

 dense (Dense)               (None, 32)                160032

 batch_normalization (BatchN  (None, 32)               128
 ormalization)

 dropout (Dropout)           (None, 32)                0

 dense_1 (Dense)             (None, 16)                528

 batch_normalization_1 (Batc  (None, 16)               64
 hNormalization)

 dropout_1 (Dropout)         (None, 16)                0

 dense_2 (Dense)             (None, 6)                 102

=================================================================
Total params: 2,160,854
Trainable params: 2,160,758
Non-trainable params: 96
_________________________________________________________________
```

Figure 4: Model1 Architecture

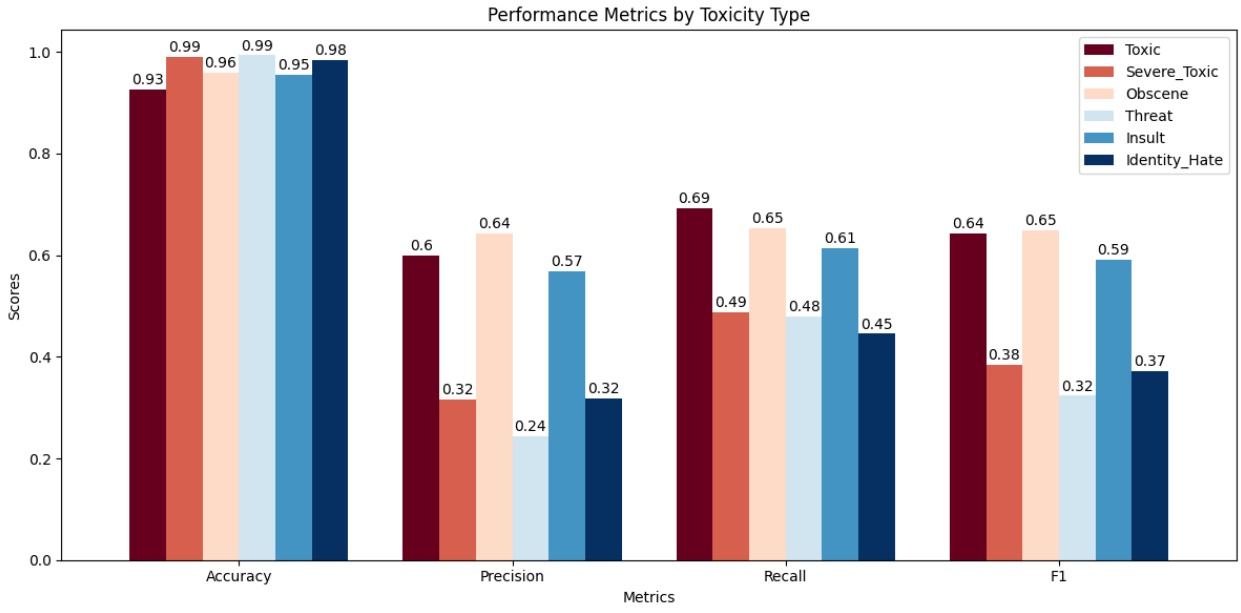

Figure 5: Model1 Test Data Performance

5.1.2 Experiment 2

In experiment 2, a CNN with 1D convolutional layer, followed by global max pooling, dense layers and dropout layers was constructed. Compared with DNN, CNN in experiment 2 has a significant improvement in the F1 score of minority classes.

```
Model: "sequential_1"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 embedding_1 (Embedding)     (None, 50, 100)           2000000

 conv1d (Conv1D)             (None, 48, 128)           38528

 global_max_pooling1d (Globa  (None, 128)              0
 lMaxPooling1D)

 dense_3 (Dense)             (None, 50)                6450

 dropout_2 (Dropout)         (None, 50)                0

 dense_4 (Dense)             (None, 6)                 306

=================================================================
Total params: 2,045,284
Trainable params: 2,045,284
Non-trainable params: 0
_________________________________________________________________
```

Figure 6: Model2 Architecture

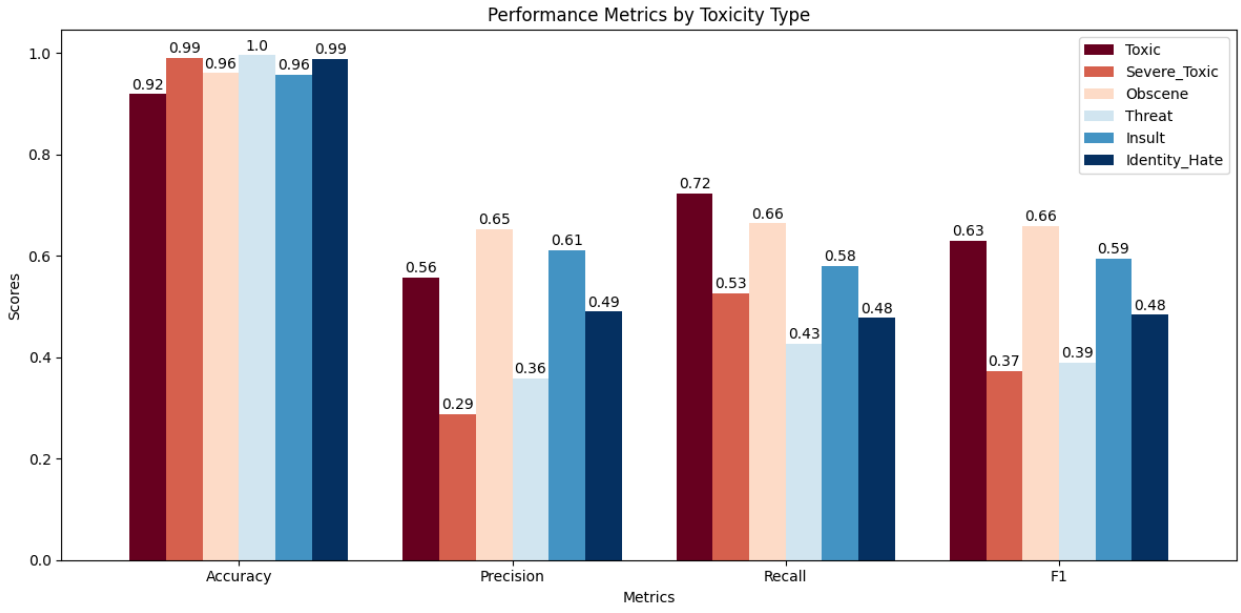

Figure 7: Model2 Test Data Performance

5.1.3 Experiment 3

Based on the CNN model in experiment 2, an extra LSTM layer and normalization layer is added after the global max pooling layer. Compared with the pure CNN in experiment 2, there's slight improvement in classifying toxic, threat, and insult classes.

```
Model: "sequential_2"
_________________________________________________________________
 Layer (type)                 Output Shape              Param #
=================================================================
 embedding_2 (Embedding)      (None, 50, 100)           2000000

 conv1d_1 (Conv1D)            (None, 48, 128)           38528

 max_pooling1d (MaxPooling1D  (None, 24, 128)           0
 )

 lstm (LSTM)                  (None, 64)                49408

 batch_normalization_2 (Batc  (None, 64)                256
 hNormalization)

 dense_5 (Dense)              (None, 50)                3250

 dropout_3 (Dropout)          (None, 50)                0

 dense_6 (Dense)              (None, 6)                 306

=================================================================
Total params: 2,091,748
Trainable params: 2,091,620
Non-trainable params: 128
_________________________________________________________________
```

Figure 8: Model3 Architecture

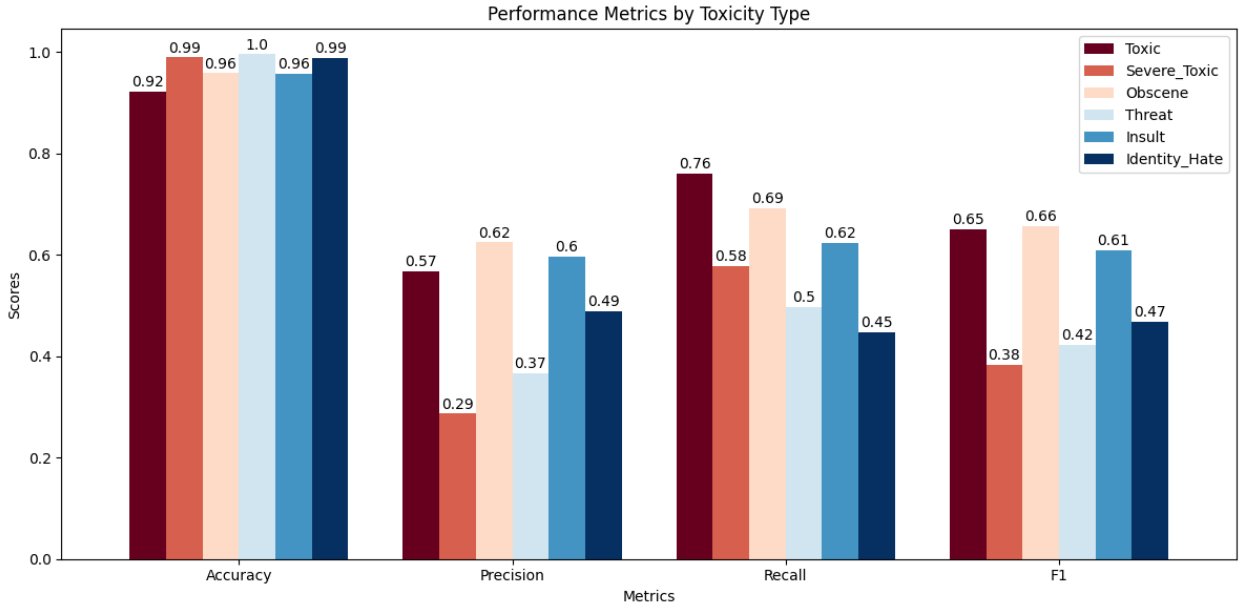

Figure 9: Model3 Test Data Performance

5.1.4 Experiment 4

In experiment 4, a RNN model with 1 Bidirectional LSTM layer (50 training units) followed by global max pooling and dense layers was constructed. The F1 scores for most subtypes of toxic comments has a obvious increase compared with all previous models.

```
Model: "sequential_1"
_________________________________________________________________
 Layer (type)                 Output Shape              Param #
=================================================================
 embedding_1 (Embedding)      (None, 50, 100)           2000000

 bidirectional_1 (Bidirectio  (None, 50, 100)           60400
 nal)

 global_max_pooling1d_1 (Glo  (None, 100)               0
 balMaxPooling1D)

 dense_2 (Dense)              (None, 50)                5050

 dropout_1 (Dropout)          (None, 50)                0

 dense_3 (Dense)              (None, 6)                 306

=================================================================
Total params: 2,065,756
Trainable params: 2,065,756
Non-trainable params: 0
_________________________________________________________________
```

Figure 10: Model4 Architecture

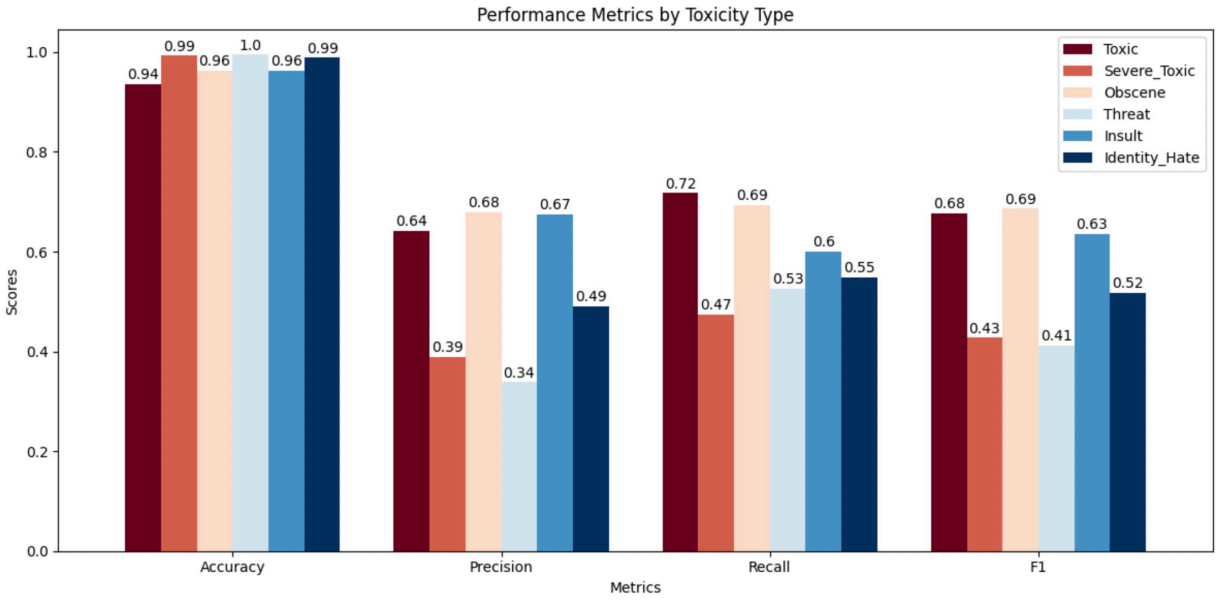

Figure 11: Model4 Test Data Performance

5.1.5 Experiment 5

In experiment 5, an extra Bidirectional LSTM layer with 50 training units was added immediately after the LSTM layer in experiment 4. This new BLSTM model has an even more

outstanding ability in classifying minority classes: severe toxic, insult, identity hate, and it is our best performing model so far.

```
Model: "sequential_2"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 embedding_2 (Embedding)     (None, 50, 100)           2000000

 bidirectional_2 (Bidirectio  (None, 50, 100)          60400
 nal)

 bidirectional_3 (Bidirectio  (None, 50, 100)          60400
 nal)

 global_max_pooling1d_2 (Glo  (None, 100)              0
 balMaxPooling1D)

 dense_4 (Dense)             (None, 50)                5050

 dropout_2 (Dropout)         (None, 50)                0

 dense_5 (Dense)             (None, 6)                 306

=================================================================
Total params: 2,126,156
Trainable params: 2,126,156
Non-trainable params: 0
_________________________________________________________________
```

Figure 12: Model5 Architecture

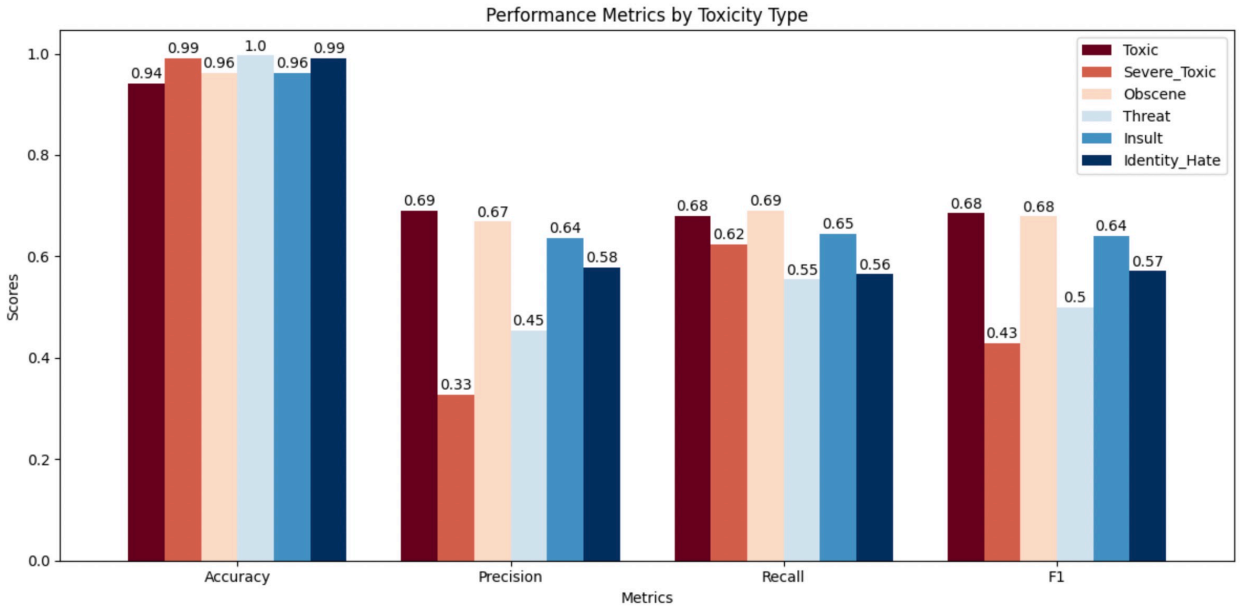

Figure 13: Model5 Test Data Performance

5.1.6 Experiment 6

In experiment 6, another 50-unit-BLSTM layer was added immediately after the 2 BLSTM layers in experiment 5. However, this extra layer drags down our model performance, as the F1 score for each sub category of toxic comment decreases compared with experiment 5.

Layer (type)	Output Shape	Param #
embedding (Embedding)	(None, 50, 100)	2000000
bidirectional (Bidirectional)	(None, 50, 100)	60400
bidirectional_1 (Bidirectional)	(None, 50, 100)	60400
bidirectional_2 (Bidirectional)	(None, 50, 50)	25200
global_max_pooling1d (GlobalMaxPooling1D)	(None, 50)	0
dense (Dense)	(None, 64)	3264
dropout (Dropout)	(None, 64)	0
dense_1 (Dense)	(None, 6)	390

Total params: 2,149,654
Trainable params: 2,149,654
Non-trainable params: 0

Figure 14: Model6 Architecture

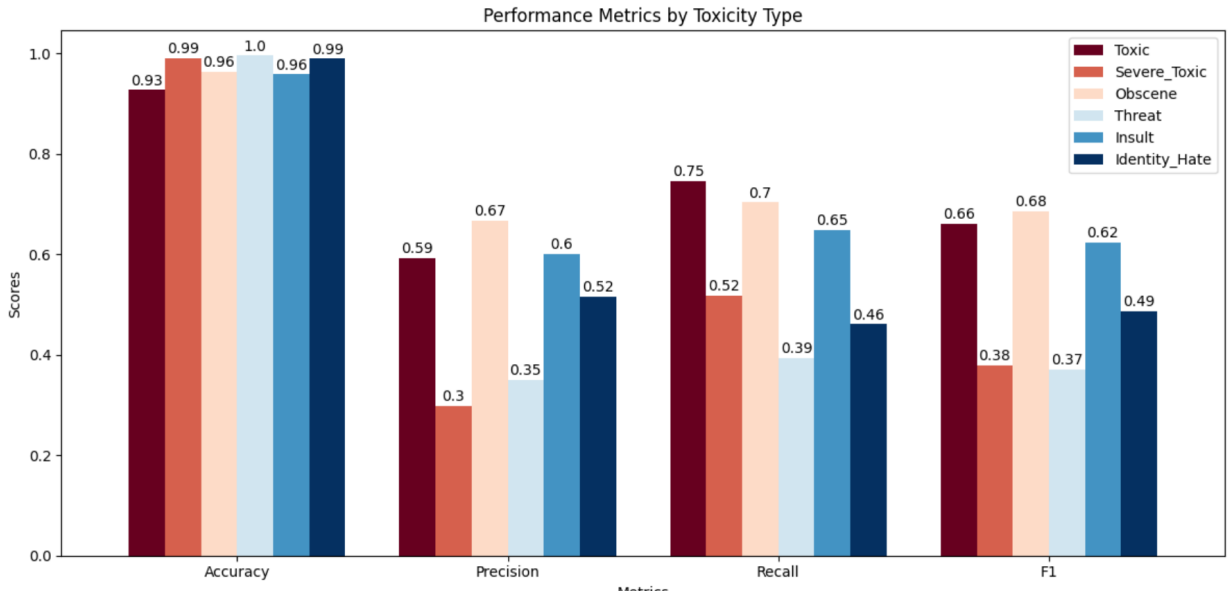

Figure 15: Model6 Test Data Performance

5.2 Summary of models

The F1 scores for each sub category in different models was demonstrated in the table below.

Subtypes	DNN	CNN	CNN + LSTM	RNN: BLSTM	RNN: 2BLSTM	RNN: 3BLSTM
Toxic	0.64	0.63	0.65	0.68	**0.68**	0.66
Severe_Toxic	0.38	0.37	0.38	0.43	**0.43**	0.38
Obscene	0.65	0.66	0.66	**0.69**	0.68	0.68
Threat	0.32	0.39	0.42	0.41	**0.50**	0.37
Insult	0.59	0.59	0.61	0.63	**0.64**	0.62
Identity_Hate	0.37	0.48	0.47	0.52	**0.57**	0.49

Figure 16: F1 Scores of All Models

The Experiment 5 model: 2 layer Bidirectional LSTM model is our best performing model in terms of F1 scores for each sub category. While the Experiment 1 model: DNN model is the worst performing model.

- **Categories with more training data has a better performance**

Comparing the F1 scores for six sub categories among all models, categories with more training data (Toxic, Obscene, Insult) have much higher F1 scores compared with classes with few training data (Severe toxic, Threat, Identity Hate). Comparing the F1 score of non-toxic and toxic comments, the gap will be even more obvious. For any type of machine learning or deep learning model, features of a class will generally be better learned when more data regarding this specific class goes into the training data. Therefore, the overall performance of our model is largely affected by the limited training data regarding the six toxic types.

- **BLSTM perform well in detecting minority class**

There's an obvious performance gap between BLSTM models, compared with CNN and

DNN models. For toxic classes that have more training data points, for example, toxic, the F1 score gap is smaller, while for minority toxic classes, such as severe toxic and identity hate, the F1 score gap is much larger. Since our dataset is an extremely imbalanced dataset, where negative samples (non-toxic comments) occupy 90% of the dataset even in the case of the largest sub category of toxic comment (toxic). We can conclude that Bidirectional LSTM may have its own advantages in learning underlying patterns of minority groups. This is possibly because the information flows both forward and backward in Bidirectional LSTM layers, which helps the model better learn the critical features from minority classes. However, the underlying reason for this better performance still needs to be verified by more literature, and so far we haven't found any.

- **Too many BLSTM layers might lead to overfitting**

Even though the BLSTM models perform better, adding too many BLSTM layers, for example, in experiment 6, will actually lead to overfitting issues. In fact, different combinations of training units have been tried in both 2 layer and 3 layer BLSTM models during the experiment process, and the result demonstrates that 3 layer BLSTM generally performs worse than 2 layer BLSTM. Thus, a 3 layer BLSTM model might be overly complex for our existing dataset, and has the tendency of overfitting.

- **DNN model is too simple for the toxic comment classification task**

A simple DNN model is not sufficient to capture the sentiment features and semantic dependencies of toxic comments of all categories, and this is possibly the reason for its worst performance.

- **CNN doesn't perform very well with imbalanced data**

Generally speaking, CNN is good at capturing position invariant features. Therefore, it makes sense to use CNN to conduct sentiment classification tasks, since CNN can well capture the features in text data, such as insult terms, threat terms, etc. (Ghelani, 2019). However, the CNN model doesn't perform really well in this research, which is possibly due to the extremely imbalanced classes in our training data.

5.3 Comparison with previous research

Compared with 2 previous research in our literature review, which is also doing classification tasks on extremely imbalanced datasets. The F1 score of this research are not very high as well, which again, proves that the imbalanced dataset will largely affect the model performance. These previous research results might also indicate that our model is performing relatively well under the circumstance of using static embeddings.

		Strong hate			**Weak hate**			**No hate**		
Classifier	Accuracy (%)	Prec.	Rec.	F-score	Prec.	Rec.	F-score	Prec.	Rec.	F-score
SVM	64.61	.452	.189	.256	.523	.525	.519	.724	.794	.757
LSTM	60.50	.501	.054	.097	.434	.159	.221	.618	.950	.747

Figure 17: Hate Speech Detection on Facebook Comments, Vigna et al.

		development set	synchronic test set	diachronic test set
Pipeline LSTM	+ word2vec	.350	.297	.342
End-to-end LSTM	+ word2vec	.378	.315	.383
Pipeline CNN	+ word2vec	.350	.298	.343
End-to-end CNN	+ word2vec	.400	.319	.388
Pipeline LSTM	+ glove	.350	.297	.342
End-to-end LSTM	+ glove	.378	.315	.384
Pipeline CNN	+ glove	.350	.298	.342
End-to-end CNN	+ glove	.415	.315	.390
Pipeline LSTM	+ fasttext	.350	.297	.342
End-to-end LSTM	+ fasttext	.378	.315	.384
Pipeline CNN	+ fasttext	.342	.295	.342
End-to-end CNN	+ fasttext	**.511**	**.423**	**.465**
majority class baseline		–	.315	.384
GermEval baseline		–	.322	.389
GermEval best submission		–	.354	.401

Figure 18: Joint Aspect and Polarity Classification for Aspect-based Sentiment Analysis, Schmitt et al.

6 Discussion

6.1 Findings and Insights

Most of the research papers we have read so far have focused on developing a robust model to detect generic toxic comments. The contribution of our study is that we examine subtypes of toxic comments.

Overall, we found that the RNN model performed the best, the DNN model performed the worst, and the CNN model performed average. The most robust model is the RNN with a 2-layer BLSTM combined with batch normalization. The sequential structure of LSTM allows the model to process text data efficiently by retaining information over time. This attribute makes it well-suited for sequence modeling tasks, which is what our project is doing for sentiment analysis. However, we need to be aware that the LSTM model may be prone to overfitting. In our experiments, the F1 scores improved when adding from 1-layer BLSTM to 2-layer BLSTM, but the F1 scores decreased when adding from 2-layer BLSTM to 3-layer BLSTM. Meanwhile, adding batch normalization helps improve model performance by reducing overfitting, and its ability for faster convergence. On the other hand, we found that the main bottleneck of DNN models and CNN models is that sometimes they don't perform well in minority subtypes. For example, the DNN model has low performance in detecting the threat class (F1 score = 0.32) and the identity hate class (F1 score = 0.37).

6.2 Limitations and Future Work

In our project, the data had a class imbalance, with the toxic subtype accounting for nearly 10% of all comments and some subtypes accounting for less than 1%. To address the class imbalance, we used F1 scores to evaluate the model performance because it provides a better assessment of the performance of the minority class than simply using accuracy. Another way to address class imbalance is resampling, which we have not explored in this project. In the future, we will explore in depth the methods of oversampling or undersampling such

as random oversampling which is replicating random records from the minority class, or SMOTE which is generating synthetic samples for the minority class.

In the process of building and optimizing the model, we found that the RNN LSTM model has an excellent performance in detecting the minority classes in the imbalanced dataset. As of now, we have not found any research papers that support our speculation. But in the future, we would like to figure out whether the good performance of the RNN LSTM model is due to its strength in learning textual data or whether it indeed has some advantages in detecting minority classes.

As we are dealing with text data, and in real life, there are often typos, misspellings, or slang words in comments, which would not be presented in the regular corpus, making it unable to be processed for word embedding. We are interested in exploring in depth how to perform sentiment analysis on such words and sentences.

7 Ethical Considerations

The trade-off between free speech and a safe environment: Content moderation plays a key role in creating a safe and diverse online environment. A model that automatically detects toxic content can effectively reduce the likelihood of verbal violence and cyberbullying, further supporting individuals to freely express their opinions without fear of harassment. However, while implementing content moderation on toxic content, we should ensure that the model is fair and equitable in terms of not restricting legitimate discussion and diverse viewpoints. Excessive moderation and filtering of user-generated content can impede freedom of expression and accessibility of information.

8 Conclusion

This study examines the performance of deep learning models in natural language processing on a highly-skewed, multilabel dataset. With extensive experiments, we demonstrate that

2-layer BLSTM outperforms DNN, CNN, hybrid CNN + LSTM, 1-layer and 3-layer BLSTM. This state-of-the-art RNN-based approach achieves competitive results in identifying toxic content, leveraging its ability to capture order dependence in sequential data. In our upcoming plan, we aim to explore sampling techniques to address class imbalance and handle out-of-vocabulary words from intentionally obfuscated content.

References

Ghelani, S. (2019b, June 2). Text classification-RNN's or CNN's? Medium. https://towardsdatascience.com/text-classification-rnns-or-cnn-s-98c86a0dd361

Kraus, M., & Feuerriegel, S. (2019). Sentiment analysis based on rhetorical structure theory:learning deep neural networks from discourse trees. Expert Systems with Applications, 118, 65–79. https://doi.org/10.1016/j.eswa.2018.10.002

Schmitt, M., Steinheber, S., Schreiber, K., & Roth, B. (2018). Joint Aspect and polarity classification for aspect-based sentiment analysis with end-to-end neural networks. Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing. https://doi.org/10.18653/v1/d18-1139

Toxic comment classification challenge. Kaggle. (2018). https://www.kaggle.com/competitions/jigsaw-toxic-comment-classification-challenge/data

Vigna, F. D., Cimino, A., Dell'Orletta, F., & Petrocchi, M. (2017, January). Hate me, hate me not: Hate speech detection on facebook. Researchgate. https://www.researchgate.net/publication/316971988_Hate_me_hate_me_not_Hate_speech_detection_on_Facebook

Waseem, Z., & Hovy, D. (2016). Hateful symbols or hateful people? predictive features for hate speech detection on Twitter. Proceedings of the NAACL Student Research Workshop. https://doi.org/10.18653/v1/n16-2013