

BiLSTM vs Feedforward Neural Networks for Toxicity Detection

Introduction:

Toxic online interactions often contain language such as insults, threats, and hate speech, which can lead to individuals feeling harassed and being socially excluded. For this reason, social media platforms aim to develop fair, transparent, and accurate models that contribute to safer online spaces and more inclusive communities. Because moderating content manually is very expensive and can even introduce individual biases from each moderator, these platforms have increasingly relied on automated tools that can detect toxicity and abusive content. This project focuses on building two distinct, but similar models that classify user comments into six toxicity categories using the Jigsaw Toxic Comment Classification dataset (0). The goal of this project is to design, train, and evaluate a baseline FeedForward Neural Network (FFNN) against an improved advanced Bidirectional LSTM (BiLSTM) with attention as a higher proportion of samples are judged toxic in the presence of contextual information (1), and then compare their performance using standard multi-label evaluation metrics.

Toxicity classification is a multi-label problem where each comment may exhibit zero, one, or multiple toxic behaviors. Early research and traditional approaches relied on surface-level features, e.g., logistic regression using bag-of-words approaches (2 & 3), but even though these approaches were reported to be highly predictive (4) and easily interpretable, they still suffered with false positives as the presence of certain patterns could lead to misclassification (5), which is a problem I also encountered with my models and will elaborate upon later. More recent advances in Natural Language Processing (NLP) highlight the effectiveness of deep models such as Convolutional Neural Networks (CNN) (6), Long Short-Term Memory Networks (LSTM) (7) which is well-suited for sequential text, and transformer-based architectures (like BERT) (8) which learns contextual word embeddings instead of static embeddings and have all shown a clear advantage in toxicity detection (9 & 10).

Unlike traditional binary sentiment or toxicity tasks, I frame the task as a multi-label classification problem since the labels are not mutually exclusive, where each comment may belong to *zero, one, or several* toxicity categories simultaneously (e.g., toxic, severe_toxic,

obscene, threat, insult, and identity_hate). Each label is treated as an independent binary classification task, but all are trained jointly via a shared neural network, which is why this task requires models that can capture overlapping patterns of abusive language and predict multiple outputs at once. Unlike softmax-based single-label tasks, this setup uses sigmoid activation per label and Binary Cross-Entropy with Logits (BCEWithLogitsLoss).

Methodology:

This project's methodology consists of four stages: preprocessing, feature representation, model training, and evaluation. Each comment that trains and tests the models undergoes the following preprocessing steps: lowercasing every character to maintain consistency in the vocabulary, removing punctuation and special characters (which may also remove valuable cues like emojis so further discussed below), splitting comments into individual words based on whitespace for tokenization, truncating comments that are longer than 75 tokens and padding shorter comments with <PAD> so the models can all operate on fixed-sized input tensors which covers the majority of cases while keeping model size and training time manageable, and limiting the vocabulary to the 20,000 most frequently used words in the training set (replacing any other words with a special <UNK> token). Each token is mapped to an integer index, which is then passed through a trainable embedding layer (for this project, the dimension is set at 100) that learns a dense vector representation for each word, allowing the model to learn meaningful semantic structures. Both models are then trained using BCEWithLogitsLoss, which is suitable for independent multi-label outputs, positive class weighting (pos_weight), which comes from the frequency of each label and works to counter heavy class imbalance, the Adam optimizer at a learning rate of 0.001 = weight decay for regularization, mini-batch training of size 64 which balances performance and efficiency, and 25 training epochs.

The datasets used in this project originate from the publicly available Jigsaw Toxic Comment Classification Challenge hosted on Kaggle, which contains comments scraped from Wikipedia discussion pages, specifically from “administration pages where editors can discuss improvements to articles or other Wikipedia pages” (11), where multiple human raters manually annotate each comment.

The baseline FFNN treats each comment as a bag of embeddings, converting token indices to embeddings, averaging the embeddings across the sequence, which removes word

order while simplifying and being computationally efficient, passing through 2 fully connected hidden layers (of size 64), each with a Rectified Linear Unit (ReLU) activation function and a dropout of 0.3 for regularization, and an output layer of 6 units (one per toxicity label). The advanced BiLSTM captures sequential patterns and long-distance dependencies, also converting token indices to embeddings, a LSTM layer of size 128 (which outputs a forward and backward sequence representation of `hidden_size * 2`), mean pooling over time by averaging the hidden states across the sequence, a dropout of 0.4 for regularization, and an output layer of 6 units.

Results:

Before training even began, to understand the distribution and structure of the comment texts, some basic statistics of the training dataset were computed: the median comment length is 35 tokens, the mean length is around 65 tokens, the 75th percentile is located at 73 tokens, the 85th percentile is located at 111 tokens, and the longest comment length was 1,403 tokens long. These results show that the dataset is heavily skewed toward shorter comments, with a long tail of very long inputs, which is why I decided to keep the maximum length of tokens at 75 as a reasonable compromise between computational efficiency and retention of meaningful content.

Per-Label Metrics (Baseline FFNN):

Label	Precision	Recall	F1 Score
toxic	0.3958	0.8984	0.5495
severe_toxic	0.1509	0.9375	0.2600
obscene	0.4117	0.9192	0.5687
threat	0.0214	0.8462	0.0417
insult	0.3610	0.9122	0.5173
identity_hate	0.0956	0.8169	0.1711

Per-Label Metrics (Advanced BiLSTM):

Label	Precision	Recall	F1 Score
toxic	0.4908	0.8529	0.6230
severe_toxic	0.1382	0.9625	0.2418
obscene	0.4350	0.9216	0.5910
threat	0.0191	0.7692	0.0373
insult	0.4020	0.8951	0.5548
identity_hate	0.0832	0.8028	0.1508

Per-Label Confusion Matrices (Baseline FFNN):

TP: True Positive **FP:** False Positive **FN:** False Negative **TN:** True Negative

Label: toxic	Label: severe_toxic	Label: obscene
TP: 672	FP: 1026	TP: 387
FN: 76	TN: 6205	FP: 553
		FN: 34
		TN: 7005
Label: threat	Label: insult	Label: identity_hate
TP: 11	FP: 503	TP: 58
FN: 2	TN: 7463	FP: 549
		FN: 13
		TN: 7359

Per-Label Confusion Matrices (Advanced BiLSTM):

Label: toxic	Label: severe_toxic	Label: obscene
TP: 638	FP: 662	TP: 388
FN: 110	TN: 6569	FP: 504
		FN: 33
		TN: 7054
Label: threat	Label: insult	Label: identity_hate
TP: 10	FP: 513	TP: 57
FN: 3	TN: 7453	FP: 628
		FN: 14
		TN: 7280

Comparison of Models:

Metric	Baseline FFNN	Advanced BiLSTM	Better Model
Subset Accuracy:	0.7828	0.8269	BiLSTM
Macro Precision :	0.2394	0.2614	BiLSTM
Macro Recall:	0.8884	0.8674	Baseline
Macro F1 Score:	0.3514	0.3665	BiLSTM

Some key points to note with each model. With the baseline FFNN, there was strong recall across all labels (often greater than .90), there was low precision, especially for rare labels such as *threat* and *identity_hate*, and its F1 scores ranged from ~0.04 (*threat*) to ~0.57 (*obscene*), showing how the baseline model tends to over-predict toxic labels (high recall, low

precision), which is common with highly imbalanced multi-label tasks like this. With the advanced BiLSTM, F1 scores increased across *toxic*, *insult*, *obscene*, and *identity_hate*, and it produced fewer false positives compared to the simpler baseline, basically outperforming the baseline model across almost all meaningful metrics (e.g., macro precision/accuracy/F1 score), demonstrating its strong ability to detect toxic comment while maintaining high recall, likely as a result of its sequential understanding of text. However, rare classes such as *threat*, *identity_hate*, and *severe_toxic* still remain challenging to correctly classify due to extreme label imbalance.

Discussions:

The BiLSTM with attention clearly outperformed the baseline FFNN model in nearly all aspects, proving that better sequential modeling and understanding increase precision. A bidirectional context helps a model capture nuance like sarcasm, multi-word insults, and certain threat and hate phrasing, and attention helps highlight toxic keywords like insults or slurs. The BiLSTM strikes a better balance than the FFNN, producing higher quality, more selective toxic predictions, making it significantly more effective for automating moderation, toxicity monitoring, and content filtering, where minimizing the number of false alarms and missed toxic content is necessary for healthier online discussions.

Some unexpected observations that came with the project include how certain labels, such as *threat* and *identity_hate*, remained difficult for both models to correctly classify, mainly as a result of their low frequency and how threats can be expressed and interpreted in many various ways. Also, during BiLSTM's training period, its average training loss for each epoch was fluctuating/oscillating (between ~0.42 and ~0.39), suggesting that the model was potentially highly sensitive to the learning rate, that it was overfitting on minority labels, and that it was being influenced by the “long-tail” inputs (comments greater than 150 tokens). I was also pleasantly surprised by how well the basic FFNN performed against the BiLSTM with attention, especially in recall, which can be an indicator that these shallow networks can still detect basic toxicity, possibly because toxicity can appear as obvious words or phrases.

Limitations of this project were not nonexistent, as I performed my training on a smaller subset of the training dataset (39892 samples) rather than the complete Jigsaw training dataset (160,000 samples), mainly because of my laptop's computational constraints, including limited GPU availability. This may have caused the models to not have captured the full diversity of

toxic language present in the complete dataset, explaining the dataset imbalance that I encountered, which led to a high false positive rate, unstable F1 values, and the models overfitting to the majority labels (of which pos_weight partially helped but not fully solved). Other limitations that were affected by my computational constraints include: using a fixed vocabulary size of 20,000 words, so rare slurs or coded language were often ignored, setting the maximum sequence length to 75 tokens/words, so long context for lengthy comments or multi-sentence toxicity were disregarded, keeping the number of training epochs low, so the model would be cut short during its training. I also didn't include pretrained embeddings, opting towards randomly initialized embeddings, which lack inherent semantic relationships (so "father" and "mother" would not be inherently closer in the embedding space than "father" and "princess") and prior knowledge (models learn effective representations from scratch).

Potential improvements to this area of study include integrating pretrained embeddings, like GloVe, to both models, as this will result in a big improvement in their performance, especially with precision, rare label detection, and their F1 score by better understanding uncommon, nuanced, and subtle toxic slurs. A further dive into fine-tuning different parameters and hyperparameters like the learning rate, batch size, LSTM hidden size, number of layers, dropout rate, and the embedding dimension could further enhance stability and performance (I briefly & manually tried a few different options, which is why I landed on different dropout and hidden_size numbers for the FFNN and BiLSTM). Comparing the performance results of both these models against a modern benchmark transformer model like DistilBERT will be intriguing to see just how much sequence encoders with more sophisticated attention mechanisms outperform traditional RNN-based architectures. Improving the preprocessing stage by implementing lemmatization, spelling correction, and more robust text-cleaning techniques like emoji normalization and handling elongated words ("soooo") could also improve the model's performance. Lastly, incorporating the entire training and test dataset will enable the model to learn from a more balanced dataset, resulting in more stable, reliable performance estimates that are necessary for real-world applications.

Code Availability:

The Python code of the analysis of this work can be found on GitHub:

<https://github.com/EduardoRebollar/BiLSTM-vs-Feedforward-Neural-Networks-for-Toxicity-Detection.git>.

References

- (0) Cjadams, Jeffrey Sorensen, Julia Elliott, Lucas Dixon, Mark McDonald, Nithum, and Will Cukierski. 2017. [Toxic Comment Classification Challenge](#). Kaggle.
- (1) Madhyastha, Pranava & Founta, Antigoni & Specia, Lucia. 2023. [A study towards contextual understanding of toxicity in online conversations](#). Natural Language Engineering. 29. 1-23. 10.1017/S1351324923000414.
- (2) William Warner and Julia Hirschberg. 2012. [Detecting Hate Speech on the World Wide Web](#). In Proceedings of the Second Workshop on Language in Social Media, pages 19–26, Montreal, Canada. Association for Computational Linguistics.
- (3) Zeerak Waseem and Dirk Hovy. 2016. [Hateful Symbols or Hateful People? Predictive Features for Hate Speech Detection on Twitter](#). In Proceedings of the NAACL Student Research Workshop, pages 88–93, San Diego, California. Association for Computational Linguistics.
- (4) Anna Schmidt and Michael Wiegand. 2017. [A Survey on Hate Speech Detection using Natural Language Processing](#). In Proceedings of the Fifth International Workshop on Natural Language Processing for Social Media, pages 1–10, Valencia, Spain. Association for Computational Linguistics.
- (5) Irene Kwok and Yuzhou Wang. 2013. [Locate the Hate: Detecting Tweets against Blacks](#). In Proceedings of the Twenty-Seventh AAAI Conference on Artificial Intelligence, AAAI'13, page 1621–1622. AAAI Press
- (6) Björn Gambäck and Utpal Kumar Sikdar. 2017. [Using Convolutional Neural Networks to Classify Hate-Speech](#). In Proceedings of the First Workshop on Abusive Language Online, pages 85–90, Vancouver, BC, Canada. Association for Computational Linguistics.
- (7) Pinkesh Badjatiya, Shashank Gupta, Manish Gupta, and Vasudeva Varma. 2017. [Deep Learning for Hate Speech Detection in Tweets](#). In Proceedings of the 26th International Conference on World Wide Web Companion, WWW '17 Companion, page 759–760, Republic and Canton of Geneva, CHE. International World Wide Web Conferences Steering Committee

- (8) Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. [BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding](#). In Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.
- (9) Marcos Zampieri, Shervin Malmasi, Preslav Nakov, Sara Rosenthal, Noura Farra, and Ritesh Kumar. 2019b. [SemEval-2019 Task 6: Identifying and Categorizing Offensive Language in Social Media \(OffensEval\)](#). In Proceedings of the 13th International Workshop on Semantic Evaluation, pages 75– 86, Minneapolis, Minnesota, USA. Association for Computational Linguistics.
- (10) Marcos Zampieri, Preslav Nakov, Sara Rosenthal, Pepa Atanasova, Georgi Karadzhov, Hamdy Mubarak, Leon Derczynski, Zeses Pitenis, and Çağrı Çöltekin. 2020. [SemEval-2020 Task 12: Multilingual Offensive Language Identification in Social Media \(OffensEval 2020\)](#). In Proceedings of the Fourteenth Workshop on Semantic Evaluation, pages 1425– 1447, Barcelona (online). International Committee for Computational Linguistics.
- (11) Wikipedia contributors. 2025. [Help:Talk pages](#). In *Wikipedia, The Free Encyclopedia*. Retrieved November 2025