Artificial Intelligence and Machine Learning (6CS012)

Text Classification of Tweets for Offensive Content Detection using RNN, LSTM, and Word2Vec Embeddings

Student Name: Bikesh Maharjan

Student id: 2332936

Group: L6CG12

Tutor: Sunita Parajuli

Lecturer: Mr. Siman Giri

Date: 20th May 2025

Abstract

This project is concerned with the classification of tweets as offensive or non-offensive. in our case to the deep learning models namely Recurrent Neural Networks (RNN). It uses the following tools: Long Short-Term Memory (LSTM) networks, and pretrained Word2Vec embeddings. The goal is to identify offensive content in social media as part of the moderation of content. The data set, made up of tweets labeled, were pre-processed through text cleansing, tokenization, as well as lemmatization. Three models were developed: A Simple RNN, LSTM and LSTM withWord2Vec. embeddings. The LSTM with the Word2Vec produced the best accuracy. 94.56% better than the Simple RNN (93.43%) and LSTM (94.04%). The use of pretrained embeddings enhanced performance due to capturing semantic relations. According to these findings, LSTM-based models having pretrained embeddings are useful for text classification. Future work might be able to dig into larger datasets and more complicated architectures such as transformers to enhance performance.

Table of Contents

Table of Figures

1. Introduction

Text classification is an important activity in natural language processing that has been implemented for sentiment analysis, spam detection, and content moderation. This project deals with the problem of categorizing tweets as offensive or non-offensive, and this is important on ensuring safe online environments. Recurrent Neural Networks (RNNs), and the Long Short-Term Memory (LSTM) networks are suitable for sequential data such as text thanks to their capacity of modelling temporal dependencies. Pretrained Word2Vec embeddings enhance word representations by capturing semantic relationships. The previous studies of text classification have proved the efficiency of deep learning models, but problems such as vanishing gradients in RNNs and necessity of robust embedding still remain. This project compares RNN, LSTM, and LSTM with the Word2Vec as a way of overcoming these challenges. (Zhang, X., Zhao, J., & LeCun, Y., 2015)

2. Dataset

The employed dataset is labeled set of tweets for identifying the racist or sexist. content which is publicly available from a repository. It contains 31,962 tweets, each abridged as offensive 1 or not offensive 0. The dataset was split into 80% training (25,569 tweets) and testing (6,393 tweets) sets at a ratio of 20:80%.

Preprocessing steps included:

- Converting text to lowercase.
- Growing contractions – (e.g.” don’t” to” do not”).
- Removing URLs, mentions, hashtags and special characters.
- Tokenizing and removing stopwords.
- Using lemmatization to normalize words.

The cleaned text was tokenized and padded to a maximum length on the basis of the 95th percentile of sequence lengths.

3. Methodology

3.1 Text Preprocessing

The text preprocessing pipeline implied cleansing tweets of noise. standardize input. Some of the steps were lowercase conversion, contraction expansion. URLs, mentions, and hashtags removal, stopwords removal, and lemmatization. using NLTK's WordNetLemmatizer. The cleaned text was tokenized with the use of Kera's. Tokenizer and padded to have a fixed length for input to the model. (Hochreiter, S., & Schmidhuber, J., 1997)

3.2 Model Architecture

Three models were implemented:

1. **Simple RNN:** A simple RNN with embedding layer of 100 dimensions, 64-unit RNN layer, and a dense layer with sigmoid function as an output layer.

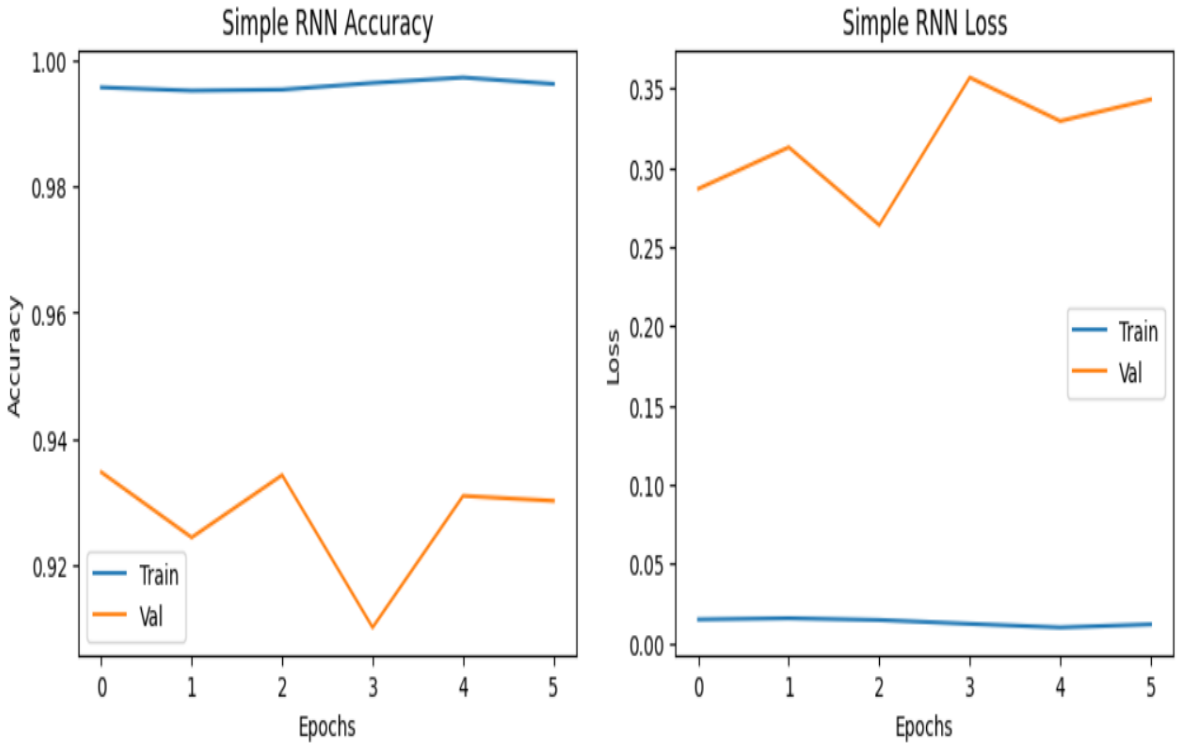

Figure 1: Simple RNN Accuracy and Loss

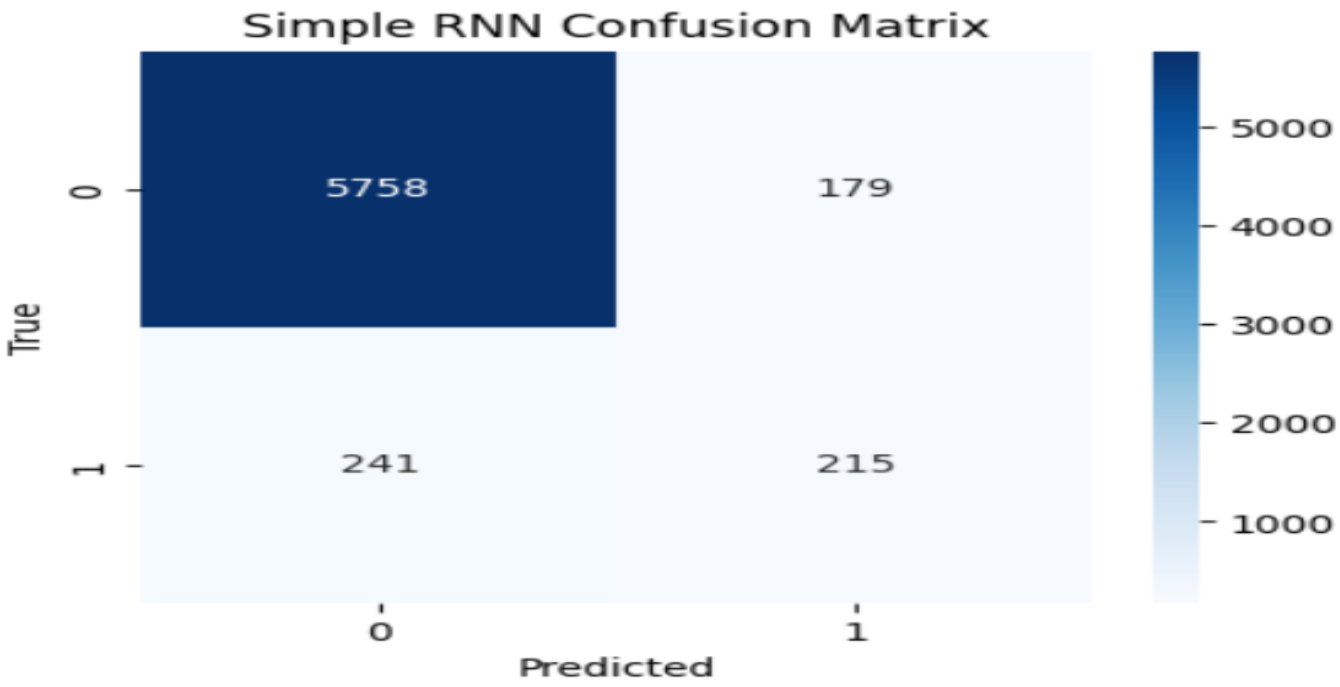

Figure 2: Simple RNN Confusion Matrix

2. **LSTM:** A LSTM model with an embedding layer (100 dimensions), a 64 passing through dense layers has been used with optimizer as ¨Adam¨ and activation ¨relu¨ unit LSTM layer, dense sigmoid output layer.

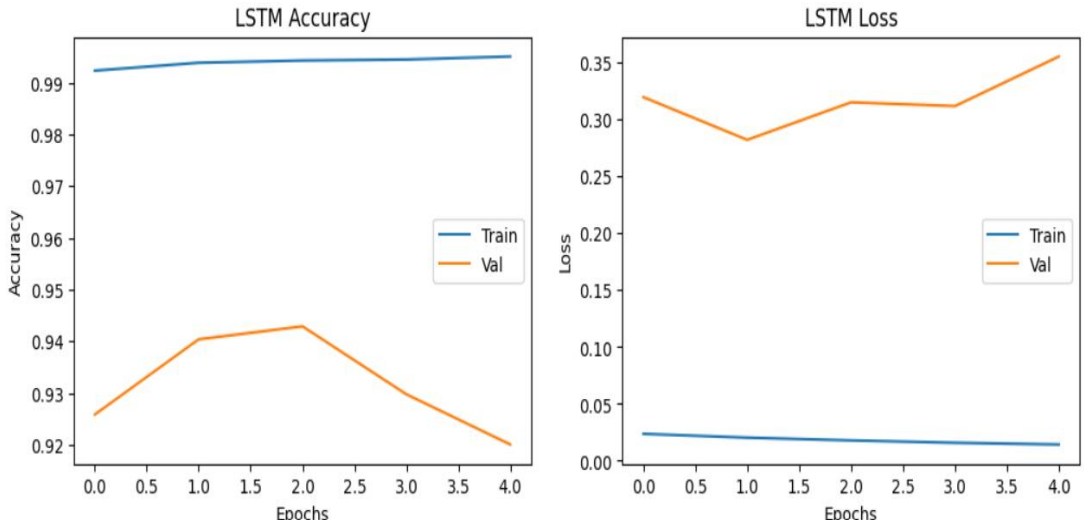

Figure 3: LSTM Accuracy and Loss

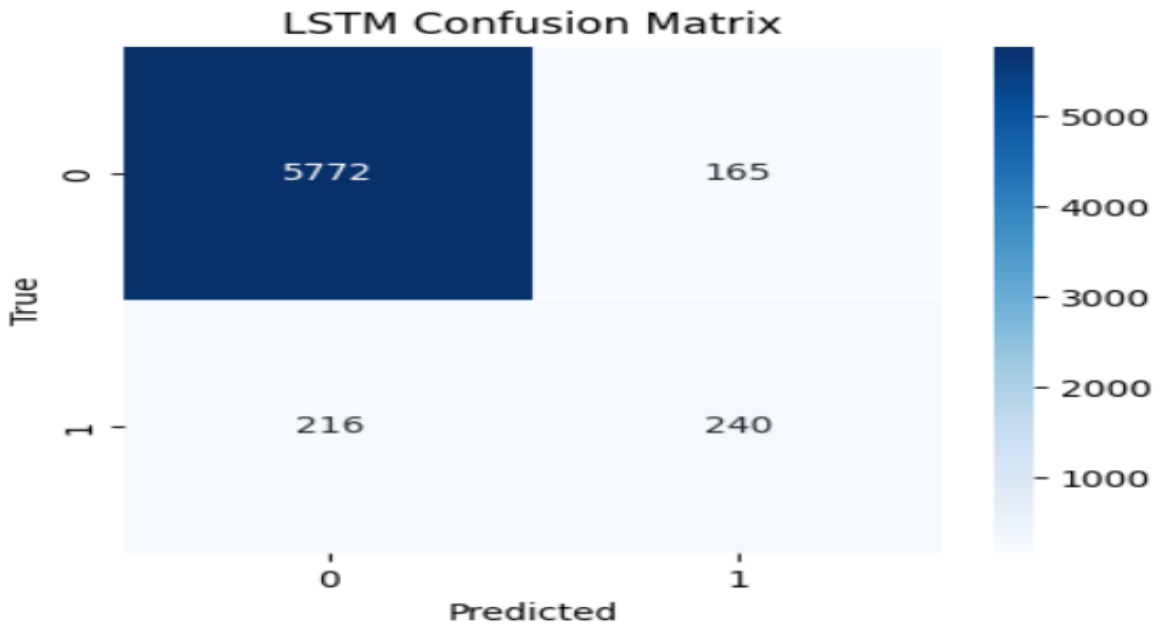

Figure 4: LSTM Confusion Matrix

3. **LSTM with Word2Vec:** An LSTM model with a pretrained Word2Vec embedding layer (300d, non-trainable) – 64 LSTM units and a – 5 nodes. dense sigmoid output layer.

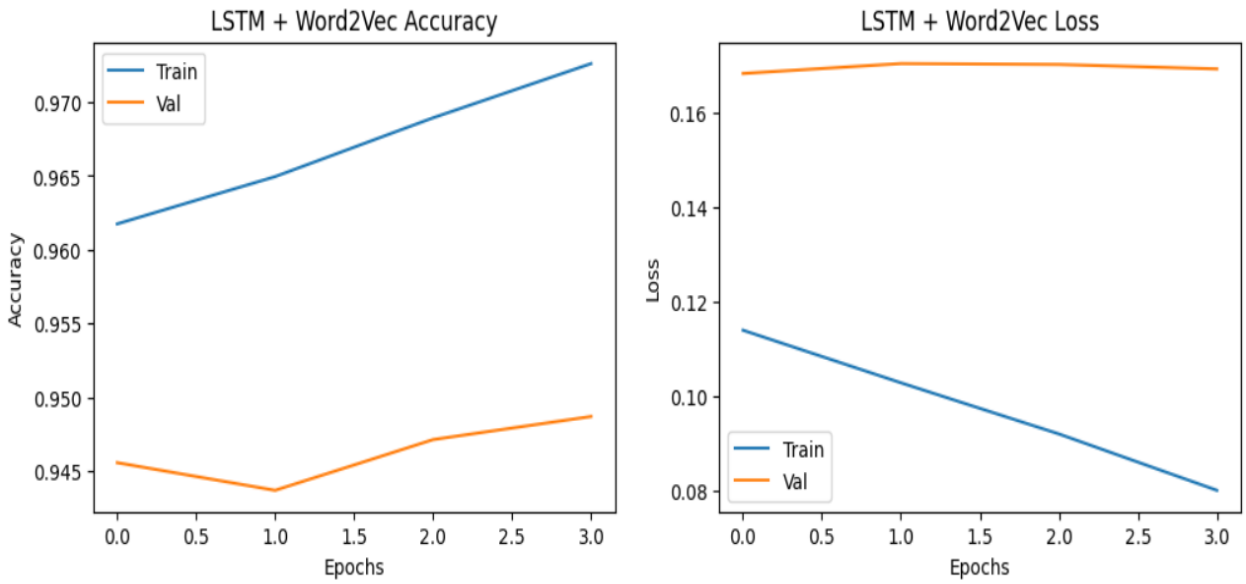

Figure 5: LSTM + Word2Vec Accuracy and Loss

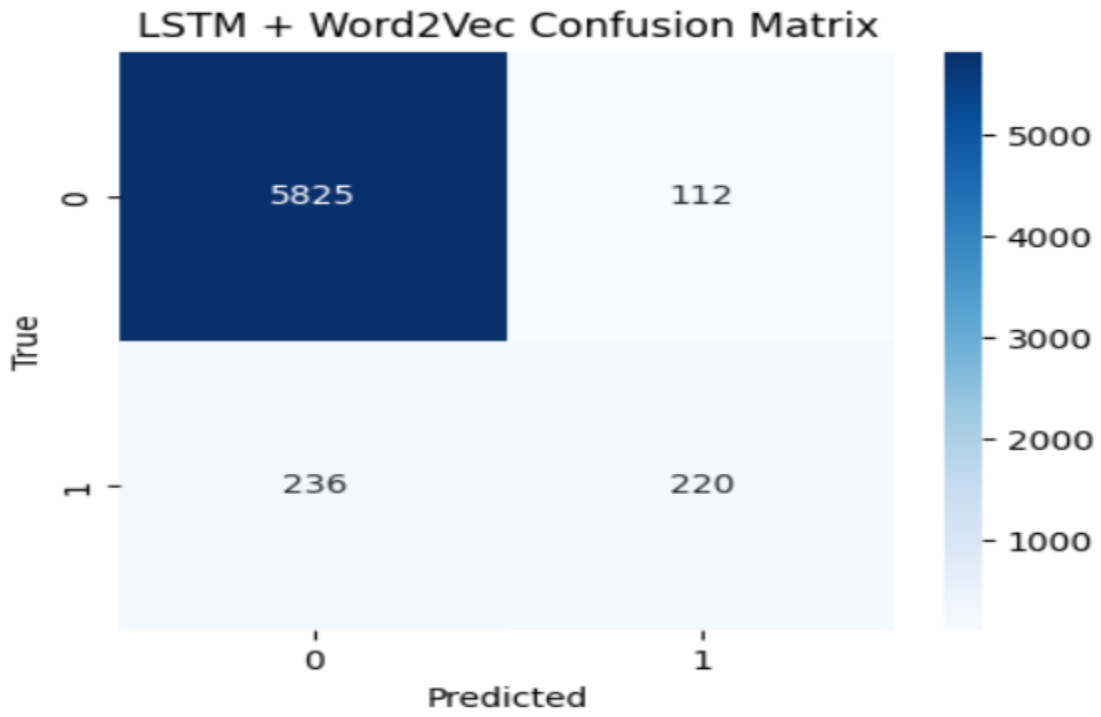

Figure 6: LSTM + Word2vec Confusion Matrix

The embeddings of TheWord2Vec were obtained from the Google News pretrained model. generating 300-dimensional vectors for the words in the vocabulary.

3.3 Loss Function

Binary cross-entropy was employed as the loss function convenient for the binary classification tasks. The choice of binary cross-entropy, or log loss as a loss function came from its applicability in binary classification issues (offensive (1) or non-offensive (0) tweets), where this is the most suitable approach. This loss function calculates the difference between true labels and the predicted probabilities, punishing deviations of class from predictions.

3.4 Optimizer

The Adam optimizer was applied to all of the models because of its adaptive learning rate. and efficiency of handling sparse gradients. The Adam (Adaptive Moment Estimation) optimizer was used for all the models because of its efficiency and robustness in optimizing the deep learning models. Adam inherits the advantages of momentum-based methods as well as adaptive learning rates, and is suitable for dealing with sparse gradients and noisy data that are typical to text classification problems with high-dimensional embeddings.

3.5 Hyperparameters Key hyperparameters included

- **Embedding dimensions:** 100 (RNN, LSTM), 300 (Word2Vec).
- **Batch size:** 32.
- **Epochs:** 10 (early stopping on validation loss).
- **Learning rate:** Default Adam rate (0.001).

4. Experiments and Results

4.1 RNN vs. LSTM Performance

The accuracy obtained during the tests by the Simple RNN and the LSTM were 93.43% and 93.38% respectively. 94.04%. The LSTM also performed better than the RNN because of its capability of learning long term dependencies and the vanishing gradient problems. The LSTM with Word2Vec embeddings obtained the best accuracy of 94.56% which is due to pretrained semantic representations. Validation loss trends showed that the models based on LSTM showed Industries was more stable convergence than the RNN.

4.2 Computational Efficiency

Training was carried out on Google Collab using GPU support. The Simple RNN had faster training times (approximately 19–21 seconds per epoch), than the LSTM (24–42 seconds per epoch) because of its less complicated structure. The Word2Vec model had comparable training times as those of the LSTM (17–21 seconds per epoch) but did more memory for 300-dimensional embedding matrix. All models were computationally feasible on standard hardware.

4.3 Training with Different Embeddings

The LSTM with the random embeddings (trained during model fitting) provided 94.04%. accuracy with the scores of 94.56% achieved after the application of the Word2Vec embeddings. The pretrained embeddings improved the generalizing capacity of the model by giving it more robust word representation, especially for low frequency words.

Model	Accuracy	Loss	Training Time
RNN	0.93%	0.0170	~43 sec
LSTM	0.94%	0.0286	~45 sec
LSTM with Word2Vec	0.94%	0.1201	~28 sec

4.4 Model Evaluation

Evaluation metrics included:

- Accuracy: LSTM +Word2Vec (94.56%)> LSTM (94.04%) > Simple RNN (93.43%).
- Confusion Matrix: LSTM + Word2Vec model demonstrated less false positives and false negatives with 98% recall for non-offensive tweets and 48%. recall for offensive tweets.
- Precision, Recall, F1-Score: The weighted accuracy in the LSTM + Word2Vec model was. F1-score of 0.94, with offensive tweets having a higher precision (0.66), as compared to the defensive tweets (0.34). to the RNN (0.55) and LSTM (0.59).

The confusion matrices and the classification reports demonstrated the models' good performance. performance on non-offensive tweets with worse recall for the offensive ones, probably due to class imbalance.

5. Conclusion and Future Work

This project showed that LSTM-based models, and in particular pretrained models, were good. Word2Vec embeddings, excel at classifying the tweets to the set of offensive and not offensive categories. Their results reach out to the accuracy of up to 94.56%. The LSTM outperformed the Simple RNN overcame vanishing gradient problem, Word2Vec embeddings helped in generalization. Lack of recall among offensive tweets was one of the limitations. because of the class imbalance and possibility of overfitting in further epochs.

Future work could involve:

- Solving class imbalance with methods such as oversampling or class weighting.
- Investigating the transformer-based models such as BERT for superior performance.
- Employing larger and a wider range of datasets to increase robustness.
- Hyperparameters tuning using Grid search. These refinements may increase the applicability of the model on real-world content moderation systems.

References

Hochreiter, S., & Schmidhuber, J. (1997). Long Short-Term Memory . *Neural Computation (MIT Press)*, 1735–1780.

Zhang, X., Zhao, J., & LeCun, Y. (2015). Character-level Convolutional Networks for Text Classification. *Advances in Neural Information Processing Systems (NeurIPS)*, 649–657.