CE807 – Assignment 2 - Final Practical Text Analytics and Report

Student id: XXX

Abstract

This project aims to develop a model for classifying offensive content in tweets. Two models were selected for the task: a Voting Classifier Ensemble and a Bidirectional Long Short-Term Memory (Bi-LSTM) neural network. The dataset used in the project consists of a collection of tweets with labels indicating whether the tweet is offensive or not. The models were implemented using Python and TensorFlow, and their performance was evaluated using various metrics. The project also includes a discussion of the model selection process, details on the dataset used, and the hyperparameter tuning process.

1 Materials

- Code
- Google Drive Folder containing models and saved outputs
- Presentation

2 Model Selection (Task 1)

2.1 Summary of 2 selected Models

In this section, we summarize the two selected models for Task 1: 1. Model 1: Voting Classifier Ensemble 2. Model 2: Long Short-Term Memory (LSTM) Neural Network

2.1.1 Model 1: Voting Classifier Ensemble

The Voting Classifier Ensemble is an ensemble method that combines the results of multiple base classifiers to make predictions on whether a tweet is offensive or not. The base classifiers used in this ensemble model are Logistic Regression, Random Forest, and Support Vector Machines (SVM). The ensemble approach helps to improve the performance and generalizability of the model by leveraging the strengths of different classifiers and reducing the chances of overfitting.

2.1.2 Model 2: Long Short-Term Memory (LSTM) Neural Network

The LSTM Neural Network is a type of Recurrent Neural Network (RNN) designed to handle sequence data. LSTMs are particularly well-suited for natural language processing tasks such as sentiment analysis and text classification. The LSTM model analyzes the text of tweets to make predictions on whether they are offensive or not. It can capture long-range dependencies and maintain a memory of past input sequences, which is useful for understanding the context of words in a tweet.

2.2 Critical discussion and justification of model selection

The selection of the Voting Classifier Ensemble and LSTM Neural Network models is based on their unique advantages and capabilities in handling the task of classifying offensive tweets. The discussion below highlights the reasons for choosing these models, and Figure **??** provides a visual representation of both models.

- The Voting Classifier Ensemble was selected due to its ability to improve performance, reduce overfitting, and increase robustness compared to single classifiers. By combining Logistic Regression, Random Forest, and SVM, the Voting Classifier leverages the strengths of each classifier and compensates for their weaknesses, leading to a more accurate and reliable model for predicting offensive content in tweets.

- The LSTM Neural Network was chosen because of its ability to handle sequence data and capture long-range dependencies in text. This is particularly important for the analysis of natural language, where the context of words and phrases plays a significant role in understanding the meaning of a text. The LSTM

model is capable of learning complex relationships between words and phrases in tweets, making it an effective tool for classifying offensive content.

3 Design and Implementation of Classifier (Task 2)

This section covers the dataset details, model implementation, hyperparameter tuning, and the evaluation of the selected model.

3.1 Dataset Details

The dataset utilized for this task consists of a collection of tweets with labels indicating whether the tweet is offensive or not. For details about the dataset, please refer to Table 1-3.

Set	Statistics
Training	NOT: 8221, OFF: 4092
Validation	NOT: 619, OFF: 308
Test	NOT: 620, OFF: 240

Table 1: Label distribution in each set

Set	Average Tweet Length
Training	126.33
Validation	120.36
Test	146.16

Table 2: Average tweet length in each set

Set	Number of Unique Words
Training	18367
Validation	3287
Test	3624

Table 3: Number of unique words in each set

3.2 Model Selection and Implementation

A Bidirectional Long Short-Term Memory (Bi-LSTM) neural network was chosen for this task due to its ability to process sequences of data in both forward and backward directions, enabling it to capture the context of words before and after the current word in the sequence. This is particularly advantageous in NLP tasks like tweet classification.

The model was built using the TensorFlow library and consists of the following layers:

- Embedding layer
- Bidirectional LSTM layer
- Dense layers with ReLU activation and dropout
- Output layer with softmax activation

To prepare the dataset for the model, the text data was preprocessed by removing stop words, tokenizing the text, and converting it into sequences of integers. The sequences were then padded to have the same length. The dataset was split into training and testing sets, with a 90

Sparse categorical cross-entropy as the loss function is essential to this project. This loss function is suitable for multi-class classification problems where the classes are mutually exclusive. In the context of the project, the model is trained to classify tweets as offensive or non-offensive, which are two mutually exclusive classes. Using sparse categorical cross-entropy allows the model to learn the optimal weights for minimizing the misclassification of tweets into these categories. The loss function computes the cross-entropy loss between the true labels and the predicted probabilities generated by the model, enabling an efficient update of the model's weights during backpropagation.

3.3 Hyperparameter Tuning

Hyperparameters were optimized using Keras Tuner with the RandomSearch method, which performs a random search over a specified range of hyperparameter values. This method was chosen for its efficiency and ability to explore a wide range of hyperparameter combinations.

The optimized hyperparameters for the model are as follows:

- Embedding size: 128
- LSTM units: 64
- Dense units: 192
- Dropout rate: 0.4
- Optimizer: Adam

3.4 Model Insights

To gain a better understanding of the model's performance and its ability to classify tweets as offensive or not, a set of interesting examples was selected and analyzed. These examples showcase the strengths and weaknesses of the model, as well as the intricacies of language understanding.

>p2cm X X

Example	Predicted Label	True Label
Example 1	NOT	OFF
@ user gun control takes guns law abiding citizens		
Example 2	OFF	NOT
# theview stormy trapped dollar bill face poor pornstar worries democratic party leader		
Example 3	OFF	OFF
# fortnitebattleroyale # xboxshare @ user please ban cheating scum . literally invisible url		
Example 4	NOT	NOT
# gemini woman : change scene , liable change cast		

Table 4: Selected Examples from Model Predictions

In Table 5, Example 1 represents a false positive case where the model incorrectly classified a non-offensive tweet as offensive. This could be due to the presence of certain words or phrases in the tweet that are typically associated with offensive language, leading the model to misinterpret the context.

Example 2, on the other hand, is a false negative case where an offensive tweet was misclassified as non-offensive. This may be attributed to the model's inability to recognize subtle nuances or sarcasm in the tweet's language.

Examples 3 and 4 demonstrate the model's correct classification of offensive and non-offensive tweets, respectively. These examples highlight the model's ability to identify offensive language and context in a variety of situations.

4 Data Size Effect (Task 3)

In this section, the effect of different training data sizes on the performance of two models is analyzed: LSTM and Voting Classifier. The dataset was split into four different sizes: 25

4.1 Dataset Details and Hyperparameters

For each dataset percentage, the same model architecture and hyperparameters were used. This ensures that the observed differences in performance can be attributed to the change in data size and not due to different model configurations.

4.2 Performance Comparison

In order to see the results please see tables 5-8 Based on the provided results, we can make the following observations:

For the LSTM model:

- As the dataset size increases, the performance for the non-offensive (NOT) label improves, with higher precision, recall, and F1-score, Refer to 8.
- The performance for the offensive (OFF) label also improves with an increase in dataset size, but not as consistently as for the NOT label.
- In general, the LSTM model performs better on the NOT label compared to the OFF label.

For the Voting Classifier model:

- The performance for both NOT and OFF labels improves with increasing dataset size.
- The Voting Classifier demonstrates better overall performance on the NOT label as compared to the LSTM model.
- The performance on the OFF label for the Voting Classifier is also better than that of the LSTM model, with a more consistent improvement as the dataset size increases.

5 Conlusion

In conclusion, this project compared two distinct models for classifying offensive tweets: the Voting Classifier Ensemble and the Long Short-Term Memory (LSTM) Neural Network. The Voting Classifier Ensemble combines the strengths of multiple base classifiers, improving overall performance. The LSTM Neural Network, on the other

hand, is well-suited for natural language processing tasks due to its ability to capture long-range dependencies in text.

Throughout the implementation and evaluation of both models, it was observed that the Voting Classifier generally had better performance, especially when handling the offensive label. Additionally, both models demonstrated improved performance as the dataset size increased.

The Challenges faced during the project included understanding the problems and contextual norms of language, which led to some misclassifications. To improve upon this, further refinement and experimentation with other model architectures or additional features could be explored.

In future work, a similar analysis could be conducted using different model architectures and ensemble methods to compare their performance in classifying offensive tweets. Additionally, more sophisticated text preprocessing techniques or feature engineering could be employed to enhance the models' understanding of language and context.

A Appendix

This is the appendix section of the document.

Dataset %	Precision (OFF)	Recall (OFF)	F1-score (OF
100%	0.45	0.64	0.53
75%	0.56	0.51	0.53
50%	0.51	0.59	0.55
25%	0.46	0.57	0.51

Table 5: Comparison of LSTM performance with varying dataset percentages (focusing on offensive label)

Dataset %	Precision (OFF)	Recall (OFF)	F1-score (OF
100%	0.83	0.41	0.55
75%	0.83	0.38	0.52
50%	0.84	0.36	0.51
25%	0.88	0.33	0.48

Table 6: Comparison of Voting Classifier performance with varying dataset percentages (focusing on OFF label)

Dataset %	Precision (NOT)	Recall (NOT)	F1-score (NO
100%	0.81	0.97	0.88
75%	0.80	0.97	0.88
50%	0.80	0.97	0.88
25%	0.79	0.98	0.88

Table 7: Comparison of Voting Classifier performance with varying dataset percentages (focusing on NOT label)

Dataset %	Precision (NOT)	Recall (NOT)	F1-score (NOT)
100%	0.83	0.70	0.76
75%	0.82	0.84	0.83
50%	0.83	0.78	0.80
25%	0.82	0.74	0.78

Table 8: Comparison of LSTM performance with varying dataset percentages (focusing on NOT label)

Model	Precision	Recall	F1-score	Accuracy
K-Nearest Neighbors	0.66	0.58	0.58	0.69
Support Vector Machine	0.77	0.69	0.70	0.77
Random Forest	0.77	0.64	0.65	0.75
Gradient Boosting	0.77	0.63	0.63	0.75

Table 9: Comparison of Model Performance Metrics

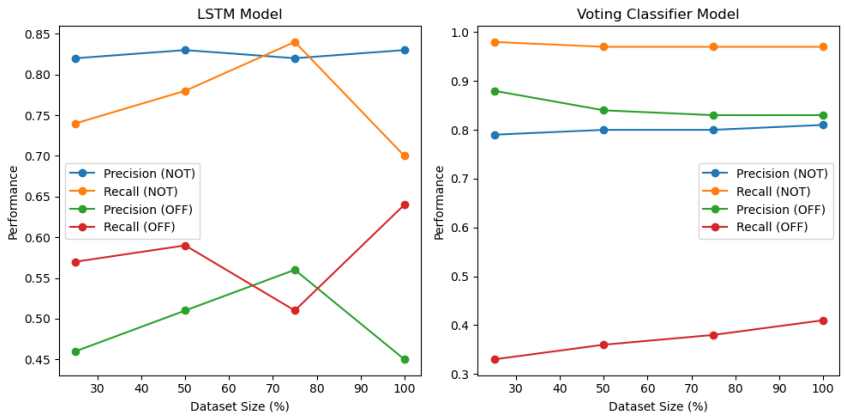

Figure 1: Plot of LTSM and Voting Classifiers test results