Text Analytics Report

Abstract

This report presents a study on the design and implementation of a classifier to detect offensive speech using the OLID dataset. Two classifiers were adopted: a Long-Short-Term Memory (LSTM) classifier and a BERT-sequence classifier. The LSTM classifier was chosen due to its ability to work with sequential data and its robustness to noisy data. The BERT-sequence classifier was selected for its state-of-the-art performance on a wide range of NLP tasks. The training of models involved data preprocessing, including data cleaning and data visualization, to understand the data and extract meaningful insights. The models were trained on different sizes of training data, and their performances were evaluated on test data. The results showed that the performance of both classifiers was affected by the size of the training data. The BERT-sequence classifier outperformed the LSTM classifier in most cases, but both classifiers had lower performance on offensive tweets due to class imbalance. Overall, this study demonstrates the effectiveness of LSTM and BERT-sequence classifiers in detecting offensive speech and provides insights into the impact of training data size on classifier performance.

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 assignment the two classifier which are adopted are: LSTM and BERT.

LSTMs based network was selected due to its architecture which can work with sequential data and is particularly effective when the order of the input matters. This is the case for many text classification tasks, where the meaning of a sentence or document can depend heavily on the order of the words. Secondly, LSTMs are robust to such noise, and can learn to extract meaning from noisy data, making them well-suited for real-world text classification tasks. A computationally friendly LSTM network could also be designed that will require less resources. Other network layers from TensorFlow were also added to LSTM network like dropouts, normalization and embedding layers. As our data contains tweets from the internet which contain noisy data, LSTM robust nature will be beneficial for this task.

BERT (Bidirectional Encoder Representations from Transformers) is a pre-trained natural language processing (NLP) model that has achieved state-of-the-art results on a wide range of NLP tasks, including text classification, named entity recognition, question answering, and more. For this assignment Bert for Sequence Classification from Huggingface (hug) was used which has classification head on its output layer to classify the labels. The idea was to load 'Bert-base-uncased' weights to the model which then will be finetuned/trained on the OLID dataset. The Bert base model was trained on next sentence prediction (NSP) and masked-language modelling (MLM) task on BookCorpus and English Wikipedia dataset which makes the model suitable for finetuning it on OLID dataset as it contains internet tweets of various sorts. In comparison to the designed LSTM model the Bert-sequence classifier has bigger neural architecture and requires GPU (Tesla T4 GPU was used for training the model) on Google Colab to complete the training in feasible time.

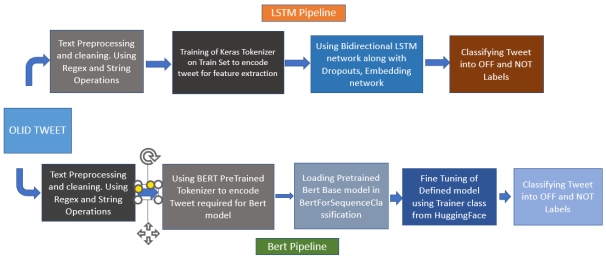

Figure 1: LSTM and BERT Pipelines

2.2 Critical discussion and justification of model selection

After fine tuning, the Bert classification model also gives fairly good results, but due to class imbalance, the overall performance for offensive tweets is low. Nonetheless, the Bert classifier's performance is better than LSTM classifier. The Pipelines of both Models are shown in Figure 1

3 Design and implementation of Classifiers (Task 2)

In this stage data preprocessing was carried out which involved data cleaning and data visualization to understand the data and extract meaningful insights from the data. The training file was used to generalize the understanding of the OLID dataset as it is assumed to represent the details of dataset which can then be extrapolated on test and valid data files.

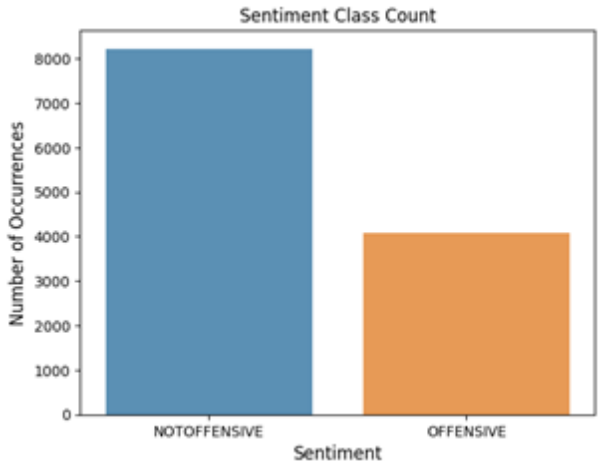

Figure 2: Bar Chart of Class Counts.

The analysis of the dataset showed that it had major class imbalance in it as the number of non-offensive tweets were twice the number of offensive tweets. Figure 2 shows this.
The sequence length of tweet was also plotted

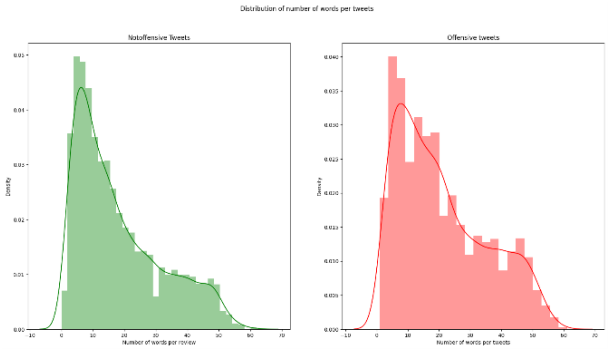

Figure 3: Tweet Length Distribution.

to determine the optimal size of sequence length which will be used as feature when designing the neural network. Figure 3 shows this.

Layer (type)	Output Shape	Param #
input_1 (InputLayer)	[(None, 55)]	0
embedding (Embedding)	(None, 55, 64)	1173760
batch_normalization (BatchNormalization)	(None, 55, 64)	256
dropout (Dropout)	(None, 55, 64)	0
conv1d (Conv1D)	(None, 51, 32)	10272
dropout_1 (Dropout)	(None, 51, 32)	0
max_pooling1d (MaxPooling1D)	(None, 25, 32)	0
bidirectional (Bidirectional)	(None, 25, 256)	164864
lstm_1 (LSTM)	(None, 64)	82176
dropout_2 (Dropout)	(None, 64)	0
dense (Dense)	(None, 1)	65

Total params: 1,431,393
Trainable params: 1,431,265
Non-trainable params: 128

Figure 4: LSTM Architecture

For LSTM classifier, the tokenizer from Keras' text preprocessing module library was used which was trained on training data to understand the vocabulary of the dataset and then can be used to encode the tweets into numerical representation that can be fed to the model (kag) . The LSTM

Model architecture is depicted in Figure 4. The parameters for model were:
Epochs: 5
Batch size: 32
Loss: Binary cross entropy
Optimizer: Adam (0.0005 LR)

For Bert-Sequence Classifier, Bert Tokenizer was used from Transformer library to encode the text in numerical representation for the model. During the training of the Bert classifier, it requires three inputs: Input ids which are the tokenized encoded ids that represent the word. Attention mask which determines the token which needs to be considered as it separates the original sequence from the padding. The target label on which the model will be trained to give prediction on.
The Bert classifier was trained using the trainer from Hugging Face, so default parameters were used except for:
Epochs:3
Batch size:32
Warm up steps :250
Weight decay : 0.01
Despite the issue of class imbalance affecting the overall performance for offensive tweets, the Bert classification model also shows satisfactory results. It outperforms the LSTM classifier model in terms of performance.

Dataset	Total	% OFF	% NOT
Train	12313	33.2	66.7
Valid	97	33.2	66.7
Test	860	27.9	72.1

Table 1: Dataset Details

Model	F1 Score
LSTM	0.729
BERT	0.85

Table 2: Model Performance

4 Data Size Effect (Task 3)

In this stage different train data sizes were used to train the classifiers, which were then evaluated on test data to determine the effect of the different data size on classifiers' performances. The training sizes used were: 25, 50, 75, 100 percent of the training data. This was achieved using the splitting functionality from Scikit-learn library. Figure 5 gives an overview of how splitting was done.

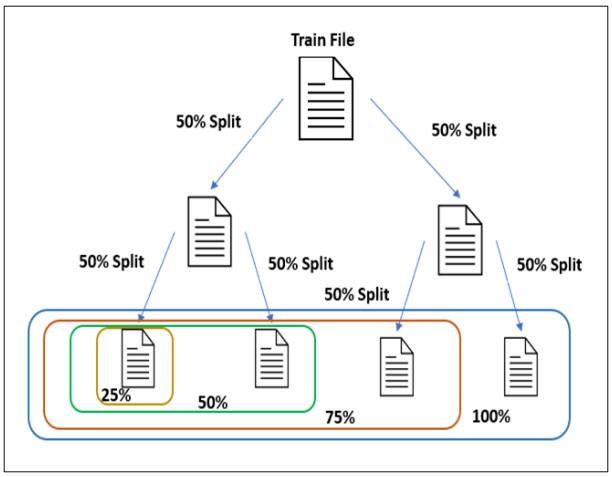

Figure 5: Dataset Split

Train Data Size	Class	precision	recall	f1-score	Overall Accuracy	Classifier Name
25%	Offensive	0.191666667	0.605263158	0.291139241	0.739534884	LSTM
25%	Notoffensive	0.951612903	0.75255102	0.84045584	0.739534884	LSTM
25%	Offensive	0.708333333	0.598591549	0.648854962	0.786046512	BERT
25%	Notoffensive	0.816129032	0.878472222	0.846153846	0.786046512	BERT
50%	Offensive	0.4625	0.460580913	0.461538462	0.698837209	LSTM
50%	Notoffensive	0.790322581	0.791599354	0.790960452	0.698837209	LSTM
50%	Offensive	0.679166667	0.673551719	0.676348548	0.818604651	BERT
50%	Notoffensive	0.872580645	0.875404531	0.873990307	0.818604651	BERT
75%	Offensive	0.520833333	0.462962963	0.490196078	0.697674419	LSTM
75%	Notoffensive	0.766129032	0.805084746	0.785123967	0.697674419	LSTM
75%	Offensive	0.608333333	0.737373737	0.666666667	0.830232558	BERT
75%	Notoffensive	0.916129032	0.858006042	0.886115445	0.830232558	BERT
100%	Offensive	0.466666667	0.598930481	0.524590164	0.763953488	LSTM
100%	Notoffensive	0.879032258	0.809806835	0.843000773	0.763953488	LSTM
100%	Offensive	0.591666667	0.797752809	0.679425837	0.844186047	BERT
100%	Notoffensive	0.941935484	0.856304985	0.897081413	0.844186047	BERT

Figure 6: Split Size Comparison

The results showed that when using a training size of 25 percent, the performance of the LSTM classifier was greatly reduced for both offensive and non-offensive classes. Similarly, the performance of the BERT classifier was also decreased primarily for the 'offensive' class.
Using 50 percent of the training data for training the models resulted in an increase in their performance. The increase in performance was more significant for the LSTM model than the BERT model from the previous data size.

Data %	Total	% OFF	% NOT
25%	3078	33.23	66.76
50%	6165	33.23	66.76
75%	9234	33.23	66.76
100%	12313	33.23	66.76

Table 3: Train Dataset Statistics of Different Size

Example Id	GT	BERT(100%)	LSTM(100%)
15923	OFF	OFF	NOT
27014	NOT	NOT	NOT
30530	NOT	NOT	NOT
13876	NOT	NOT	NOT
60133	OFF	OFF	NOT

Table 4: Comparing two Model's using 100% data: Sample Examples and model output using Model 1 & 2. GT (Ground Truth) is provided in the test.csv file.

5 Summary (Task 4)

5.1 Discussion of work carried out

In this study, we compared the performances of two state of the art models for the task of sentiment classification. We compared the models on different splits of data to see if the splitting the training data effected the scores.

5.2 Lessons Learned

It was noticed that the effectiveness of both designed classifiers was heavily influenced by the data quality. The class imbalance present in the dataset could lead to biased models and reduced accuracy. Additionally, the size of the training data was found to have a significant impact on the model's performance. As the training data size increased, the F1 score of both classifiers improved.

References

BERT — huggingface.co. https://huggingface.co/docs/transformers/model_doc/bert#transformers.BertForSequenceClassification. [Accessed 24-Apr-2023].

IMDB sentiment analysis - EDA, ML, LSTM, BERT — kaggle.com. https://www.kaggle.com/code/derrelldsouza/imdb-sentiment-analysis-eda-ml-lstm-bert/notebook. [Accessed 24-Apr-2023].