CE807 – Assignment 2 - Final Practical Text Analytics and Report

Student id: 2201829

Abstract

The proliferation of offensive language in online media raised concerns about an effective method for automatically detecting offensive language. This study mainly focussed on the classification of offensive language classification using the OLID dataset, which is a benchmark dataset for offensive language classification tasks. The given data set has been preprocessed and converted to the required format passed to the model for training and validated the trained model using the validation dataset and saved to the trained model to respective directories. The model has been loaded from the drive and evaluated using the test dataset. Two popular models Bidirectional Long Short-Term Memory (BiLSTM) and ELECTRA have been chosen for the experiment and achieved the highest accuracy where ELECTRA outperforms BiLSTM in both accuracy and f1 score.

1 Materials

This section contains all the links to the code, google colab and google drive where the program file is saved.

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

2 Model Selection (Task 1)

2.1 Summary of 2 selected Models

Recurrent Neural Network BiLSTM and pre-trained transformer model ELECTRA are the two models I have selected for training the model. Both models have been known for state-of-the-art performance in the OLID dataset for offensive language classification. A brief description of both models is as follows:

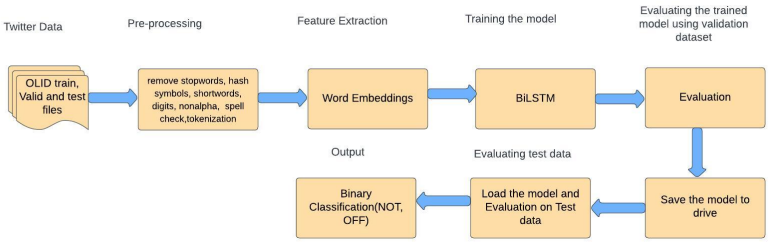
BiLSTM - Bidirectional Long Short-Term Memory, is a type of recurrent neural network architecture used for sequence modelling tasks. It consists of 2 parallel layers. One layer will process the input sequence in forward order and the other in backward order. This helps the model to capture both past and future contextual information in the input sequence. It is also capable of capturing long-term dependencies and is mainly used in NLP tasks.

ELECTRA - was created with the aim of efficiently and effectively training large-scale language models. It provides a generator-discriminator arrangement in which the generator swaps out tokens in the input sequence while the discriminator determines if the swapped tokens are genuine or artificial. The model gains the ability to extract precise contextual information from the input sequence, which is helpful for tasks like classifying offensive language. By utilising its effective training and discriminator-based learning mechanism, ELECTRA has been demonstrated to attain state-of-the-art performance on a variety of natural language processing (NLP) benchmarks, including offensive language classification.

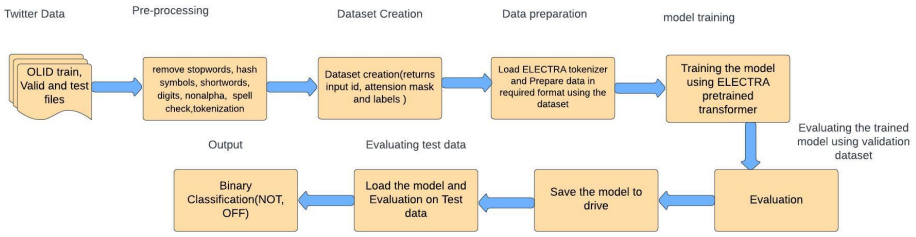
2.2 Critical discussion and justification of model selection

- I have tried 7 models, Naive Bayes, XG Boost, Logistic Regression, DistilBert, Roberta and LSTM for the experiment, BiLSTM and ELECTRA outperform other models with the best accuracy and f1 score. Table 1 contains the details of results obtained from training different models with full training data.

- BiLSTM is known for its ability to effectively model sequential data, which makes it a good choice for offensive language classification. From Table.1 it is clear that BiLSTM has an accuracy score of 80.31% and F1 Score of 72.40% compared to the other 5 models other than ELECTRA, indicating that it performed



(a) BiLSTM



(b) ElECTRA

Figure 1: Comparison of BiLSTM and ElECTRA model Pipelines

well in correctly predicting the data in the OLID dataset.

- Electra is a state-of-the-art language representation model which is pre-trained on a large corpus of text data. It used masked language modelling objectives. It shows superior performance compared to various NLP tasks, even for offensive language classification. The results in Table.1, we can interpret that ELECTRA is the best-performing model among the others with the highest accuracy score of 83.89% and F1 score of 78.99%.

Based on the results from the experiment, BiLSTM and ELECTRA have the highest accuracy and F1 Score compared to Naive Bayes, XG Boost, Logistic Regression, DistilBERT and Roberta. This indicates that ELECTRA and BiLSTM are more capable of accurately predicting offensive language in the OLID dataset. Also, both of these models are known for the ability to model sequential dependencies and capture contextual representation, which is one of the most important parts in text classification tasks. Based on the superior performance and capabilities in prediction and considering the above point I have chosen both BiLSTM and ELECTRA as my models.

Model	Accuracy	F1 Score
Naive Bayes	79.88	56.42
XG Boost	78.95	44.98
Logistic Regression	79.56	51.14
DistilBERT	78.23	66.55
Roberta	78.56	66.98
BiLSTM	**79.96**	72.59
ELECTRA	**83.60**	78.75

Table 1: Model Performance

3 Design and implementation of Classifiers (Task 2)

This section contains the details of the dataset, and implementation steps of the model including hyperparameter tuning and other important steps. Along with this, some comparisons of interesting examples based on the output of both models are also included.

- ***Dataset*** - Offensive Language Identification Dataset(OLID) dataset has been used as the dataset for this classification task. We have three datasets mainly- Train, valid and test datasets where the Train dataset is used for training the model and the valid dataset set is used to validate the trained model. The test dataset will be used to test the trained after loading the model from the saved directory. The details about the 3 datasets were

described in detail in Table 2. From the initial analysis, the dataset seems to be imbalanced and the majority of the data in all the datasets belong to class 'NOT' with percentages around 66% and 33% 'OFF' in the train and validation dataset. (Zampieri et al., 2019)

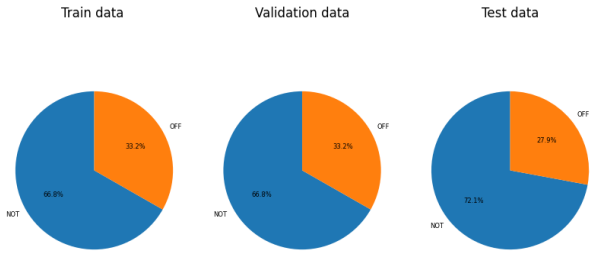


Figure 1. Pie-chart representation of the percentage of each class

- ***Model implementation*** - BiLSTM and ELECTRA are the two models used for the implementation. The model implementation of both models has been described below:

1. **BiLSTM** - The main parts of implementing BiLSTM model are as follows:
a. *Preprocessing* - The preprocessing portion of the code involves receiving data from a CSV file, handling diacritics, and deleting mentions, URLs, and hashtag symbols using regular expressions. Additionally, it involves tokenizing tweets and deleting stop words, short words, numerals, and non-alphanumeric characters. To create the string representation, label values are changed, and tokens are rejoined. A data frame containing the transformed data will be returned.
b. *Tokenization* - The data in text format will be tokenized using the 'Tokenizer' class from Keras. The number of words is set to 10000 indicating only the most frequent 10000 words will e kept in the tokenized vocabulary. The out-of-vocabulary token is also defined in the code for the words that are not present in the vocabulary.
c. *Padding* - The tokenized will be padded to a fixed length of 500 to ensure that all input sequences have the same length, which is important in LSTMS models.
d. *Model Architecture* - The model has been created using the Sequential class from Keras. The model consists of an embedding layer with a maximum feature value of 10000, embedding dimension of 128 and input length of 500. The BiLSTM layer has 128 units with a dropout rate of .3 to prevent overfitting. The activation function used in the output layer is sigmoid for binary classification.
e. *Model Training* - The model has been compiled using binary cross-entropy loss function and Adam optimizer. The evaluation matrices used during the training include - binary accuracy, precision and recall. The model has been trained using the processed data and validated using the processed validation data.
g. *Fine Tuning* - The model has been fine-tuned by training the model with different values for learning rates, maximum length, epochs, batches etc. The parameters have been finalized with a combination of values which gave higher accuracy and f1 score.
h.*Saving the model* - The trained model, tokenizers and vectorizers have been saved to respective directories so that they can be used and reused for testing.
i. *Evaluation* - The tokenizer and model have been loaded from the directory and tokenized the test data using the tokenizer and predicted the output using the model loaded from the directory. The predicted data has been appended to the existing test file and saved to the respective directory as an output.csv file.
2. **ELECTRA** - ELECTRA given the best results over BiLSTM and other models I have tried. Following steps contains the implementation of ELECTRA hugging face model for offensive language classification:

a. *Preprocessing* - BiLSTM and ELECTRA have the same cleaning process which is explained above. After cleaning of data, the dataset needs to be created in the format which the ELECTRA model can use for training. For that, by taking text, label, tokenizer and maximum length as input, the tokenizer will tokenize the text, add special tokens, and create a tokenized representation of PyTorch tensors. The tokenized data with labels will be returned and it will be used as the input for the model for training and validation dataset.
b. *Loading ELECTRA tokenizer and model* - From hugging face model hub, load tokenizer

and model. Define the training arguments.
c. *Model Training* - The model has been trained with the dataset created and validated using the validation file.
d. *Fine Tuning* - Fine-tuning has been done with different values for learning rate, number of the epoch, batch sizes etc and I have chosen the parameters which have given better accuracy and performance.
e. *Save the model* - Trained model has been saved to the respective for reusability
f. Evaluation- The model and tokenizer saved in the drive have been loaded and used for prediction and evaluated the model using the classification report, confusion matrics and rocscore.

- ***Comparison of model performance*** - Following are the important comparison identified after the evaluation with the test dataset:
From the classification report, it is clear that ELECTRA outperforms BiLSTM in several performance metrics. ELECTRA has the highest test accuracy of 83.6% compared to BiLSTM with an accuracy of 80.2% shows that ELECTRA has overall better accuracy. While considering the precision value, for both classes ELECTRA has the highest score indicating the ability of ELECTRA to correctly predict positive and negative instances. Considering the recall value, BiLSTM has a higher recall value for 'NOT' while ELECTRA has a higher recall value for predicting 'OFF', indicating better capability to capture the positive instance of classes. ELECTRA also has higher F1 scores for both classes showing a better balance between precision and recall.
Based on the evaluation matrices, we can conclude that ELECTRA performs better than BiLSTM in terms of accuracy, F1-score, precision, and recall suggesting that ELECTRA will be better suitable for this classification task.

- Table.3 contains some interesting examples of comparison of prediction obtained from both models. In the first example, we can see that there is no hate words have been used and our BiLSTM model failed to predict it, but the ELECTRA model predict it correctly. Considering the last example of Table 5, we can see that there is hate or offensive word is there, but it is not considered as offensive, both of our models failed to predict it.

Dataset	Total	% OFF	% NOT
Train	12,314	33.23	66.77
Valid	928	33.29	66.81
Test	861	27.99	72.12

Table 2: Dataset Details

Model	Accuracy	F1 Score
BiLSTM	**80.31**	72.40
ELECTRA	**83.89**	78.99

Table 3: Model Performance

4 Data Size Effect (Task 3)

This section contains the overall effect of data size in the performance of the model.

- Table 4. depicts the number of data records and percentage of each class in the dataset of 25%, 50%, 75% and 100% of data. I have made sure that all the data in 25% is included in 50% and all data in 50% is included in 75% of data and also manually validated using Excel.

- I have tried changing the hyperparameters for datasets of different data sizes and found that the number of epochs needed for them varies. Smaller datasets required more number of epoch than larger ones. I have also noticed the learning rate also has an effect on the performance of the model of different sizes. But for the comparative study, I have used the same parameters for all the data sizes.

- *Critical Discussion of performance compared to different data sizes* - From the experiment with different data sizes it is clear that datasets with larger datasets give more accuracy compared to the smaller ones. For BiLSTM there is a huge difference in almost all evaluation matrices of 25% dataset and 100%. While for ELECTRA there is not much difference in the accuracy of models of different data sizes small as 2% difference. From this, we can say that for BiLSTM there is a huge impact on a number of training data, while ELECTRA

can perform well and can give high accuracy for small datasets as well. The detailed comparison of performance measurements of each dataset is marked for both validation and test data.

- Figure 3 and Figure 4 depict the graph of f1-score vs data size. From the two graphs, we can see that the f1-score is increasing with an increase in data size. Only in one case of the BiLSTM model the f1-score got decrease, Otherwise, it is consistently increasing with an increase in data size. Also, we can see that accuracy of test data is more compared to validation data.
  On the other hand for ELECTRA, there is not much difference in the f1-score of 25% and the other datasets. There is only a negligible difference. From this, we can interpret that for the ELECTRA model, the data size has less impact on overall performance. Additionally, if we analyze the figure.4 we can see that the f1-score of the test data set is almost consistent for all the data sizes. So, one more thing we can learn from this is quality of data will also matter in creating a good model. So we can conclude that data size and quality of data have is directly proportional to creating an efficient model.

- Table 6 and 7 contains the examples from the test file, which has the predictions done by BiLSTM and ELECTRA on trained with different data sizes. Considering the last example of both models, we can see that all models failed to predict it as 'NOT' as it contains words which belong to the hate or offensive category but considering the semantic meaning it won't be an offensive language. Consider the first example in both table it does not contain the any hate word but it semantic meaning makes it offensive, it can be considered to an ambiguous category.

Data %	Total	% OFF	% NOT
25%	3078	50.00	50.00
50%	6156	50.05	49.95
75%	9236	44.30	55.70
100%	12314	33.23	66.76

Table 4: Train Dataset Statistics of Different Size

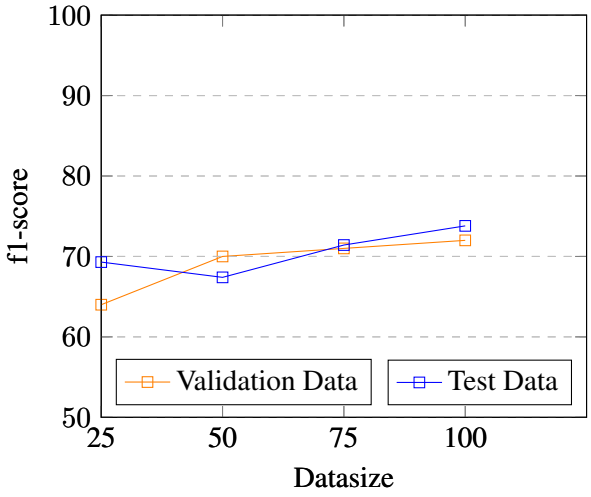


Figure 3. BiLSTM model data-size comparison

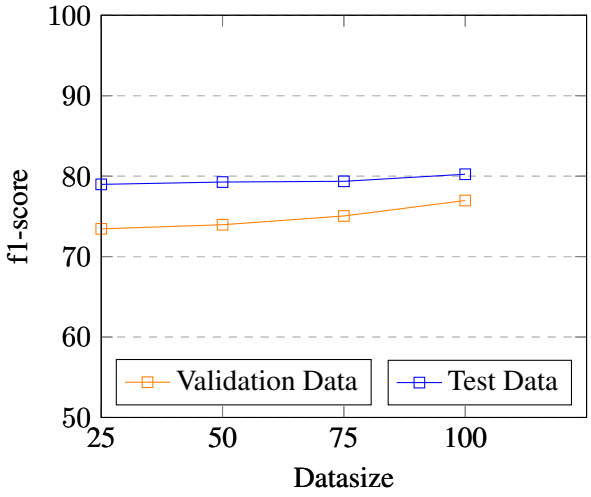


Figure 4. ELECTRA model data-size comparison

You need to use Table 6 7 and 5 to compare the model's output and provide exciting insights. Note that you can't have some examples in all tables.

5 Summary (Task 4)

This section contains the summary of all the work carried out so far in this experiment along with lessons learned after completing the experiment.

5.1 Discussion of work carried out

The main objective of this experiment is to create an offensive language classification model using 2 models and compare the result. And, also to find the impact of data size in creating an efficient model. I have created two classifiers using BiLSTM and ELECTRA. Both of them performed well, while ELECTRA outperforms BiLSTM in all evaluation matrices. Also, it gives a state-of-the-art accuracy with an f1-score of 80.2%, an accuracy of 84% and a roc score of 79. Also studied the

Example %	GT	M1(100%)	M2(100%)
#Conservatives @USER - You're a clown! URL	OFF	NOT	OFF
@USER Antifa has TS level influence. It's scary.	OFF	NOT	NOT
And dicks. URL	NOT	NOT	OFF
@USER Be sure to send out the left's antifa thugs.	NOT	OFF	OFF
Are you fucking serious? URL	NOT	OFF	OFF

Table 5: 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.

Example %	GT	M1(25%)	M1(50%)	M1(75%)	M1(100%)
#Conservatives @USER - You're a clown! URL 1	OFF	NOT	NOT	OFF	NOT
@USER Antifa has TS level influence. It's scary.	OFF	NOT	NOT	NOT	NOT
And dicks. URL	NOT	NOT	NOT	NOT	NOT
@USER Be sure to send out the left's antifa thugs.	NOT	NOT	NOT	OFF	OFF
Are you fucking serious? URL	NOT	NOT	OFF	OFF	OFF

Table 6: Comparing Model Size: Sample Examples and model output using Model 1 with different Data Size

Example %	GT	M2(25%)	M2(50%)	M2(75%)	M2(100%)
#Conservatives @USER - You're a clown! URL 1	OFF	OFF	OFF	NOT	OFF
@USER Antifa has TS level influence. It's scary.	OFF	NOT	NOT	NOT	NOT
And dicks. URL	NOT	OFF	OFF	OFF	OFF
@USER Be sure to send out the left's antifa thugs.	NOT	OFF	OFF	NOT	OFF
Are you fucking serious? URL	NOT	OFF	OFF	OFF	OFF

Table 7: Comparing Model Size: Sample Examples and Model output using Model 2 with different Data Size

impact of data size in creating the model, and concluded that BiLSTM shows much more difference in the f1-score of different data sizes compared to ELECTRA. ELECTRA gives an approximately consistent performance in the test dataset, while a maximum 6% between 25% and 100% in validation datasets. Also, we can see that the f1-score of the validation set is lower than the f1-score of the test dataset in both models. From that, we can also assume that the quality of data also matters in getting good accuracy. Test data might be more similar to train data or validation data may have additional features that the model failed to capture.

5.2 Lessons Learned

From this experiment, I got a chance to learn two new models, BiLSTM and ELECTRA, which are known for state-of-the-art performance in text classification tasks. I have learned to implement it and fine-tune it to get better accuracy and improve performance.
Difference preprocessing steps have been explored and applied in the dataset to get better performance. Also, noticed that the computational time and resources required for both of the models are more compared to other models. ELECTRA took more training than BiLSTM but gives high accuracy in performance compared to BiLSTM.
Data size impact has been studied and concluded that the data size has a direct impact on the BiLSTM model, while ELECTRA shows consistent performance in the test dataset makes me think that ELECTRA can perform well even with small datasets.

6 Conlusion

The study has been done for implementing a classification model for detecting offensive language using two state-of-the-art models BiLSTM and ELECTRA. Both of them performed well with high accuracy, but ELECTRA outperform BiLSTM in all evaluation matrices. Also done a comparative study to find the impact of data size on the performance of a model trained with 25%, 50%, 75% and 100% data. And find that BiLSTM has a direct impact on data size while ELECTRA has not had much impact and it almost consistently performed well for all the data sizes.

References

Marcos Zampieri, Shervin Malmasi, Preslav Nakov, Sara Rosenthal, Noura Farra, and Ritesh Kumar. 2019. Predicting the Type and Target of Offensive Posts in Social Media. In *Proceedings of NAACL*.