Assignment 1

Mehregan Nazarmohseni Fakori, Shafagh Rastegari, Fatemehzahra Ghafari Ghomi and **Mohammad Pourtaheri**
Master's Degree in Artificial Intelligence, University of Bologna
{ mehrega.nazarmohseni, shafagh.rastegari, fatemezahra.ghafari, mohammad.pourtaheri }@studio.unibo.it

Abstract

This project addresses the task of sexism detection on some English tweets, aiming to classify them as either sexist or not sexist. To achieve this, we employ two distinct approaches: a Long Short-Term Memory (LSTM) network and a Transformer model. Both methods are evaluated on a given dataset, with comparing their performance on provided metric analysis. Our results highlight the strengths and limitations of each model, which are 2 LSTM model an a Transformer model.

1 Introduction

Sexism is a widespread issue in online platforms, where harmful language spreads unfair treatment of genders and creates toxic environments. This study addresses Task 1 - Sexism Identification from the EXIST 2023 shared task, which involves classifying tweets as sexist or not sexist. Leveraging a subset of the EDOS dataset, we employ two approaches for this task: traditional LSTM (Long Short-Term Memory) networks and modern Transformer models. By comparing the performance of these models, this study aims to evaluate their effectiveness in identifying sexism in tweets. In section 2 shows how we preprocess data and how we create embedding matrix from pre-trained GloVe embedding. In section 3 describe the more detail of the models and preprocessing methods. In section 4 we show our result and analyze them.

2 System description

The system was designed to address sexism detection in English tweets, using a dataset biased toward the Not sexist class. At first, the datasets were cleaned by removing irrelevant elements like emojis, hashtags, mentions, and URLs to enhance text quality. Additionally, we adopted two other approaches for cleaning the dataset to have better data for training, correcting typos and lowercase text. In our models, for the Embedding Layer, we use GloVe embeddings of dimension 100 with some additional vectors randomly generated for Out-Of-Vocabulary words. In the Baseline model, the provided procedure is followed by a Bidirectional LSTM layer, a Dense layer with Sigmoid activation, and then a Dropout layer. The other LSTM model has been constructed by adding an additional LSTM layer to the Baseline model. For the Transformer, we use a model specifically trained for hate speech detection, namely twitter-roberta-base-hate.

3 Experimental setup and results

It is a classification task, so each tweet should have a label, but in the dataset there are six labels per tweet, so we use majority voting to achieve one label for each ones. The preprocessing contains loading data into Pandas DataFrame, cleaning data, and removing unwanted columns from the dataset as explained in previous section. After that, we loaded GloVe embedding with the provided dimension and created the vocabulary. We handled Out-Of-Vocabularly terms with random embedding and added special tokens for unknown(["UNK"]) and padding(["PAD"]) to the embedding matrix. We have defined the architecture of our two LSTM models, so we only need to assemble the layers. The activation function of the final Dense layer is Sigmoid to make sure that the output of the model is the probability of the input belonging to the class. We use BinaryCrossEntropy as a loss function. We use the Optuna package in the project to find the best parameters for the model. Monitored the accuracy metric during training, due to the impossibility of using F1-scores here. We chose Adam as an optimizer with the provided learning rate and weight decay. We trained the model for 3 seeds and stored quality metrics for each one. At the end, we tested our models on the validation set and measured the

	F1-score	F1 Standard Deviation
Baseline	0.788	0.022
Model 1	0.813	0.022
Transformer	0.838	0.367

Table 1: Comparison performances between all models.

quality of the networks by calculating the F1-score, accuracy, and loss. The overall performance of the models is considered by the macro average. In Table 1, we show that Model 1 works better than the Baseline model, based on our results. For error analysis, we display the misclassified samples, and create a confusion matrix, and a precision-recall curve for each model on each seed. For the Transformer, we use the existing model for training the data. It also runs with 3 seeds and for each, we compute the F1-score. The error analysis of the model is like the LSTM model.

4 Discussion

Our experiments confirmed that on the average of the F1-score among 3 seeds that run models, Model 1 performs better than the Baseline model between the LSTM models, and the Transform is better than those. In all three models, there were 10 tweets that predicted incorrectly, and there are 41 tweets that at least one model classify them incorrectly. It could be due to a number of factors, such as tweets being too short and not providing enough clues for accurate prediction, or the phrases in the tweets not appearing in the training data. As a result, we concluded that Out-Of-Vocabulary words with random values may lead to this problem.

5 Conclusion

This study investigated sexism identification on tweets using LSTM and Transformer models. Our results demonstrate that both approaches effectively classify sexist and not sexist tweets, while with Transformer showing higher accuracy due to their advanced architecture. We got some misclassified tweets discussed in the previous paragraph (see section 4). In conclusion, using advanced strategies based on string similarity for Out-Of-Vocabulary words rather than using random and a better approach for voting tweets between annotators may result in better performance.