Assignment 1

Leonardo Chiarioni, Shariq Mohd Ansari, Federico Faccioli
Master's Degree in Artificial Intelligence, University of Bologna
{ leonardo.chiarioni, mohdshariq.ansari, federico.faccioli2 }@studio.unibo.it

Abstract

With this report we aim to assess the ability of deep learning models to solve NLP tasks, more precisely a text classification task, by using different text processing techniques and different models. We observed how the quality of the text plays a role in the performance and how more complex models can understand more complex text, reducing the bias introduced by the data.

1 Introduction

The task we addressed consists in identifying sexism inside a dataset made of tweets. Given the sequential nature of the data, LSTMs and Transformers are good candidates for solving our task, by being able to capture long-term contextual dependencies in the text. Pre-processing of the text is also necessary in order to keep only relevant information for the classification tasks; Regular expression patterns are a standard way to identify and clean text by defining simple and standard text processing. Given its simplicity, this method is heavily based on hand-written rules and may not capture the inherent variety of expressions of the same concept.

The approach we followed consists of processing the tweets in 3 different ways, in order to create text with different "complexities". The tweets are then properly encoded and used to train 3 different architectures, with increasing complexity as well.

We run a total of 9 experiments, one for each combination of text and architecture. With our research we demonstrated how text processing influences the models variance and how more complex models can reduce bias introduced by dataset imbalance and how they can better understand more human-like text.

2 System description

In order to solve our tweets classification task, we implemented 2 custom architecture with Keras API and a pre-trained transformer architecture from HuggingFace API, using Python programming language. The custom models' implementation can be found under section 4 of the notebook. It is worth to say that the embedding layer can efficiently handle tweets of different sizes and can be fine-tuned during training. The classification layer uses a sigmoid function to compute the output, which will correspond to the probability of a tweet to be sexist (greater than 0.5) or not. The transformer architecture used, referred here as RoBERTa is the twitter-roBERTa-base-hate, publicly available on HuggingFace. It is pre-trained on 58M tweets and fine-tuned for hate speech detection.

3 Data

The dataset, already split in train, validation and test, contains english and spanish tweets classified by 6 annotators. The initial cleaning set up can be found in section 1 of the notebook. The cleaned dataset is unbalanced, with a bias towards the Non-Sexist class. The second step is text cleaning, which can be found in section 2 of the notebook. As third step, we prepared 3 different datasets according to different further processing techniques. We have:

- lemmatized tweet: lemmatization of the pre-processed tweet;
- lemmatized tweets without stop-words: lemmatization of the pre-processed tweet with stopwords removal;
- simple tweets: plain pre-processed tweet with no further processing

Fourth and final step, we built a vocabulary for each dataset and we encoded it using GloVE embedding, with an embedding size of 50. Due to the different processing techniques, the length of the

Dataset	Vocabulary Size	OOV Terms
lemmatized	8735	842 (9.64%)
no stopwords	8728	844 (9.67%)
simple	10493	845 (8.05%)

Table 1: Vocabulary sizes and oov terms.

Model	Dataset	Test F1	Val F1
Baseline	lemmatized	0.732	0.749
	no stopwords	0.727	0.706
	simple	0.731	0.752
Model1	lemmatized	0.777	0.799
	no stopwords	0.723	0.695
	simple	0.758	0.784
roBERTa	lemmatized	0.856	0.895
	no stopwords	0.813	0.841
	simple	0.845	0.889

Table 2: Model evaluation on the validation and test sets.

vocabulary and the out-of-vocabulary terms (OOV) differs from each of the dataset, as shown in Table 1. The number of OOV terms is almost unchanged; this suggest that most of them may come from misspellings and patterns not correctly detected by our simple cleaning process. It is worth to note that the embedding is used only to train the custom models, since the transformer architecture has its own vocabulary.

4 Experimental setup and results

With the datasets and the models ready we set up our experiments by training and evaluating every model with each of the dataset we built. The evaluation has been done both on the validation and the test set; since the dataset is unbalanced the evaluation metric used is the f1-score. For the custom models we used Adam optimizer with binary cross-entropy loss and a learning rate of 10^{-3}, while for the transformer we used AdamW optimzer with binary cross-entropy loss and a learning rate of 10^{-5}. By default, the HuggingFace transformers library implements a linear learning rate scheduler with weight decays=10^{-1}. For robust estimation, each of the experiment has been run with 3 different seeds (42,100,666) and average values were taken. The numeric results are shown in table 2, while the confusion matrices are accessible in Section 7 of our notebook.

5 Discussion

The results obtained with our experiments proved how data processing is critical in solving our tweet classification task; by looking at the results in Table 2, every model coherently reached the best performance in lemmatized tweets, with very close results for simple processed ones. Worst performances have been found on the lemmatized text without stopwords. Especially for the custom models, the fact that we use a trainable embedding layer with pre-initialized weights from GloVE definitely played a role. Increasing model complexity has successfully helped in improving the score with respect to our baseline and showed a better "understanding" of human-like text, as showed by results on simple processed tweets. The confusion matrices, available in Section 7 of our notebook, provide even better insight into what our models improved in; focusing on the datasets that provided the best results we can see how models with increasing complexity increased the percentage of positive (Sexist) predictions, reducing the bias introduced by the dataset's imbalance, and how the transformer architecture was able to reduce both false negative and false positive. For the error analysis, we analyzed some of the tweets at the end of the cleaning process that contained OOV terms and compared them with the original ones. The example shown in Section 7 of our notebook contains several "Hindi" terms and references to a very specific concept, for which there is no explicit information. Those terms are not part of the english language and especially of the GloVE vocabulary. Tweets like this inject irrelevant knowledge and add noise to the training process, making it difficult even for humans to classify the text.

6 Conclusion

With our research we showed how Recurrent Neural Networks are very good Deep Learning models to solve an NLP classification task, with transformer architecture being able to reduce dataset's imbalance and outperforming LSTM-based architectures. We showed how the cleaning process of the dataset is crucial and how simple regular expressions are powerful but at the same time are not able to capture all the variety of data representations and misspellings. In our results we expected to achieve lower performances on the simple processed tweets, while the results were comparable with the lemmatized ones. In order to

improve even more, we could have experimented more sophisticated data cleaning techniques, such as word separation (in case of word not spaced) and removal of repeated letters. For what concerns the architectures, we could have done a better tuning of hyperparameters and we could have experimented regularization techniques (dropout layers, learning rate schedulers) in order to reduce overfitting.

References