NLP: Assignment 1

Alberto Genovese, Simone Reale, and **Alessandro Tutone**
Master's Degree in Artificial Intelligence, University of Bologna
{ alberto.genovese5, simone.reale3, alessandro.tutone }@studio.unibo.it

Abstract

This work aims to address the **EXIST 2023 Task 1** on the detection of sexist behaviors, using information from tweets on the Internet. In the first part, two models were employed: a ***Bidirectional LSTM*** with only one LSTM layer and a second ***Bidirectional LSTM*** with twice the number of LSTM layers. The evaluation of the two models allowed us to compare their performance and their ability to solve the given task. In the second part, we adapted the ***Twitter-RoBERTa-base for Hate Speech Detection*** model, a *Transformer-based model*, in both its monolingual and multilingual versions. The evaluation obtained demonstrates the potential of these models to address sexism detection. However, the results also highlight the challenges and difficulties in understanding and accurately managing natural language.

1 Introduction

Sexism is a widespread issue in online social networks. Although human supervision has traditionally been the primary approach to dealing with this problem, thanks to the rapid advancement of AI, the detection of such harmful contents can now be automated. Several models such as *SVMs*, *RNNs*, *LSTMs*, and much more, are able to perform classification in various contexts. However, the nuanced nature of sexism, often expressed subtly or implicitly, poses further challenges for automated detection, showing that human expertise can still provide valuable feedback to refine AI models and ensure accurate detection of sexist content.

To deal with this problem, experiments have been conducted by evaluating both architectures introduced before on two languages, English and Spanish, with different setups. The *LSTM models (Baseline, Model 1)* are trained with three different random seeds *(42, 132, 1337)* and the evaluation is done both on the average performances of each model and on their comparison. On the other hand, the *RoBERTa-based and XLM-RoBERTa-based Transformer models* are run with the best hyperparameter setup found. Their evaluation is presented from two different points of view:

- generalisation capabilies against the LSTM architecture;
- the gap in performance between a monolingual model and a its multilingual version, finetuned on a single language.

What can be noticed from the experiments is that randomness is something that should not be overlooked, since changing the seed results in variations of the outcomes that cannot be neglected. In addition, input data and the way they are processed before pre-training or training in general play a crucial role, enhancing models' capabilities.

2 System description

The LSTM architectures are designed according to the original paper (Hochreiter and Schmidhuber, 1997), where the embedding layer is initialized with different static models for the two languages. ***GloVe*** is used for the English vocabulary and ***FastText*** (Bojanowski et al., 2017) for the Spanish one, unifying their way of managing unknown words, in order to make the comparison as fair as possible. Then, the hyperparameters selection follows the direction of the paper "*A comprehensive study of deep bidirectional LSTM RNNs for acoustic modeling in speech recognition*" (Zeyer et al., 2017). Regarding this research, better performances have been achieved by using the *NAdam (Nesterov-accelerated Adaptive Moment Estimation)* optimizer, proposed in the paper as the most efficient method on this type of architecture.

Conversely, the RoBERTa-based Transformer is trained in a more standardized way. Experimentally, it has been found out after many trials of freezing combinations of trainable weights that the behavior is quite the same as fine-tuning the entire network. Therefore, the simplest approach has been chosen to avoid increasing the overall complexity.

3 Experimental setup and results

For both the *Bidirectional LSTM* models, the following hyperparameters are used:

- *batch size:* **16**
- *training epochs:* **10**
- *hidden dimension:* **64**
- *optimizer:* **NAdam**
- *momentum:* **0.9**
- *metrics:* **accuracy** & **F1-score**
- *loss:* **binary cross-entropy**

On the other hand, slightly different hyperparameters are used for the *Transformer-based* model:

- *batch size:* **16**
- *training epochs:* **10**
- *learning rate:* **5e-6**
- *epsilon:* **1e-8**
- *metrics:* **accuracy** & **F1-score**
- *loss:* **binary cross-entropy**

Any hyperparameter that is not mentioned above is kept as default.

Model	**F1 val**	**Seed**	**F1 test**
Baseline LSTM	0.797	1337	0.732
Model 1 LSTM	0.815	42	0.776
Transformer	0.849	42	0.813

Table 1: English models results

Model	**Seed**	**F1 val**
Model 1 LSTM (Best)	1337	0.780
Transformer	42	0.805

Table 2: Spanish models results

4 Discussion

Considering the best outcomes of each model with the related seed, as expected, the worst results have been achieved by the "*Baseline LSTM*" model. Further improvements have been exploited by the "*Model 1 LSTM*", showing better generalization capabilities, even if the best performing model remains the *Transformer* for both languages.

Looking at the errors, what can be inferred from these experiments is that the main causes of any lack of performance are related to *data pre-processing and their distribution*. The more complex the language structures are, the worse the performance of the models in analyzing them, as in the case of Spanish. In addition, class imbalance towards non-sexist tweets for English data influences how the models learn, even if a deeper analysis of the tweets showed that there are some recurrent errors, due to the presence of gender-related words (in particular, the word "woman"). This behavior induces the models to show a sort of *cultural bias*, as shown in the following false-positive examples:

> *400237: "Yup I hate when men rape and kill women."*
>
> *400409: "Today women eye makeup. And those nails too."*

For any future development, it is recommended to work first on the data, trying to enhance data cleaning or to apply some methods to re-balance the distribution. Regarding the model, more complex architectures can be implemented to optimize the results. Any further deduction on the topic can be found in the error analysis part of the notebook.

5 Conclusion

Our work leads to the conclusion that implementing models, like *LSTM* and *Transformers* could be a reasonable choice to improve in tasks like *sexism detection*. The results achieved were not bad, but considering a real use inside a more complex system, several considerations have to be done. A cost analysis would be crucial to understand which type of error to reduce (*False Positives* or *False Negatives*), depending on the application. In addition, the accuracy is still too low for a fully automated process, meaning that a human supervision is still required.

As final considerations, the Transformer represents a big improvement with respect to LSTMs, not surprisingly, but still shows some room for refinements. Indeed, one point to still work on is to limit the dependency of the model from single words by applying some filtering or exploring preprocessing strategies that focus on the semantic of the text.

References

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2017. Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, 5:135–146.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural Computation*, 9(8):1735–1780.

Albert Zeyer, Patrick Doetsch, Paul Voigtlaender, Ralf Schlüter, and Hermann Ney. 2017. A comprehensive study of deep bidirectional lstm rnns for acoustic modeling in speech recognition. In *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 2462–2466.