TheNorth @ HaSpeeDe 2: BERT-based Language Model Fine-tuning for Italian Hate Speech Detection

Eric Lavergne, Rajkumar Saini, György Kovács and Killian Murphy
Luleå Tekniska Universitet
eric.lavergne@gmx.fr
rajkumar.saini@ltu.se
gyorgy.kovacs@ltu.se
killian.murphy@telecom-sudparis.eu

Abstract

English. This report was written to describe the systems that were submitted by the team "TheNorth" for the HaSpeeDe 2 shared task organised within EVALITA 2020. To address the main task which is hate speech detection, we fine-tuned BERT-based models. We evaluated both multilingual and Italian language models trained with the data provided and additional data. We also studied the contributions of multitask learning considering both hate speech detection and stereotype detection tasks.

1 Introduction

Organised as part of the 7th EVALITA evaluation campaign (Basile et al., 2020), the HaSpeeDe 2 shared task focuses on the detection of online hate speech (Sanguinetti et al., 2020) in Italian-Hate speech occurs frequently on social media. It is defined as "any communication that disparages a person or a group on the basis of some characteristics such as race, colour, ethnicity, gender, sexual orientation, nationality, religion, or other characteristics" (Nockleby, 2000). Regulating all user messages is very time-consuming for a human, and this is one of the reasons why automatic methods are important.

Beside the main task of binary hate speech classification - aimed at deciding whether a message contains hate speech or not - the HaSpeeDe 2 shared task has two more sub-tasks. One being stereotype detection, and the other the identification of nominal utterances. All tasks being evaluated both on in-domain (tweets) data, and out-of-domain (newspaper headlines) data. Here, we tackle both the main task, and the first sub-task of Stereotype Detection that is potentially useful for the main task. For this sub-task the organisers use the following definition of Stereotype: "a standardized mental picture that is held in common by members of a group and that represents an oversimplified opinion, prejudiced attitude, or uncritical judgment" (Merriam-Webster, 2020).

Here, we have two binary classification tasks. A simple way to perform text classification is based on bag-of-words representation counting the number of occurrences of each word within text. It is often combined with term frequency-inverse document frequency (Sparck Jones, 1988) (TF-IDF) representation. TF-IDF allows the frequencies to be normalized according to how often the words appear in all documents. With the rise of neural networks, word vectors have provided useful features for text classification tasks. Recurrent Neural Networks as the Bidirectional Long-Short Term Memory (BiLSTM) network (Schuster and Paliwal, 1997) have then be used to encode the long-term dependencies between the words. These systems were the most successful in the previous HaSpeeDe campaign (Bosco et al., 2018).

In (Aluru et al., 2020), the authors showed that when dealing with very low monolingual resources, multilingual approaches can be interesting for hate speech. In (Polignano et al., 2019b), the AlBERTo monolingual Italian BERT-based language model was trained that outperformed the state-of-the-art on the HaSpeeDe 2018 evaluation task (Polignano et al., 2019a).

We have chosen to deepen the approach of fine-tuning a BERT based language model, comparing multilingual and monolingual settings. We also assessed the contribution of additional hate speech data from different online sources. We finally submitted the results of the same model fine-tuned with and without multitask learning between hate speech and stereotype detection tasks.

2 System Description

2.1 Fine-tuning process

The chosen classification approach is to fine-tune a BERT-based language model. This kind of approach is the state-of-the-art for many text classifications tasks today (Sun et al., 2019; Seganti et al., 2019). BERT is a language model which aims to learn the distribution of language (Devlin et al., 2018). It is trained with the prediction of masked tokens in a text. The next sentence prediction task that was used simultaneously for training has been removed for some later BERT-based models such as RoBERTa (Liu et al., 2019). BERT is a Transformer. In a Transformer, the recurrence of Recurrent Neural Networks is replaced by the mechanism of attention (Vaswani et al., 2017).

It has been shown that it is possible to fine-tune these models for many downstream natural language processing tasks, including the one we are interested in, which is text classification. This can be achieved by removing the language modelling head and replacing it by a head appropriate for the target task. The designers of BERT prepared this by adding a token at the beginning of each text sequence, named CLS for classification. The purpose of this token is to contain the information useful for the classification task at the end of the forwarding process. Then a classifier head can just take this CLS token as input to classify the whole text sequence. In our case we decided to add a simple linear layer with a softmax on top of it, for simplicity and because it is efficient enough since the other layers are fine-tuned.

2.2 Layer-wise learning rate

An important consideration of fine-tuning described in (Sun et al., 2019) is the choice of the learning rate. Besides being as usual the most important hyper-parameter in the gradient descent learning algorithm, it could also be responsible here for some catastrophic forgetting if it were too high. Catastrophic forgetting refers to the fact of erasing the information of the weights of the pre-trained model and can happen when the gradient updates are too high.

Moreover, the learning rate can be gradually decreased in the first layers of the models. It aims at limiting the update in these first layers that have been showed to contain the most primal information about the language. One can think of the classical example in computer vision neural networks where the basics shapes features are extracted by the first layers and the task-specific combinations are processed in the last ones. Thus we applied layer-wise learning rate with the following geometric equation: the learning rate in a layer is the one of the following multiplied by a decay factor γ between 0 and 1.

$$LR_{k-1} = \gamma \times LR_k$$

where LR_k is the learning rate of the k-th layer.

Then the case when γ is one is the case of classic fine-tuning with the same learning rate everywhere, and the case when γ is zero is the case of feature extraction with the whole language model weights that are frozen and only the parameters of the classification head are trainable. This hyper-parameter γ was learned with the others during the hyper-parameters tuning process.

2.3 Monolingual and multilingual language models

We compared the use of several language models. Many models similar to BERT have been trained since 2018, and a lot are available for use. Although the models are often first and foremost trained for English, multilingual models have been trained on data of several languages in order to counteract the lack of data for some languages. It is the case of mBERT and XLM-Roberta (Conneau et al., 2020). Also machine learning researchers trained monolingual models for their own language, as CamemBERT for French and AlBERTo or UmBERTo for Italian. Multilingual models have the advantage that they are trainable on data in different languages; it is very useful for low-resources tasks. However, they are expected to perform in dozens of languages while monolingual models focus on just one, with the same number of parameters. For this reason, monolingual models often perform better when sufficient data is available, as we show here.

We evaluated two multilingual models, mBERT and XLM-RoBERTa, and three Italian monolingual models, AlBERTo, UmBERTo, and PoliBERT. AlBERTo was pretrained on TWITA, that is a collection of Italian tweets (Polignano et al., 2019b). UmBERTo was pretrained on Commoncrawl ITA exploiting OSCAR Italian large corpus (Parisi et al., 2020). Finally, PoliBERT was fine-tuned for sentiment analysis on Italian tweets by its creators (Barone, 2020).

We tried to use more data, with different settings. For the multilingual models, we could use all type of hate speech data. For the monolingual models, we used the little data available for Italian but we tried also to use translated multilingual data. These additions were not conclusive, so we stuck to the HaSpeeDe 2 data for the submissions.

2.4 Random search hyper-parameters tuning

The tuning of the hyper-parameters is relevant in order to get good results, and that is especially the case for the learning rate and the layer-wise decay factor γ. We tuned hyper-parameters with random search which has been shown to be often more efficient than grid-search (Bergstra and Bengio, 2012). The hyper-parameters to be tuned are the batch size, the learning rate, the layer-wise multiplier and the length of the model (maximum number of tokens). We did ten trials for each language model. The number of epochs is selected with early stopping on the validation macro F1-score with a split of 80/20. Table 1 shows the best hyper-parameters obtained that have been used for the systems submitted.

Hyper-parameter	Value
Learning rate	2.10-4
Layer-wise γ	0.35
Batch Size	32
Max Length	100
Language Model	UmBERTo

Table 1: Hyper-parameters used for our HaSpeeDe 2 submission after the tuning process

It is very important that the learning rate and the layer-wise multiplier γ are tuned simultaneously because the choice of the multiplier strongly modifies the amplitude of the gradient.

2.5 Multitask Learning

We evaluated the usage of multitask learning between the two classification tasks of the competition that are hate speech detection and stereotype detection. Multitask learning consists of learning to perform several tasks. It can be done by learning the tasks simultaneously with common first layers but task-specific heads (Ruder, 2017). In our case each task has its own output linear layer. When the tasks should be based on similar representations, it is supposed to do a good regularization with useful shared representations. It is then a kind of transfer learning. The error analysis conducted on HaSpeeDe 2018 evaluation suggests a significant correlation between the usage of stereotype and hate speech (Francesconi et al., 2019). Moreover, they showed that the false positive rate of hate speech tweets is slightly bigger for tweets with stereotype.

A question that arises when doing multitasking is the way to combine the loss of the tasks in one. The simple solution is to sum them uniformly. It might not be the best solution when there is imbalance between the tasks, for instance when the scale of the outputs of one is much higher than the others. A solution brought by (Kendall et al., 2017) is to use trainable weights based on uncertainty. (Liebel and Körner, 2018) upgrades the regularisation term of this solution and (Gong et al., 2019) shows in a benchmark that this last solution is often the best. We evaluated this solution and we compared with the single-task setting.

2.6 Cross-validation ensembling and submitted models

Two submissions are allowed during the HaSpeeDe 2 test phase. We chose to submit a fine-tuned UmBERTo trained separately for each of the two tasks and a fined-tuned UmBERTo with multitasking on both Stereotype and Hate Speech detection. The hyper-parameters used to train these models were presented in Table 1.

Since we compared the different language models with 5-fold cross-validation, we then ensembled the 5 models obtained for each fold in order to get the final model. The ensembling was done by considering the mean of the probabilities returned by each model.

3 Data Description

The organisers provided a train dataset of 6,839 tweets, annotated with Hate Speech and Stereotype labels (as described in Table 2).

Dataset	HS	Ster
Development Data (Tweets)	0.404	0.445
Test Data (Tweets)	0.492	0.450
Test Data (News)	0.362	0.350

Table 2: Distribution of Hate Speech and Stereotype labels in HaSpeeDe 2 data.

The test data of HaSpeeDe 2 consists of two subsets: an in-domain set (1,263 tweets) and an out-of-domain set (500 newspaper headlines).

The hate speech labels are slightly unbalanced towards non-hate speech. Thus we tried to use adapted losses to prevent tendency towards non-hate speech predictions. We used class-weighted loss, which assigns a higher weight to the observations from the minority class in the computing of the loss. We also tried to use a smoothed F1-score – a differentiable loss in phase with the F1. Neither approach improved the results in a significant way.

The pre-processing was simple. We removed emoticons and hashtags and we replaced urls and user names with associated tags as done in the evaluation data. Each tweet was padded with a size of 100. Then we used the pre-processing and tokenization pipeline specific to each language model as provided by the authors of the models.

4 Results

4.1 Macro F1-score

The metric used for the evaluation is the macro F1-score. The F1-score of a class is computed by calculating the harmonic mean between the precision and recall for this class. The macro F1-score is the mean between the F1-scores for each class. It is less sensitive to the imbalance between the classes.

4.2 Baselines

We used several baselines to evaluate our results during the development process. The first ones are those obtained by dummy classifiers, one that always predicts the most frequent class and the other one that makes a random stratified prediction according to the distribution of the classes in the training data. We also computed the results of more developed systems, that are a TF-IDF bag of words and a BiLSTM with trainable word vectors inputs.

The HaSpeeDe 2 organisers provided two baseline systems after the results were submitted. The first is a most frequent class predictor and the second is a linear SVM with unigrams, char-grams and TF-IDF representation.

4.3 Validation Results

We tuned the hyper-parameters for each evaluated language model as described in Section 2.4. For each language model, we then computed 5-fold cross-validation results on HaSpeeDe 2 training data. The averages of the 5 macro F1-scores are shown in Table 3.

System	**HS**	**Ster**
Baselines		
Most Frequent Class	0.374	0.353
TF-IDF Bag-of-words	0.703	0.677
Word vectors + BiLSTM	0.721	0.654
Multilingual language models		
mBERT	0.757	0.716
XLM-RoBERTa	0.761	0.677
Italian language models		
AlBERTo	0.773	0.716
PoliBERT	0.795	**0.733**
UmBERTo	**0.799**	**0.733**

Table 3: Macro F1-scores averaged over 5-fold cross-validation on HaSpeeDe 2 training data.

4.4 Test Results

The scores of our two systems evaluated on the HaSpeeDe 2 test data are summarized in Table 4. These systems are 5 UmBERTo models trained on each of the 5 training folds and ensembled. The second system is the same as the first with the use of multitask learning.

System	**Tweets**	**News**
Hate Speech Detection		
Most Frequent Class	0.337	0.389
Classic Features + SVM	0.721	0.621
UmBERTo	0.790	0.671
UmBERTo + Multitasking	**0.809**	0.660
Best HaSpeeDe 2	**0.809**	**0.774**
Stereotype Detection		
Most Frequent Class	0.355	0.394
Classic Features + SVM	0.715	0.669
UmBERTo	**0.772**	0.685
UmBERTo + Multitasking	0.768	0.647
Best HaSpeeDe 2	**0.772**	**0.720**

Table 4: Macro F1-scores on HaSpeeDe 2 test datasets.

5 Discussion

5.1 Multilingual and monolingual models

According to Table 3, multilingual models performed worse than monolingual models based on HaSpeeDe 2 data alone, although they achieved respectable results.

Moreover, even when we used additional data from other languages to train the multilingual models, they still did not manage to outperform the monolingual models, as we were hoping they would.

Within the Italian models, UmBERTo and PoliBERT performed better than AlBERTo on these tasks. While the good performance of PoliBERT can be linked to its pre-training for a tweet classification task (sentiment analysis) potentially useful for hate speech detection, it is more difficult to explain the competitiveness of UmBERTo, which was trained on data not coming from Twitter and less numerous than for AlBERTo. One explanation could be the better quality of this data, or a better optimisation by its creators.

5.2 Out-of-domain data and in-domain data

Our results on the HaSpeeDe 2 test dataset are summarized in the Table 4. The results obtained on in-domain data correspond to what we expected from our cross-validation results. Our systems achieved the best macro F1-scores on the in-domain test set (Tweets) for both hate speech and stereotype detection. However, the results on out-of-domain data (News) are far from being as good. This can be explained by the different distribution of this data compared to the training data.

Table 5 shows the confusion matrix for our first system evaluated on out-of-domain data. The error is mostly due to the high number of false negatives. The classifier predicts too many sequences as non-hate speech. This suggests that this classifier trained with hate speech on Twitter is struggling to detect hate speech in newspaper headlines. It can be assumed that hate speech in newspapers is more subtle, with less coarseness and aggressiveness that make it easier to detect on Twitter.

	Predicted False	Predicted True
False	312	7
True	117	64

Table 5: Hate Speech Confusion matrix for UmBERTo evaluated on news test data.

5.3 Multitasking Benefits

We have chosen to submit a system with multitask learning on both Stereotype and Hate Speech detection and an other one without, in order to study the benefits of it. Indeed, the system with multitasking learning performed much better on the in-domain data for the hate speech detection task. It is not the case however for the out-of-domain data, neither for the stereotype detection task.

Table 6 describes in more detail the differences between the predictions of the two systems for data containing stereotypes and data not containing stereotypes. We observed that the improvement linked to multitask learning consists mainly in a reduction in the number of false positives in favour of the number of true negatives in data not labeled as Stereotype. Assuming that hate speech makes significant use of stereotype, one could suppose that the multitask model has learned to discard some data that do not have the characteristics of stereotypes and are therefore unlikely to contain hate speech.

Data labeled as Stereotype		
	Predicted False	Predicted True
False	+3	-3
True	+7	-7
Data not labeled as Stereotype		
	Predicted False	Predicted True
False	+28	-28
True	+1	-1

Table 6: Hate Speech Confusion matrix of the multitask system minus the one of the single-task system, for Stereotype and Non Stereotype tweets test data.

6 Conclusion

In this work, we compared the fine-tuning of multilingual and monolingual BERT-based language models for hate speech detection. We also investigated the addition of multitask learning with the Stereotype detection task linked to hate speech. We obtained the best macro F1-scores of HaSpeeDe 2 on the in-domain test data. However, the results were worse for out-of-domain test data, and further research could be conducted to better understand the reasons for this and address it.

References

Sai Saketh Aluru, Binny Mathew, Punyajoy Saha, and Animesh Mukherjee. 2020. Deep Learning Models for Multilingual Hate Speech Detection.

Gianfranco Barone. 2020. Politic BERT based Sentiment Analysis. `https://huggingface.co/`

unideeplearning/polibert_sa. accessed on Sept 18, 2020.

Valerio Basile, Danilo Croce, Maria Di Maro, and Lucia C. Passaro. 2020. EVALITA 2020: Overview of the 7th Evaluation Campaign of Natural Language Processing and Speech Tools for Italian. In Valerio Basile, Danilo Croce, Maria Di Maro, and Lucia C. Passaro, editors, *Proceedings of Seventh Evaluation Campaign of Natural Language Processing and Speech Tools for Italian. Final Workshop (EVALITA 2020)*, Online. CEUR.org.

James Bergstra and Y. Bengio. 2012. Random Search for Hyper-Parameter Optimization. *The Journal of Machine Learning Research*, 13:281–305, 03.

Cristina Bosco, Felice Dell'Orletta, Fabio Poletto, M. Sanguinetti, and M. Tesconi. 2018. Overview of the EVALITA 2018 Hate Speech Detection Task. In *EVALITA@CLiC-it*.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised Cross-lingual Representation Learning at Scale.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *CoRR*, abs/1810.04805.

Chiara Francesconi, Cristina Bosco, Fabio Poletto, and M. Sanguinetti. 2019. Error Analysis in a Hate Speech Detection Task: The Case of HaSpeeDe-TW at EVALITA 2018. In *CLiC-it*.

Ting Gong, Tyler Lee, Cory Stephenson, Venkata Renduchintala, Suchismita Padhy, Anthony Ndirango, Gokce Keskin, and Oguz Elibol. 2019. A comparison of loss weighting strategies for multi-task learning in deep neural networks. *IEEE Access*, PP:1–1, 09.

Alex Kendall, Yarin Gal, and Roberto Cipolla. 2017. Multi-Task Learning Using Uncertainty to weigh Losses for Scene Geometry and Semantics. *CoRR*, abs/1705.07115.

Lukas Liebel and Marco Körner. 2018. Auxiliary Tasks in Multi-task Learning. *CoRR*, abs/1805.06334.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. RoBERTa: A Robustly Optimized BERT Pretraining Approach.

Merriam-Webster. 2020. stereotype, noun. https://www.merriam-webster.com/dictionary/stereotype. Accessed on 2020-11-05.

John T. Nockleby. 2000. *Hate Speech*. Macmillan, New York.

Loreto Parisi, Simone Francia, and Paolo Magnani. 2020. UmBERTo: an Italian Language Model trained with whole word Masking. https://github.com/musixmatchresearch/umberto. accessed on Sept 18, 2020.

Marco Polignano, Pierpaolo Basile, Marco De Gemmis, and Giovanni Semeraro. 2019a. Hate Speech Detection through AlBERTo Italian Language Understanding Model. In *NL4AI@AI*IA*.

Marco Polignano, Pierpaolo Basile, Marco De Gemmis, Giovanni Semeraro, and Valerio Basile. 2019b. AlBERTo: Italian BERT Language Understanding Model for NLP Challenging Tasks Based on Tweets. In *Proceedings of the Sixth Italian Conference on Computational Linguistics (CLiC-it 2019)*, volume 2481. CEUR.

Sebastian Ruder. 2017. An Overview of Multi-Task Learning in Deep Neural Networks. *CoRR*, abs/1706.05098.

Manuela Sanguinetti, Gloria Comandini, Elisa Di Nuovo, Simona Frenda, Marco Stranisci, Cristina Bosco, Tommaso Caselli, Viviana Patti, and Irene Russo. 2020. Overview of the EVALITA 2020 Second Hate Speech Detection Task (HaSpeeDe 2). In Valerio Basile, Danilo Croce, Maria Di Maro, and Lucia C. Passaro, editors, *Proceedings of the 7th evaluation campaign of Natural Language Processing and Speech tools for Italian (EVALITA 2020)*, Online. CEUR.org.

M. Schuster and K.K. Paliwal. 1997. Bidirectional recurrent neural networks. *Trans. Sig. Proc.*, 45(11):2673–2681, November.

Alessandro Seganti, Helena Sobol, Iryna Orlova, Hannam Kim, Jakub Staniszewski, Tymoteusz Krumholc, and Krystian Koziel. 2019. NLPR@SRPOL at SemEval-2019 Task 6 and Task 5: Linguistically enhanced deep learning offensive sentence classifier. In *SemEval@NAACL-HLT*.

Karen Sparck Jones, 1988. *A Statistical Interpretation of Term Specificity and Its Application in Retrieval*, page 132–142. Taylor Graham Publishing, GBR.

Chi Sun, Xipeng Qiu, Yige Xu, and Xuanjing Huang. 2019. How to Fine-Tune BERT for Text Classification? In Maosong Sun, Xuanjing Huang, Heng Ji, Zhiyuan Liu, and Yang Liu, editors, *Chinese Computational Linguistics*, pages 194–206, Cham. Springer International Publishing.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is All you Need. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 5998–6008. Curran Associates, Inc.