Hate Speech Detection in Italian twitters

Pietro Epis, Michele Milesi and **Anna Valanzano**
Master's Degree in Artificial Intelligence, University of Bologna
{ pietro.epis, michele.milesi, anna.valanzano}@studio.unibo.it

Abstract

Most research efforts for the Hate Speech Detection task are directed toward producing higher-quality models for the English language. There are fewer and less powerful tools for detecting Hate speech in other languages, such as Italian. In this work, we will explore monolingual and multilingual BERT models in the Italian language to solve the first two tasks of EVALITA 2020, Automatic Hate Speech and Stereotype Detection.

1 Introduction

Hate speech is any kind of communication that attacks a person or a group based on their characteristics, such as gender, religion and race. Given its spread in social media, nowadays, Automated Hate Speech Detection is a fundamental tool and one of the main NLP topics. In this work, we consider tasks A and B of Evalita 2020, which concerns the automatic detection of hateful content in Italian Twitter messages and news. The Main Task A of the challenge is Hate Speech Detection, i.e., a binary classification task aimed at determining whether the message contains Hate Speech or not. Instead, Pilot Task B is Stereotype Detection, i.e., a binary classification task aimed at determining whether the message contains a Stereotype.

These tasks can be handled by simple Information retrieval approaches such as TF-IDF, which would classify a Tweet based on the frequency of "hate words". However, these types of approaches would dramatically fail in this Dataset, because, as will be further explained in Section 3, words are similarly distributed in the two classes. Another approach is given by Recurrent Neural Networks as Bidirectional Long-Short Term Memory networks, which encode the long-term dependencies between words. These systems reached very good results for the Hate Speech Detection Task in Evalita 2018.

As BERT is one of the most eminent architectures for different NLP tasks that achieved state-of-the-art performance in text classification, we have chosen to deepen the approach of fine-tuning BERT-based pre-trained models.

Since most research efforts for Hate Speech Detection are directed toward producing higher-quality models for the English language, there are fewer and less powerful tools for detecting Hate speech in other languages, such as Italian. In this work, we will explore different models for Hate Speech Detection in the Italian language, and we will compare monolingual and multilingual models. "Due to the variety of text corpus in terms of languages used for training, multilingual models find notable benefits over multiple applications, specifically for languages that are low in resources. However, the monolingual models, when used in the corresponding language, outperform the multilingual versions in tasks like text classification" (Velankar et al., 2022). This is because monolingual models focus on just one language, with the same number of parameters. Therefore, we expect monolingual models to perform better when sufficient data is available. We fine-tuned several BERT-based pre-trained models (and reported only the most significant ones). We will observe different behaviors of the models in the two tasks and in the two types of data, in-domain and out-domain data. However, the hypothesis about the victory of monolingual models over multilingual models will be confirmed for the Hate Speech Task.

2 System description

We tested several pre-trained models for the two tasks (hate speech detection and stereotype detection). As multi-lingual models, we considered the XLM Roberta (xlm-roberta-base, xlm-roberta-large) and multilingual BERT (bert-base-multilingual-cased). Multilingual language models have pushed the state-of-the-art on cross-lingual understanding tasks by jointly pretraining large

Transformer models on many languages. XLM-RoBERTa model is a multilingual version of Roberta pre-trained on 2.5TB of data containing 100 languages, while multilingual BERT is pre-trained on the top 104 languages. Thanks to the use of Transformer based masked language model, "XLM-RoBERTa outperforms mBERT on a variety of cross-lingual benchmarks"(Conneau et al., 2020). Multilingual BERT models capture universal semantic structures, but they do gloss over language-specific differences. Consequently, a number of language-specific BERT models have been developed to fill that need. These models almost always showed better performance on the language they were trained for than the universal model (Nozza et al., 2020). We experimented with two mono-lingual BERT models for Italian dbmdz/bert-base-italian-cased and Hate-speech-CNERG/dehatebert-mono-italian. These monolingual models should provide high performance if they are trained on enough amount of Italian data (Nozza et al., 2022). The Hate-speech-CNERG/dehatebert-mono-italian model is specifically developed to detect hate speech in the Italian language.

The data are tokenized with a general Autotokenizer from HuggingFace, and the sentences are padded or, if necessary, truncated to the maximum length accepted by the model. For the training procedure, we used the Hugginface Trainer class, which provides an API for feature-complete training in PyTorch.

Each row of training DataFrame contains a Tweet and the corresponding true labels, 1 if it contains hate/stereotypes, 0 otherwise. Instead, the test DataFrame contains tweets and news, mixing in-domain and out-domain data. The DataFrames have been converted into Datasets objects (from Hugginface's library) to ease the tokenization and training procedure.

3 Data

The dataset for the several tasks has been provided by the organizers of the challenge, which made it available in a GitHub repository. It is worth pointing out that all text data in the dataset is expressed in Italian language, and is related to minorities, such as Immigrants, Muslims and Roma communities, either with or without hatred meaning and either expressed or not in form of stereotype. The representation and labeling of these concepts are described in the following. The usage of the data was not straightforward, since it was downloadable as protected zip files, therefore requiring a password to be read. The access to data was granted after filling in a form for participants' information collection and license terms acceptance. Specifically, data is organized in three zip archives, namely *haspeede2_dev.zip*, *haspeede2_test.zip* and *haspeede2_reference.zip*. The first one contains the training set, whereas the latter two are related to the test set, and differ by the fact that data into *haspeede2_reference.zip* is labeled, whereas the target columns are missing in *haspeede2_test.zip*. The training set is wholly composed of tweets, whereas the test set is split into two files, one made up by tweets, hence in-domain with respect to the data on which the model will be trained, and one that collects text from newspaper headlines, therefore out-of-domain. Eventually, we obtained all the files to carry out every task, but we only took into account the ones related to tasks A and B of the challenge, and dropped the files for task C, which were out of our purpose. Every file of our interest in the dataset is in *tsv* format, i.e. tab-separated values. We maintained the original structure of the dataset: each instance is described by four features, respectively the *id* of the tweet, the *text* to be classified, and the two target fields *hate speech* and *stereotype*, implied to accomplish the two goals respectively of hate speech and stereotype classification.

As shown in Figure 1, the distribution of the classes Hate or Non-Hate for task A into the training, validation and test set is slightly unbalanced. Instead, the distribution of the classes Stereotype and Non-

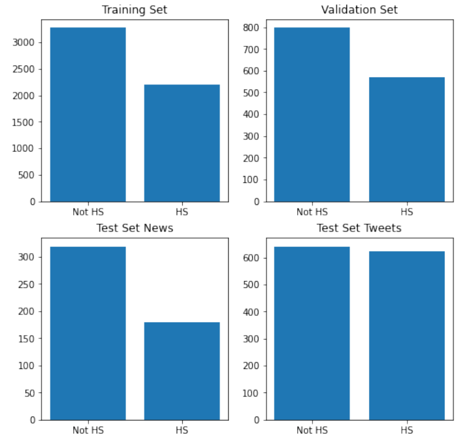

Figure 1: Distribution of classes (Hate and Non Hate) across training, validation, and test set

Stereotype for task B are less unbalanced.
After a statistical analysis of the most present words in the dataset (after removing stop-words), we can affirm that it is pretty hard to classify a text as hateful according only to the frequency of words, without considering the semantics of the sentence. Indeed, there are some words that are pretty similar in the two classes such as "migranti" but also counterproductive words such as "rom". Indeed, if lots of non-hateful texts refer to nomadic communities, these types of texts in the test set would probably be classified as non-hateful. We show some intuitive graphs about the distribution of words in Figure 2 and Figure 3. In order to take the data in a format that could foster the learning and remove the focus from irrelevant (even if frequent) parts, we carried out a consistent preprocessing phase, before passing them to the model. In particular, the first steps aimed to remove standard and not meaningful placeholders of the corpus, such as "URL" (that stands for a generic URL in the original tweet) and "@user" (representing the tag of the user, censored in the dataset). Furthermore, we tried to enrich the contribution provided by the hashtags, first of all by removing the "#" character. Then, exploiting the advantage that they are often expressed in *camelCase* format (thus easing the split into words), the hashtags are converted into a list of words that can help the model to classify the tweet.

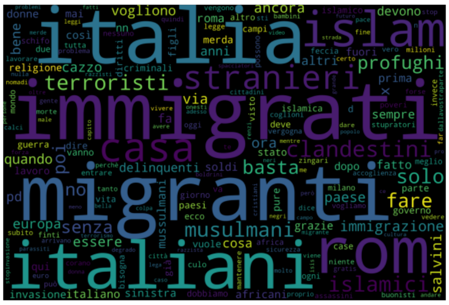

Figure 2: Distribution of words in hateful texts

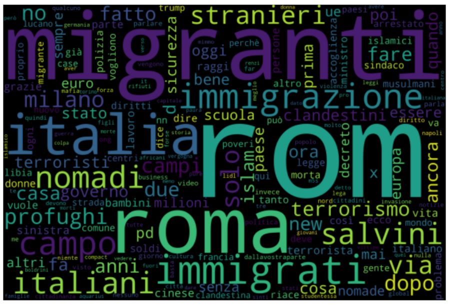

Figure 3: Distribution of words in non-hateful texts

All characters but usual letters, numbers and spaces were removed, all text turned into the lowercase format, and multiple spaces were reduced to a single one. In the attempt to take the text as standardized and general as possible, we experimented also with stemming and lemmatization, both in conjunction and separately. In spite of this intuition, unfortunately, we could not manage to get any improvement in terms of classification metrics from these procedures, hence we preferred discarding them and keeping the text as it is in its original form. In particular, we made use of *SnowBall* stemmer, provided by *nltk* package, and of *simplemma* package for lemmatization, both easily providing support for the Italian language.

Parameter	**Value**
Batch size	8
Learning rate	3e-5
Weight decay	0.01
Number of epochs	3
Optimizer	Adafactor

Table 1: Hyper-parameters setting for hate speech detection model

4 Experimental setup and results

The models have been trained with different hyper-parameters, in particular, we tuned the following ones: *(i)* number of epochs, *(ii)* learning rate, *(iii)* weight decay, *(iv)* batch size and *(v)* optimizer. Furthermore, we checked whether any preprocessing techniques significantly affected the performance, e.g., stemming and lemmatization. In order to make the experiments repeatable and have a fair comparison between models, we set the random seed to 42. In Table 2 are shown the results on the validation dataset of the hate speech models and the results on the validation dataset of the stereotype models. Both tables show only the results obtained with the best set of hyper-parameters. We report the best configuration of hyper-parameters of the best model for Hate Speech Detection (*Hate-speech-CNERG/dehatebert-mono-italian*) in Table 1. The values of the hyper-parameters of the best model for Stereotype Detection (*xlm-roberta-base*) are equal to the previous one, except for the optimal number of epochs which is 5. Both models perform worse with lemmatization and stemming.

Model	Task	Accuracy	macro-F1
xlm-roberta-base	Hate	0.78	0.75
	Stereotype	**0.74**	**0.73**
xlm-roberta-large	Hate	0.74	0.73
	Stereotype	0.54	0.35
bert-base-multilingual-cased	Hate	0.72	0.70
	Stereotype	0.71	0.69
Hate-speech-CNERG/dehatebert-mono-italian	Hate	**0.79**	**0.79**
	Stereotype	0.71	0.69
dbmdz/bert-base-italian-cased	Hate	0.74	0.71
	Stereotype	0.71	0.68

Table 2: Best accuracy and F1 metrics on the validation set of the hate speech detection and stereotype detection tasks

Dataset	Task	Accuracy	macro-F1	macro-f1 baseline
Tweet	Hate	0.75	0.75	0.72
	Stereotype	0.75	0.75	0.71
News	Hate	0.74	0.66	0.62
	Stereotype	0.74	0.67	0.67

Table 3: Results of the best models for tasks A and B on the test set and the relative macro-F1 metric of the baseline model of the challenge

Once the two best models are found, they were trained on the training set with the best hyperparameters, and then the predictions on the test set have been computed. Two metrics have been chosen to evaluate our models: accuracy and macro-F1. The former is the standard and the most used metric for the classification task, the latter is chosen because it is the benchmark metric used by the competition, so we are able to compare our models with the results of other challenge participants. In Table 3 are given the results of the hate speech detection and the results of the stereotype detection tasks on the test set, furthermore, the results obtained by the baseline model of the challenge are reported in order to make an easy comparison.
The experiments were executed on Google Colaboratory with a free GPU (usually it was an NVIDIA TESLA T4 with 16GB of RAM). Furthermore, the main third-party libraries on which the project is based are `pytorch` and `transformers` (provided by HuggingFace). However, other supporting libraries have been used such as `scikit-learn` (for evaluation) and `nltk` (for dataset analysis).

5 Discussion

Given the results reported in Section 4, we can assert that the best model for the hate speech detection task is *dehatebert-mono-italian*, whereas for stereotype detection it is *xlm-roberta-base*. Even if the best performance on Hate Speech Task was achieved with a monolingual model, the difference between the multilingual and monolingual models seems to be not very significant. All the models reached worse results on task B than on task A. This is because some models we exploited have been initially devised specifically for the task of hate detection, and the idea of taking advantage of them to fulfill the task of Stereotype Detection is just an adaption. Moreover, Stereotype Detection is intrinsically more difficult, since it also entails a subjective aspect, as we will discuss in Section 5.1. We can compare the results of the model with the ones of the participants of the challenge (Sanguinetti et al., 2020). Almost all the models outperform the results of the original baseline proposed by challenge *Baseline_SVC* (whose performance is shown in the last column of Table 3) and respectable results with respect to the other participants.

5.1 Error Analysis

As out-domain data (News) are unseen during the training, the model achieved the best macro F1-scores on the in-domain test set (Tweets) for both hate speech and stereotype detection. We report the confusion matrices for out-domain data for task A in Table 4.

	False	True
False	303	16
True	114	66

Table 4: Task A (hate detection) for news

The rows of the matrices refer to the true labels, while the columns to the predicted ones. In task A, among the total number of errors (130) the most frequent error is represented by false negatives (114). This means that the model tends to make mistakes on hateful text (that it classifies as non-hateful), while it tends to identify well non-hateful content. These errors can be partly due to unbalanced of the data. The major presence of non-hateful samples with respect to hateful ones (as described in Section 3) in all the splits, may increase the model's probability of predicting "non-hate", and therefore the number of false negatives. Moreover, these errors may be partly justified also by the presence of several sentences in the dataset which would be difficult for even humans to classify, such as

> Anziana rapinata sull'autobus, i due nomadi in fuga si rifugiano al campo di via Candoni.

Nevertheless, there are also sentences that should be unquestionably classified as hate, but that are classified as non-hate, such as

> Giorgia Meloni, una sonora lezione a Madame Boldrini: "Vergogna, ai profughi case e diritti. Italiani umiliati..."

As reported in Table 5, we can observe the same (even worse) phenomenon becomes for task B, where only a little less than half of the texts containing stereotypes are properly labeled (it classifies as stereotypes only 72 samples over 175).

	False	True
False	298	26
True	103	72

Table 5: Task B (stereotype detection) for news

As described in Section 3 for task B the classes are more balanced than in task A. Also in this task, there are sentences very hard to classify

> Il piano per fermare l'ondata di clandestini: 10mila rimpatri al mese

and others that should be unquestionably classified as Stereotypes, such as

> «I rom non sono uguali a noi». E lo studio di Piazza Pulita applaude il giovane ospite

In general, we would have this high number of false negatives because stereotypical content toward a given target might be expressed using in very subtle forms, that the system struggles to capture. Moreover, for both tasks, the targets of hateful and stereotypes (Muslims, immigrants, Roma) may be not balanced. In particular, the systems seem to struggle to recognize hateful/stereotypical content against Roma: this may be caused by an imbalance in the training and test sets, that can be intuitively understood by Figure 2 and Figure 3 in Section 3.

6 Conclusion

We tested several pre-trained monolingual and multilingual models for the two tasks of Evalita 2020. According to the literature, monolingual models should outperform the multilingual ones, but we did not observe a significant difference between them. The task of Stereotype Detection turned out to be more challenging than Hate Speech Detection, because stereotypical content toward a given target might be expressed using in very subtle forms. As expected, we obtained the best macro F1-scores on the in-domain test data and worse results for out-of-domain test data. Further research could be conducted to combine our models with lexical resources to support the recognition of hate and stereotypes and to understand how much the two tasks are correlated. Indeed, the presence of stereotype is more frequent in hateful tweets and vice versa.

7 Links to external resources

The dataset can be obtained at this link with the password `zNw3tCszKWcpDahq`.

References

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. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8440–8451, Online. Association for Computational Linguistics.

Debora Nozza, Federico Bianchi, and Giuseppe Attanasio. 2022. HATE-ITA: Hate speech detection in Italian social media text. In *Proceedings of the Sixth Workshop on Online Abuse and Harms (WOAH)*, pages 252–260, Seattle, Washington (Hybrid). Association for Computational Linguistics.

Debora Nozza, Federico Bianchi, and Dirk Hovy. 2020. What the [mask]? making sense of language-specific bert models.

Manuela Sanguinetti, Gloria Comandini, Elisa Nuovo, Simona Frenda, Marco Stranisci, Cristina Bosco, Tommaso Caselli, Viviana Patti, and Irene Russo. 2020. Haspeede 2 @ evalita2020: Overview of the evalita 2020 hate speech detection task.

Abhishek Velankar, Hrushikesh Patil, and Raviraj Joshi. 2022. *Mono vs Multilingual BERT for Hate Speech Detection and Text Classification: A Case Study in Marathi*, pages 121–128.