Montanti @ HaSpeeDe2 EVALITA 2020: Hate Speech Detection in Online Contents

Elia Bisconti
University of Pisa
eliabisconti@gmail.com

Matteo Montagnani
University of Pisa
matteo.montagnani8@gmail.com

Abstract

English. This report describes an approach to face a task regarding the identification of hate content and stereotypes within tweets. Two models will be shown, both presented to the *HaSpeeDe* competition proposed by EVALITA 2020. They are based on a Logistic Regression model that takes different types of embedding as input. The best system shows interesting results.

Italiano. *In questa relazione viene mostrato un approccio volto ad affrontare un task riguardante l'identificazione di contenuti d'odio e stereotipi all'interno di tweets. Sono stati realizzati due modelli, presentati alla competizione HaSpeeDe proposta da EVALITA 2020. Entrambi si basano su un modello di Logistic Regression che prende in input diversi tipi di embedding. Il miglior sistema evidenzia dei risultanti interessanti.*

1 Introduction

The use of *bad words* and *bad language* has always been a subject of debate. The spread of social media platforms, such as Twitter and Facebook, has fostered the growth of hate speech online. These sites have been urged to treat and remove offensive content, but the phenomenon is so pervasive that the manual way of filtering out hateful tweets is not enough. For that reason, the development of automatic recognition systems is increasingly important. To date, the use of Natural Language Processing (Bird et al., 2009) is fundamental in this field. Most of the systems proposed so far are based on manual feature extraction (Joulin et al., 2016), even if in recent years some approaches based on Deep Learning techniques (Badjatiya et al., 2017) have been proposed. EVALITA organized the second edition of an NLP competition for *Hate Speech Detection* (Basile et al., 2020), intending to analyze various techniques for automatic recognition systems.The main goal was to classify a sentence as *hate speech* or even as *stereotyping*. The organizers provided us an in-domain dataset for training and testing and another out-domain. In this report, we will show a classical supervised approach with the aim of obtaining good results regarding the out-of-domain test.

2 Tasks Description

The task proposed in the competition (Sanguinetti et al., 2020) consists of three parts, but only the first two ones will be examined in this article; they correspond to the following sub-tasks:

- **Subtask A** - *Hate Speech Detection*: it consists of a binary classification task aimed at determining the presence or the absence of hateful content in the text towards a given target.
- **Subtask B** - *Stereotype Detection*: it consists of a binary classification task aimed at determining the presence or the absence of a stereotype, therefore an oversimplified opinion, prejudiced attitude, or uncritical judgment, toward a given target. This aims to boost the investigation of its occurrences, especially in a hateful context.

The performances of the participating systems are evaluated on a corpus of Italian tweets as in the previous edition and also on a set of mixed text genres, such as newspapers, comments and headlines.

3 Dataset

The dataset used is the one provided by the competition organizers. In particular, the entire dataset is split into one Training Set composed of tweets and two test sets: an in-domain (based on tweets) and a smaller out-of-domain (based on newspaper phrases) test set. Overall, the Training Set includes 6,839 Italian tweets distributed as in Tables 1 and 2.

	Hate Speech	Not Hate Speech
TR Set	2766	4073

Table 1: Distribution of Hate Speech on the Training-set

	Stereotype	Not Stereotype
TR Set	3042	3797

Table 2: Distribution of Stereotype on the Training-set

As we can see, the data are not well distributed. Regarding the Hate Speech Training Set, we have that sixty percent of the data are classified as *hate speech*. The Stereotype Training Set is also a little unbalanced, with fifty-five percent of the data classified as *non-stereotype*.

4 Proposed Approach

In this section, the proposed approaches will be described, focusing on what has been developed for the preprocessing of data, the used embeddings and models. Some decisions regarding the choice of models and the extraction of features were made based on the results obtained in other related works.

4.1 Preprocessing

A Tweet is a text message with a maximum length of 280 characters. It may contain elements such as hashtags, mentions, links and emoticons.
An example of a tweet extracted from the dataset is shown below:

"@user La società multirazziale... #migranti #profughi #rom URL"

As we can see in the example, the dataset provided has already been preprocessed, censuring names and URLs, probably for privacy.

The preprocessing phase that we faced implements a series of functions aimed at modifying a tweet to eliminate useless elements and to standardize it. Punctuation, emoji and any symbols are also eliminated. The tweet is also transformed into a *lower case* representation as shown:

"la società multirazziale migranti profughi rom"

Regarding this phase, the transformation of the single words from an inflected form to root or canonical form was also carried out, respectively, through *stemming* and *lemmatization*. We tried to consider these characteristics during the feature selection phase. However, these attempts will not be mentioned further, as they did not produce relevant results.

4.2 Feature vectors

The preprocessed tweets were used to generate the feature useful for classification purposes. Both tasks were addressed with the same types of representation and the same models.

- *TF-IDF Vector:* (Qaiser and Ali, 2018) the idea for the use of this function was to give more importance to the less frequent, but relevant, words. The vectors were generated using the *TfidfVectorizer* class present in the *scikit-learn* library.

- *DistilBert:* (Wolf, 2019) this is a pre-trained model. A single output vector with a size of 768 is considered, corresponding to the result of the first position of what the model received in input, that is the special token [CLS], used for the sentence-level classification.

- *GloVe:* (Pennington et al., 2014) we used a pre-trained model that returns a vector representation of words. The database, extracted from Twitter, includes more than 2 billion phrases, which generated about 27 billion tokens.

These three types of features were used both individually and in combination with each other by concatenation. To decrease the size of these vectors and to speed up the training phase, a *features Selection* phase is also performed using a *Random Forest Classifier*.

5 Systems and Results

For both tasks, we tried the use of an SVM Classifier with *kernel RBF*, a Logistic Regression and a Random Forest. As already mentioned, each of these models has taken various concatenations of the previous feature vectors as input.
We tested each model using 3-fold cross-validation and performed a grid-search to iterate over the models and all the parameters.
As a result of this search, the best final model was undoubtedly the Logistic Regression that has performed well also in previous papers (Davidson et al., 2017). As for the input features, we expected that the concatenation of features extracted with the different techniques described above would lead to the best results. Unexpectedly, instead, the best results were obtained in the validation phase with the use of TFIDF only. The second best one was obtained with the TFIDF concatenated with the DistilBert vectors. These two systems represent the two runs submitted to the competition. Overall, the difference in the results between the first and the second model is considerable; therefore, we will show in the following table the F1 values obtained with the best run, for tasks A and B, respectively.

TaskA	Tweets TS		News TS	
F-score	NoHS	HS	NoHS	HS
	0.750	0.735	0.835	0.615
M-F1	0.7432		0.7256	

Table 3: Task A - Results for the Logistic Regression with Tfidf

TaskB	Tweets TS		News TS	
F-score	NoST	ST	NoST	ST
	0.724	0.690	0.824	0.608
M-F1	0.7076		0.7166	

Table 4: Task A - Results for the Logistic Regression with Tfidf

Beyond the macro-F1 values obtained, it is interesting to note the behavior of the model with regard to the out-domain Test Set in both tasks. In particular, the F-scores show worse values in the classification of sentences that actually contain hate speech or stereotyping. This is actually due to low Recall values (about 0.51 for both tasks) which is probably due to the fact that the model is trained on a different type of data.

6 Discussion

Observing the results on the in-domain Test Set, our best models obtained a ranking of 15/27 and 6/12 respectively for tasks A and B. Regarding the out-domain Test Set, they obtained the third-best score in both tasks. The result obtained with the first Test Set confirms that the proposed approach turned out to be too simplistic. However, it's interesting to notice how such a simple system achieved a good placement in the out-of-domain test-set. An explanation of that could be the way the Training Set was preprocessed. In fact, each tweet has been transformed into a plain text, without taking into consideration any characteristic of a 'social' language. This may have positively influenced the model in predicting the out-of-domain classification.
A further observation to be made about the dataset concerns a lack of correlation between the use of *bad words* and the presence of hateful contents in a phrase. This fact shows how Offensive Language Detection and Hate Speech Detection are related topics, but they remain two distinct tasks (Davidson et al., 2017). Also, many times these kinds of bad words are probably used in an ironic way or to emphasize a sentence, especially in the Italian language.

7 Conclusion

The participation in the Hate Speech Detection 2020 competition proposed by Evalita is derived from purely academic purposes.
We focused on using different types and combinations of embeddings. Surprisingly, the best results were obtained with the use of Tfidf only instead of the use of a combination of more sophisticated embeddings such as GloVe and DistilBert. After a feature selection phase carried out through a Random Forest, the results obtained through a Linear SVM and a Logistic Regression were compared. The latter was the best.
We are aware that the presented system does not introduce new elements with respect to the state of the art of current technologies. Despite this, it was interesting to observe the different results obtained in relation to the composition of the Test Set.

The project was completely developed in python, and the code is publicly available at the following link:

https://github.com/eliabisconti/haspeede

References

Pinkesh Badjatiya, Shashank Gupta, Manish Gupta, and Vasudeva Varma. 2017. Deep learning for hate speech detection in tweets. In *Proceedings of the 26th International Conference on World Wide Web Companion*, pages 759–760.

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.

Steven Bird, Ewan Klein, and Edward Loper. 2009. *Natural language processing with Python: analyzing text with the natural language toolkit*. " O'Reilly Media, Inc.".

Thomas Davidson, Dana Warmsley, Michael Macy, and Ingmar Weber. 2017. Automated hate speech detection and the problem of offensive language. *arXiv preprint arXiv:1703.04009*.

Armand Joulin, Edouard Grave, Piotr Bojanowski, and Tomas Mikolov. 2016. Bag of tricks for efficient text classification. *CoRR*, abs/1607.01759.

Jeffrey Pennington, Richard Socher, and Christopher D Manning. 2014. Glove: Global vectors for word representation. 14:1532–1543.

Shahzad Qaiser and Ramsha Ali. 2018. Text mining: Use of tf-idf to examine the relevance of words to documents. *International Journal of Computer Applications*, 181, 07.

Manuela Sanguinetti, Gloria Comandini, Elisa Di 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. 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.

Victor Sanh Lysandre Debut Julien Chaumond Thomas Wolf. 2019. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter.