Merging datasets for hate speech classification in Italian

Paula Fortuna[1] **Ilaria Bonavita**[2] **Sérgio Nunes**[1,3]

(1) INESC TEC and (3) FEUP, University of Porto
Rua Dr. Roberto Frias, s/n 4200-465 Porto PORTUGAL
paula.fortuna@fe.up.pt, sergio.nunes@fe.up.pt
(2) Eurecat, Centre Tecnològic de Catalunya
Carrer de Bilbao, 72, 08005 Barcelona

Abstract

This paper presents an approach to the shared task HaSpeeDe within Evalita 2018. We followed a standard machine learning procedure with training, validation, and testing phases. We considered word embedding as features and deep learning for classification. We tested the effect of merging two datasets in the classification of messages from Facebook and Twitter. We concluded that using data for training and testing from the same social network was a requirement to achieve a good performance. Moreover, adding data from a different social network allowed to improve the results, indicating that more generalized models can be an advantage.

ll manoscritto presenta un approccio per la risoluzione dello shared task HaSpeeDe organizzato all'interno di Evalita 2018. La classificazione è stata condotta con caratteristiche del testo estratte con word embedding e utilizzando algoritmi di deep learning. Abbiamo voluto sperimentare l'effetto dell'integrazione di messaggi di Facebook e Twitter ha e abbiamo ottenuto due risultati. 1) Addestrare modelli con un dataset integrato migliora le performance di classificazione in datasets provenienti dai singoli social network suggerendo una migliore capacità di generalizzazione del modello. 2) Tuttavia, utilizzare modelli addestrati su datasets provenienti da un social network per classificare messaggi provenienti da un altro social network comporta un peggioramento delle performance indicando che è indispensabile includere nel train set messaggi dello stesso social network che si è interessati a classificare nel test set.

1 Introduction

In the last few years, there is a growing attention to the automatic detection of hate speech in text. This appears as an answer to the increased spreading of online abuse in social networks. Several evaluation initiatives have been presenting different yet related classification tasks, e.g. TRAC (Kumar et al., 2018). Shared initiatives such as this, have the advantage of promoting the development of different but comparable solutions for the same problem, within a short period of time. In this paper, we describe the participation of the "Stop PropagHate" team in the HaSpeeDe task within Evalita 2018 (Bosco et al., 2018).

The goal of this task is to improve the automatic classification of hate speech in Italian. More specifically, there were three sub-tasks, promoting the development of features that would work independently of social network. For the task HaSpeeDe-FB, only the Facebook dataset could be used to train the model and classify Facebook data; for HaSpeeDe-TW, only the Twitter dataset could be used to classify Twitter data; and for the Cross-HaSpeeDe, only the Facebook dataset could be used to classify the Twitter and vice versa.

In our approach, we focused on understanding the effects of merging the two provided datasets. As features, we used word embeddings and deep learning for classification with a simple dense neural network. In this paper, we present the details of our approach, our results and conclusions.

2 Related Work

Previous research in the field of automatic detection of hate speech can give us insight into how to approach this problem. Two surveys summarize previous research and conclude that the approaches rely frequently on Machine Learning and classification (Schmidt and Wiegand, 2017; Fortuna and Nunes, 2018).

Regarding the automatic classification of messages, one first step is the gathering of training data. Several studies published datasets considering hate speech with different classification systems (Ross et al., 2017; Waseem and Hovy, 2016; Davidson et al., 2017; Nobata et al., 2016; Jigsaw, 2018). Although these could be useful datasets, the annotated language is not Italian. Regarding this language, the two existent datasets are used in this task (Del Vigna et al., 2017; Poletto et al., 2017; Sanguinetti et al., 2018).

After data collection, one of the most important steps when using classification is the process of feature extraction (Schmidt and Wiegand, 2017). Different methods are used, for instance word and character n-grams (Liu and Forss, 2014), perpetrator characteristics (Waseem and Hovy, 2016), othering language (Burnap and Williams, 2016) or word embedings (Djuric et al., 2015). Regarding the classification algorithms, the more common are, for instance, SVM (Del Vigna et al., 2017) or Random forests (Burnap and Williams, 2014). Another popular approach, due to its good results, is deep learning (Yuan et al., 2016; Gambäck and Sikdar, 2017; Park and Fung, 2017).

Different studies proved that deep learning algorithms outperform previous approaches. This was the case when using character or token-based n-grams with Recurrent Neural Network Language Model (RNN) (Mehdad and Tetreault, 2016); user behavioral characteristics with neural network composed of multiple Long-Short-Term-Memory (LSTM) (Park and Fung, 2017); Convolutional Neural Networks (CNN), LSTM and FastText (Badjatiya et al., 2017); morpho-syntactical features, sentiment polarity and word embedding lexicons with LSTM (Del Vigna et al., 2017); users' tendency towards racism or sexism with RNN (Pitsilis et al., 2018); abusive behavioral norms, available metadata, patterns within the text with RNN (Founta et al., 2018); n-grams, tf-idf, POS, sentiment, misspellings, emojis, special punctuation, capitalization, hashtags with CNN and GRU (Zhang et al., 2018); and word2vec with Convolutional Neural Networks (CNN) (Gambäck and Sikdar, 2017).

In this work, we propose an innovative approach in hate speech detection by merging different datasets, in the sequence of a previous experiment (Fortuna et al., 2018). We merged two datasets for aggression classification and the results showed that, although training with similar data is an advantage, adding data from different platforms allowed slightly better results.

Regarding the specificities of our approach in this contest, the main research question of our work concerns the effects of merging new datasets on the performance of models for hate speech classification. Accordingly with the previous study, we hypothesize that merging datasets will lead to a better performance. Additionally, we want to investigate how models perform when only data from different sources was used in the training.

In the next sections, we present our methodology and approach to this problem.

3 Methodology

3.1 Data

The data proposed for this task results of joining a collection of Facebook comments from 2016 (Del Vigna et al., 2017) with a Twitter corpus developed in 2018 (Poletto et al., 2017; Sanguinetti et al., 2018). Both consist of a total amount of 4,000 comments/tweets, randomly split into development and test set, of 3,000 and 1,000 messages respectively. The data format is the same with three tab-separated columns, each one representing the ID of the message, the text and the class (1 if the text contains hate speech, and 0 otherwise).

3.2 Text pre-processing

As a first step, we load the messages, remove the retweet marker "RT" in case of the tweets, and also the URL links present in the text.

3.3 Feature extraction and classification

We follow a methodology of classification with training, testing and validation. Keeping 30% of the data for validation allows us to estimate the results we would achieve in the contest. We use word embeddings and deep learning as presented in previous literature (Chollet and Allaire, 2018). We use the keras R package (Allaire et al., 2018) and make our approach available in a public repository[1].

3.3.1 Word embedding

In the procedure of feature extraction, we vectorize the text. We start by tokenizing the

[1] https://github.com/StopPropagHate/experiment_evalita_HaSpeeDe

data, considering only the top 10,000 words in the dataset. Additionally, we consider only the first 100 words of the tweets. We use the functions text_tokenizer, fit_text_tokenizer, texts_to_sequences and pad_sequences in our extraction.

3.3.2 Deep Learning

For the classification we use 10 fold cross-validation and apply a simple dense neural network. We use binary_crossentropy for loss with the rmsprop optimizer, we define the custom metric F1, so that it would be in according to the contest used metric. Regarding the model, we instantiate an empty model and we customize it:

- First we add an embedding layer where we specify the input_length (100, the maximum length of the messages) and give the dimensionality of the input data (dimensional space of 10,000). We add a dropout of 0.25.
- We flatten the output and add a dense layer, specified with 256 unit, with "relu" as a parameter. We add a dropout of 0.25.
- We add a dense layer with just a single neuron to serve as the output layer. Aiming for a single output, we use a sigmoid activation.

We use keras_compile function to compile and fit the model. We use batch size 128 and we tune the number of epochs starting by using 10. We also feed the model with the classes weights, corresponding to the frequencies of the classes in the training set. We average the F1 and loss results of the 10 folds for each epoch. For the epoch number, we kept the maximum number before overfitting to happen (the results only improving in the training set, but not in the test set). We save the final model and apply it to the validation data, with the function keras_predict. We conduct a permutation test in order to have a p-value associated to the F1.

4 Tasks and runs description

We conduct three different experiments following the procedure described in Section 3.

Task HaSpeeDe-FB In the HaSpeeDe-FB run1, we train and test with Facebook data. In the HaSpeeDe-FB run2, we mix Facebook with the Twitter provided data and see the effect in predicting hate speech in Facebook.

Task HaSpeeDe-TW We follow a similar procedure, but we switched the roles of Facebook and Twitter data. For theHaSpeeDe-TW run1 only Twitter data is used. In a second run HaSpeeDe-TW run2, we mix data for training and use Twitter for testing.

Task Cross-HaSpeeDe This is a proposed out-of-domain task. In the Cross-HaSpeeDe-FB, only the Facebook dataset can be used to classify Twitter data. In the Cross-HaSpeeDe-TW, only the Twitter dataset is used to classify Facebook data.

5 Results and Discussion

We separated the conditions with testing data from Facebook from Twitter and we compared three different conditions: the training data is from the same social network (1), the training data is both from and not from the social network (2), and the training data is not from the social network (3).

5.1 Results for Tuning and Validation

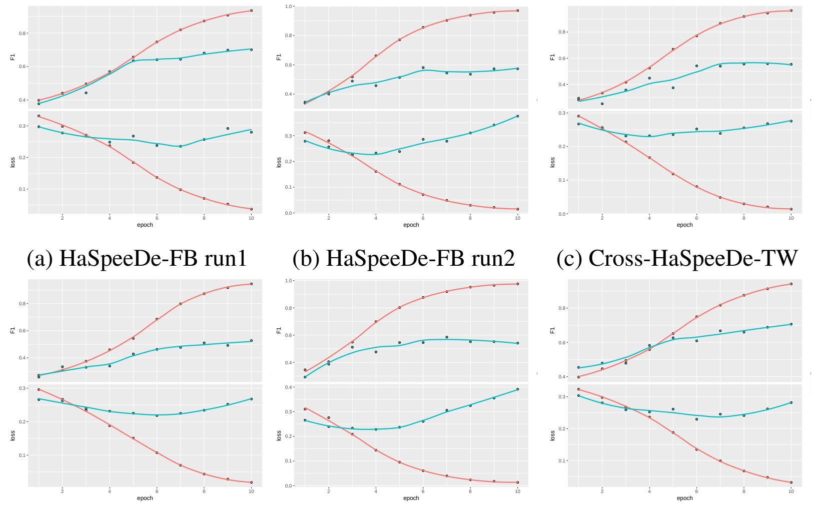
For each of the runs in our experiment we tuned the epoch parameter and we analyzed the average of the 10 folds for each of the 10 epochs (Figure 1). The decided number of epochs for each run is presented in the Table 1. We concluded that using mixed data for training (Condition 2) has a better performance (F1) than using data only from the social network (Condition 1). Additionally, using data only from other social network (Condition 3) provided poor results. Finally, classifying Facebook data was easier than Twitter data.

C.	**system**	**epoch**	**F1**	***p-value***
1	HaSpeeDe-FB run1	7	0.723	0.001
2	**HaSpeeDe-FB run2**	**3**	**0.738**	**0.001**
3	Cross-HaSpeeDe-TW	4	0.284	0.001
1	HaSpeeDe-TW run1	6	0.630	0.001
2	**HaSpeeDe-TW run2**	**4**	**0.679**	**0.001**
3	Cross-HaSpeeDe-FB	6	0.434	1

Table 1: F1 and respective *p-value* achieved in the validation set and respective Condition (C.).

5.2 Contest Results

Regarding the contest results (Table 2), similarly to the validation results we verified again that using mixed data for training (Condition 2) is better. Also in this case we verified that using only data from a different social network provided much worse results (Condition 3). Opposing to the validation results we found here that generally classifying Facebook data was more difficult than Twitter data.



(a) HaSpeeDe-FB run1 (b) HaSpeeDe-FB run2 (c) Cross-HaSpeeDe-TW

(d) HaSpeeDe-TW run1 (e) HaSpeeDe-TW run2 (f) Cross-HaSpeeDe-FB

Figure 1: Average of the 10 folds, for the metric F1 and loss, both for the training folds (blue) and validation fold (red). The results present each of the runs submitted by the team.

			Not HS			**HS**			
Test data	**C.**	**Run**	**Precision**	**Recall**	**F-score**	**Precision**	**Recall**	**F-score**	**Macro-Avg F-score (P.)**
Facebook	1	HaSpeeDe-FB run1	0,478	0,7089	0,571	0,8195	0,6307	0,7128	0,6419 (13)
	2	**HaSpeeDe-FB run2**	**0,4923**	**0,6965**	**0,5769**	**0,8195**	**0,6573**	**0,7295**	**0,6532 (12)**
	3	Cross-HaSpeeDe_TW	0,3606	0,9133	0,517	0,8461	0,2274	0,3585	0,4378 (11)
Twitter	1	HaSpeeDe-TW run1	0,7952	0,8964	0,8428	0,7058	0,5185	0,5978	0,7203 (11)
	2	**HaSpeeDe-TW run2**	**0,8628**	**0,7721**	**0,8149**	**0,6101**	**0,7438**	**0,6703**	**0,7426 (10)**
	3	Cross-HaSpeeDe_FB	0,6579	0,3727	0,4759	0,3128	0,5956	0,4102	0,443 (12)

Table 2: Macro Averaged F score and position (P.) achieved in the contest, respective Condition (C.) and Run. Precision, Recall and F-score are also provided for each of the classes hate speech (HS) and not hate speech (Not HS).

Regarding the main finding of this experiment, the results show that in this contest adding new data from a different social network brought improved performance. However, in the scope of this work it was not possible to investigate the reasons for this. One possibility may be the increased number of instances in the training when adding new datasets. Also using data from a different social network may bring less overfitting from training with only a dataset.

6 Conclusion

Throughout our approach to this shared task, our goal was to measure the effects of merging new datasets on hate speech classification. Supported by a previous experiment, we expected that adding data would help the classification. Indeed, we verified that merging datasets allowed us to have a small improvement of the results.

Complementary to this result, we tried the same approach following the same method and idea, in the Evalita 2018 AMI task. Merging datasets did not help for misoginy classification. In this case, we found that merging extra misogynistic or hate speech data kept the mysoginy classification with similar performance.

The reason why merging datasets worked in one case and not in the other remains unclear, and requires exploration in future studies. Possible variables interfering are the number of messages used for training and also the number of distinct words in the data.

Acknowledgments

This work was partially funded by the Google DNI grant Stop PropagHate.

References

J. J. Allaire, Francois Chollet, Yuan Tang, Daniel Falbel, Wouter Van Der Bijl, and Martin Studer. 2018. R interface to 'keras'. *Computer software manual](R package version 2.1.6). Retrieved from https://CRAN. R-project. org/package= keras.*

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. International World Wide Web Conferences Steering Committee.

Cristina Bosco, Felice DellOrletta, Fabio Poletto, Manuela Sanguinetti, and Maurizio Tesconi. 2018. Overview of the EVALITA 2018 Hate Speech Detection Task. In Tommaso Caselli, Nicole Novielli, Viviana Patti, and Paolo Rosso, editors, *Proceedings of the 6th evaluation campaign of Natural Language Processing and Speech tools for Italian (EVALITA'18)*, Turin, Italy. CEUR.org.

Peter Burnap and Matthew L. Williams. 2014. Hate speech, machine classification and statistical modelling of information flows on Twitter: Interpretation and communication for policy decision making. In *Proceedings of Internet, Policy & Politics*, pages 1–18.

Pete Burnap and Matthew L. Williams. 2016. Us and them: identifying cyber hate on Twitter across multiple protected characteristics. *EPJ Data Science*, 5(1):11.

Francois Chollet and J. J. Allaire. 2018. *Deep Learning with R*. Manning Publications Co., Greenwich, CT, USA, 1st edition.

Thomas Davidson, Dana Warmsley, Michael Macy, and Ingmar Weber. 2017. Automated Hate Speech Detection and the Problem of Offensive Language. In *Proceedings of ICWSM*.

Fabio Del Vigna, Andrea Cimino, Felice Dell'Orletta, Marinella Petrocchi, and Maurizio Tesconi. 2017. Hate me, hate me not: Hate speech detection on facebook. In *Proceedings of the First Italian Conference on Cybersecurity*, pages 86–95.

Nemanja Djuric, Jing Zhou, Robin Morris, Mihajlo Grbovic, Vladan Radosavljevic, and Narayan Bhamidipati. 2015. Hate speech detection with comment embeddings. In *Proceedings of the 24th International Conference on World Wide Web*, pages 29–30. ACM2.

Paula Fortuna and Sérgio Nunes. 2018. A survey on automatic detection of hate speech in text. *ACM Computing Surveys (CSUR)*, 51(4):85.

Paula Fortuna, José Ferreira, Luiz Pires, Guilherme Routar, and Sérgio Nunes. 2018. Merging datasets for aggressive text identification. In *Proceedings of the First Workshop on Trolling, Aggression and Cyberbullying (TRAC-2018)*, pages 128–139.

Antigoni-Maria Founta, Despoina Chatzakou, Nicolas Kourtellis, Jeremy Blackburn, Athena Vakali, and Ilias Leontiadis. 2018. A unified deep learning architecture for abuse detection. *arXiv preprint arXiv:1802.00385*.

Björn Gambäck and Utpal Kumar Sikdar. 2017. Using Convolutional Neural Networks to Classify Hate-speech. In *Proceedings of the First Workshop on Abusive Language Online*, pages 85–90.

Jigsaw. 2018. Toxic comment classification challenge identify and classify toxic online comments. Available in `https://www.kaggle.com/c/jigsaw-toxic-comment-classification-challenge`, accessed last time in 23 May 2018.

Ritesh Kumar, Atul Kr. Ojha, Shervin Malmasi, and Marcos Zampieri. 2018. Benchmarking Aggression Identification in Social Media. In *Proceedings of the First Workshop on Trolling, Aggression and Cyberbulling (TRAC)*, Santa Fe, USA.

Shuhua Liu and Thomas Forss. 2014. Combining n-gram based similarity analysis with sentiment analysis in web content classification. In *International Joint Conference on Knowledge Discovery, Knowledge Engineering and Knowledge Management*, pages 530–537.

Yashar Mehdad and Joel Tetreault. 2016. Do characters abuse more than words? In *Proceedings of the SIGdial 2016 Conference: The 17th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 299–303.

Chikashi Nobata, Joel Tetreault, Achint Thomas, Yashar Mehdad, and Yi Chang. 2016. Abusive language detection in online user content. In *Proceedings of the 25th International Conference on World Wide Web*, pages 145–153. International World Wide Web Conferences Steering Committee.

Ji Ho Park and Pascale Fung. 2017. One-step and Two-step Classification for Abusive Language Detection on Twitter. In *Proceedings of the First Workshop on Abusive Language Online*.

Georgios K Pitsilis, Heri Ramampiaro, and Helge Langseth. 2018. Detecting offensive language in tweets using deep learning. *arXiv preprint arXiv:1801.04433*.

Fabio Poletto, Marco Stranisci, Manuela Sanguinetti, Viviana Patti, and Cristina Bosco. 2017. Hate speech annotation: Analysis of an italian Twitter corpus. In *CEUR WORKSHOP PROCEEDINGS*, volume 2006, pages 1–6. CEUR-WS.

Bjorn Ross, Michael Rist, Guillermo Carbonell, Benjamin Cabrera, Nils Kurowsky, and Michael Wojatzki. 2017. Measuring the reliability of hate speech annotations: The case of the european refugee crisis. *arXiv preprint arXiv:1701.08118*.

Manuela Sanguinetti, Fabio Poletto, Cristina Bosco, Viviana Patti, and Marco Stranisci. 2018. An italian Twitter corpus of hate speech against immigrants. In *Proceedings of LREC*.

Anna Schmidt and Michael Wiegand. 2017. A survey on hate speech detection using natural language processing. *SocialNLP 2017*, page 1.

Zeerak Waseem and Dirk Hovy. 2016. Hateful symbols or hateful people? predictive features for hate speech detection on Twitter. In *Proceedings of NAACL-HLT*, pages 88–93.

Shuhan Yuan, Xintao Wu, and Yang Xiang. 2016. A two phase deep learning model for identifying discrimination from tweets. In *International Conference on Extending Database Technology*, pages 696–697.

Ziqi Zhang, David Robinson, and Jonathan Tepper. 2018. Detecting hate speech on Twitter using a convolution-gru based deep neural network. In *European Semantic Web Conference*, pages 745–760. Springer.