RuG @ EVALITA 2018: Hate Speech Detection In Italian Social Media

Xiaoyu Bai[*], **Flavio Merenda**[*∓], **Claudia Zaghi**[*], **Tommaso Caselli**[*], **Malvina Nissim**[*]
[*] Rikjuniversiteit Groningen, Groningen, The Netherlands
[∓] Università degli Studi di Salerno, Salerno, Italy
f.merenda|t.caselli|m.nissim@rug.nl x.bai.5|c.zaghi@student.rug.nl

Abstract

English. We describe the systems the RuG Team developed in the context of the Hate Speech Detection Task in Italian Social Media at EVALITA 2018. We submitted a total of eight runs, participating in all four subtasks. The best macro-F1 score in all subtasks was obtained by a Linear SVM, using hate-rich embeddings. Our best system obtains competitive results, by ranking 6th (out of 14) in HaSpeeDe-FB, 3rd (out of 15) in HaSpeeDe-TW, 8th (out of 13) in Cross-HaSpeeDe_FB, and 6th (out of 13) in Cross-HaSpeeDe_TW.

Italiano. *Illustriamo i dettagli dei due sistemi che il Team RuG ha sviluppato nell'ambito dell'esercizio di valutazione su riconoscimento di messagi d'odio in testi da Social Media per l'italiano. Abbiamo partecipato a tutti e quattro i sottotask, inviando un totale di otto predizioni. La migliore macro-F1, è ottenuta da un SVM che usa embedding polarizzati, costruiti sfruttando contenuto ricco di odio. Il nostro miglior sistema ha ottenuto dei risultati competitivi, classificandosi 6° (su 14) in HaSpeeDe-FB, 3° (su 15) in HaSpeeDe-TW, 8° (su 13) nel Cross-HaSpeeDe_FB, e 6° (su 13) in Cross-HaSpeeDe_TW.*

1 Introduction

The use of "bad" words and "bad" language has been the battleground for freedom of speech for centuries. The spread of Social Media platforms, and especially of micro-blog platforms (e.g. Facebook and Twitter), has favoured the growth of online hate speech. Social media sites and platforms have been urged to deal with and remove offensive and/or abusive content but the phenomenon is so pervasive that developing systems that automatically detect and classify offensive on-line content has become a pressing need (Bleich, 2014; Nobata et al., 2016; Kennedy et al., 2017).

The Natural Language Processing and Computational Social Science communities have been receptive to such urgency, and the automatic detection of abusive and/or offensive language, trolling, and cyberbulling (Waseem et al., 2017; Schmidt and Wiegand, 2017) has seen a growing interest. This has taken various forms: datasets in multiple languages[1], thematic workshops[2], and shared evaluation exercises, such as the GermEval 2018 Shared Task (Wiegand et al., 2018), and the SemEval 2019 Task 5: HateEval[3] and Task 6: OffensEval[4]. The EVALITA 2018 Hate Speech Detection task (haspeede)[5] (Bosco et al., 2018) also falls in the latter category, and focuses on the automatic identification of hate messages from Facebook comments and tweets in Italian. We participated in this shared task with two different models, exploiting the concept of *polarised embeddings* (Merenda et al., 2018). The details of our participation are the core of this paper. Code and outputs are available at https://github.com/tommasoc80/evalita2018-rug.

2 Task

The haspeede task derives from the harmonization process of originally separate annotation efforts from two research groups, converging onto a uniform label granularity (Del Vigna et al., 2017; Poletto et al., 2017; Sanguinetti et al., 2018). For details on the data see Section 3.1, and the task

[1] http://bit.ly/2RZUlKH
[2] https://sites.google.com/view/alw2018
[3] http://bit.ly/2EEC7Me
[4] http://bit.ly/2P7pTQ9
[5] http://di.unito.it/haspeedeevalita18

overview paper (Bosco et al., 2018).

The hate detection task is articulated in four binary (hate vs non-hate) sub-tasks, two in-domain, two cross-domain. The in-domain sub-tasks require training and test data to belong to the same text type, either Facebook (HaSpeeDe-FB) or Twitter (HaSpeeDe-TW), while the cross-domain sub-tasks require training on one text type and testing on the other: Facebook-Twitter (Cross-HaSpeeDe_FB) and Twitter-Facebook (Cross-HaSpeeDe_TW).

3 Data and Resources

All of our runs for all subtasks are based on supervised approaches, where data (and features) play a major role for the final results of a system. Furthermore, our contribution adopted a closed-task setting, i.e. we did not include any training data beyond what was provided within the task. We did however build enhanced distributed representations of words exploiting additional data (see Section 3.2). This section illustrates the datasets and language resources used in our submissions.

3.1 Resources Provided by the Organisers

The organizers provided a total of 6,000 labeled Italian messages for training, split as follows: 3,000 comments from Facebook, and 3,000 messages from Twitter. For test, they subsequently made available 1000 instances for each text type. Table 1 illustrates the distribution of the classes in the different text types both in training and test data. Note that the distribution of labels in the test data is unknown at developing time.

Table 1: Distribution of the labeled samples in the training and test data per text type.

Text type	Class	Training	Test
Facebook	non-hate	1,618	323
	hate	1,382	677
Twitter	non-hate	2,028	676
	hate	972	324

Although the task organisers have balanced the datasets with respect to size, and have adopted the same annotation granularity (hate vs. non-hate), the two datasets are very different both in terms of class distribution (i.e. 46.06% of messages labelled as hateful in Facebook vs. 32.40% in Twitter in training) and with regard to their contents. For instance, the Facebook data is concerned with *general topics* that may contain hateful messages such as immigration, religion, politics, gender issues, while the Twitter dataset is focused on *specific targets*, i.e., categories or groups of individuals who are likely to become victims of hate speech (migrants, Muslims, and Roma[6]). It is also interesting to note that the label distribution in the Facebook test data is flipped compared to training, with a strong majority of hateful comments.

3.2 Additional Resources: Source-Driven Embeddings

We addressed the task by adopting a closed-task setting. However, as a strategy to potentially increase the generalization capabilities of our systems and tune them towards better recognition of hate content, we developed hate- and offense-sensitive word embeddings.

To do so, we scraped comments from a list of selected Facebook pages likely to contain offensive and/or hateful content in the form of comments to posts, extracting over 1M comments. We built word embeddings over the acquired data with the `word2vec` tool skip-gram model (Mikolov et al., 2013), using 300 dimensions, a context window of 5, and minimum frequency 1. In the remainder of this paper we refer to these representations as "hate-rich embeddings". More details on the creation process, including the complete list of Facebook pages used, and a preliminary evaluation of these specialised representations can be found in (Merenda et al., 2018).

4 Systems and Runs

We detail in this section our final submissions. The models have been developed in parallel to our participating systems at the GermEval 2018 Shared Task (Bai et al., 2018), sharing with them some core aspects.

4.1 Run 1: Binary SVM

Our first model is a Linear Support Vector Machine (SVM), built using the LinearSVC scikit learn implementation (Pedregosa et al., 2011).

We performed minimal pre-processing by removing stop words using the Python module `stop-words`[7], and lowercasing the tokens.

[6]The Romani, Romany, or Roma are an ethnic group of traditionally itinerant people who originated in northern India and are nowadays subject to ethnic discrimination.

[7]`https://pypi.org/project/stop-words/`

We used two groups of surface features, namely: i.) word n-grams in the range 1–3; and ii.) character n-grams in the range 2–4. The sparse vector representation of each (training) instance is then concatenated with its dense vector representation, as follows: for every word *w* in an instance *i*, we derived a 300 dimension representation, $\vec{w}$, by means of a look-up in the hate-rich embeddings. We performed max pooling over these word embeddings, $\vec{w}$, to obtain a 300 dimension representation of the full instance, $\vec{i}$. Words not covered in the hate-oriented embeddings are ignored. Finally, class weights are balanced and SVM parameters use default values ($C = 1$).

4.2 Run 2: Binary Ensemble Model

Our second submission uses a binary ensemble model, which combines a Convolutional Neural Network (CNN) system and the linear SVM (Section 4.1), with a logistic regression meta-classifier on top. Predictions on training data are obtained via ten-fold cross-validation.

In the ensemble model, each input instance to the meta-classifier is represented by the concatenation of four features: a) the class predictions for that instance made by the SVM, b) the predictions of the CNN, and c) two additional surface-level features: the instance's length in terms of characters and the percentage of offensive terms in the instance. This latter feature is obtained via a look-up in a list of offensive terms in Italian obtained from the article *Le Parole per ferire* by Tullio De Mauro[8] and the "bad words" category in the Italian Wiktionary. The feature is expressed by the ratio between the frequency of any of the instance's tokens comprised in the list and the instance's length in terms of tokens. Figure 1 shows the features fed to the ensemble meta-classifier.

The CNN is an adaptation of available architectures for sentence classification (Kim, 2014; Zhang and Wallace, 2015), using Keras (Chollet and others, 2015), and is composed of: i.) a word embeddings input layer using the hate-rich embeddings; ii.) a single convolutional layer; iii.) a single max-pooling layer; iv.) a single fully-connected layer; and v.) a sigmoid output layer.

The max-pooling layer output is flattened, concatenated, and fed to the fully-connected layer composed of 50 hidden-units with the ReLU activation function. The final output layer with the sigmoid activation function computes the distribution of the two labels. (Other network hyperparameters: `Number of filters`: 6; `Filter sizes`: 3, 5, 8; `Strides`: 1). We used binary cross-entropy as loss function and Adam as optimiser. In training, we set a batch size of 64 and ran it for 10 epochs. We also applied two dropouts: 0.6 between the embeddings and the convolutional layer, and 0.8 between the max-pooling and the fully-connected layer.

[8] `https://bit.ly/2J4TPag`

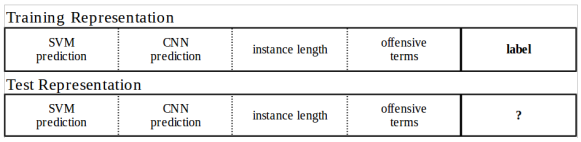

Figure 1: Feature representation of each sample fed to the ensemble model. On top, the representation of a training sample, on bottom, the representation of a test sample.

5 Results and Ranking

Table 2 reports the results and ranking for our runs for all four subtasks. We also include the scores of the CNN (not submitted to the official competition), marked with a $*$.[9]

Table 2: System results and ranking, including the out-of-competition runs for CNN alone.

Subtask	Model[10]	Rank	Macro F1
HaSpeeDe-FB	SVM	6/14	**0.7751**
	Ensemble	9/14	0.7428
	CNN*	n/a	0.7138
HaSpeeDe-TW	SVM	3/15	**0.7934**
	Ensemble	9/15	0.7530
	CNN*	n/a	0.7363
Cross-HaSpeeDe_FB	SVM	8/13	**0.5409**
	Ensemble	9/13	0.4845
	CNN*	n/a	0.4692
Cross-HaSpeeDe_TW	SVM	6/13	**0.6021**
	Ensemble	7/13	0.5545
	CNN*	n/a	0.6093

The SVM models obtain, by far, better results than the Ensemble models. It is likely that the Ensemble systems suffer from the lower performances of

[9] Being allowed to submit a maximum of two runs per subtask, we based our choice of models on the results of a 10-fold cross validation of the three architectures on the training data.

[10] The SVM correposnds to run id 1 and the Ensemble model to run id 3 in the official submitted runs - see Submissions-Haspeede in the GitHub repository `https://github.com/tommasoc80/evalita2018-rug/tree/master/Submissions-Haspeede`

the CNN. We also observe differences in performance on the two datasets across the subtasks.

Table 3: SVM's performance per class

Subtask	non-hate		hate	
	P	R	P	R
HaSpeeDe-FB	0.6990	0.6904	0.8531	0.8581
HaSpeeDe-TW	0.8577	0.8831	0.7401	0.6944
CrossHaSpeeDe_FB	0.8318	0.4023	0.3997	0.8302
CrossHaSpeeDe_TW	0.4375	0.6934	0.7971	0.5745

In-domain, in absolute terms, we do better on Twitter (.7934) than on Facebook (.7751), and this is even truer in relative terms, as performance overall in the competition is better on Facebook (best: 0.8288) than on Twitter (best: 0.7993). Our high score on HaSpeeDe-TW comes from high precision and recall on non-hate, while for HaSpeeDe-FB, we do well on the hate class. This can be due to label distribution (hate is always minority class, but more balanced in Facebook), but also to the fact that we use Facebook-based hate-rich embeddings, which might push towards better hate detection.

Cross-domain, results are globally lower, as expected, with best scores on Cross-HaSpeeDe_FB and Cross-HaSpeeDe_TW of 0.6541 and 0.6985, respectively (Bosco et al., 2018). Our models experience a more substantial loss when trained on Facebook and tested on Twitter (in Cross-HaSpeeDe_FB we lose over 25 percentage points compared to HaSpeeDe-TW, where the Twitter test set is the same), than viceversa (we lose ca. 17 percentage points on the Facebook test set).

6 Discussion

The drop in performance in the cross-domain settings is likely due to topics, and data collection strategies (general topics on Facebook, specific targets on Twitter). In other words, despite the use of hate-rich embeddings as a strategy to make the systems generalize better, our models remain too sensitive to training data, which is strongly represented as word and character n-grams.

The impact of the hate-rich embeddings is most strongly seen in HaSpeeDe-FB and Cross-HaSpeeDe_FB, with recall for the hate class being substantially higher than for the non-hate class. This could be due to the fact that the hate-rich embeddings have been generated from comments in Facebook pages, that is, the same text type as the training data in the two tasks, so that possibly some jargon and topics are shared. While this has a positive effect when training and testing on Facebook (HaSpeeDe-FB), it has instead a detrimental effect when testing on Twittter (Cross-HaSpeeDe_FB), since this dataset has a large majority of non-hate instances, and we tend to over-predict the hate class (see Table 3).

In HaSpeeDe-TW and Cross-HaSpeeDe_TW (training on Twitter) the impact of the hate-rich embeddings is a lot less clear. Indeed, recall for the hate class is always lower than non-hate, with the large majority of errors (more than 50% in all runs) being hate messages wrongly classified as non-hateful, thus seemingly just following the class imbalance of the Twitter trainset.

In both datasets, hate content is expressed either in a direct way, by means of "bad words" or direct insults to the target(s), or more implicitly and subtly. This latter type of hate messages is definitely the main source of errors for our systems in all subtasks. Finally, we observe that in some cases the annotation of messages as hateful is subject to disagreement and debate. For instance, all messages containing the word *rivoluzione* [revolution] are marked as hateful, even though there is a lack of linguistic evidence.

7 Conclusion and Future Work

Developing our systems for the Hate Speech Detection in Italian Social Media task at EVALITA 2018, we focused on the generation of distributed representations of text that could not only enhance the generalisation power of the models, but also better capture the meaning of words in hate-rich contexts of use. We did so exploiting Facebook on-line communities to generate *hate-rich embeddings* (Merenda et al., 2018).

A Linear SVM system outperformed a meta-classifer that used predictions from the SVM itself, and a CNN, due to the low performance of the CNN component. Major errors of the systems are due to implicit hate messages, where even the hate-rich embeddings fail. A further aspect to consider in this task is the difference in text type and class balance of the two datasets. Both of these aspects have a major impact on system performance in the cross-genre settings.

Finally, to better generalize to unseen data and genres, future work will focus on developing systems able to further abstract from the actual lexical content of the messages by capturing general

writing patterns of haters. One avenue to explore in this respect is "bleaching" text (van der Goot et al., 2018), a newly suggested technique used to fade the actual strings into more abstract, signal-preserving representations of tokens.

References

Xiaoyu Bai, Flavio Merenda, Claudia Zaghi, Tommaso Caselli, and Malvina Nissim. 2018. RuG at GermEval: Detecting Offensive Speech in German Social Media. In Josef Ruppenhofer, Melanie Siegel, and Michael Wiegand, editors, *Proceedings of the GermEval 2018 Workshop*.

Erik Bleich. 2014. Freedom of expression versus racist hate speech: Explaining differences between high court regulations in the usa and europe. *Journal of Ethnic and Migration Studies*, 40(2):283–300.

Cristina Bosco, Fabio Poletto Dell'Orletta, Felice, Manuela Sanuguinetti, and Maurizio Tesconi. 2018. Overview of the EVALITA 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.

François Chollet et al. 2015. Keras. `https://keras.io`.

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 (ITASEC17), Venice, Italy, January 17-20, 2017*, pages 86–95.

George Kennedy, Andrew McCollough, Edward Dixon, Alexei Bastidas, John Ryan, Chris Loo, and Saurav Sahay. 2017. Technology solutions to combat online harassment. In *Proceedings of the First Workshop on Abusive Language Online*, pages 73–77.

Yoon Kim. 2014. Convolutional neural networks for sentence classification. *arXiv preprint arXiv:1408.5882*.

Flavio Merenda, Claudia Zaghi, Tommaso Caselli, and Malvina Nissim. 2018. Source-driven Representations for Hate Speech Detection, proceedings of the 5th italian conference on computational linguistics (clic-it 2018). Turin, Italy.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

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.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. 2011. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.

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.

Manuela Sanguinetti, Fabio Poletto, Cristina Bosco, Viviana Patti, and Marco Stranisci. 2018. An Italian Twitter Corpus of Hate Speech against Immigrants. In Nicoletta Calzolari (Conference chair), Khalid Choukri, Christopher Cieri, Thierry Declerck, Sara Goggi, Koiti Hasida, Hitoshi Isahara, Bente Maegaard, Joseph Mariani, Hlne Mazo, Asuncion Moreno, Jan Odijk, Stelios Piperidis, and Takenobu Tokunaga, editors, *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan, May 7-12, 2018. European Language Resources Association (ELRA).

Anna Schmidt and Michael Wiegand. 2017. A survey on hate speech detection using natural language processing. In *Proceedings of the Fifth International Workshop on Natural Language Processing for Social Media. Association for Computational Linguistics, Valencia, Spain*, pages 1–10.

Rob van der Goot, Nikola Ljubešić, Ian Matroos, Malvina Nissim, and Barbara Plank. 2018. Bleaching text: Abstract features for cross-lingual gender prediction. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, volume 2, pages 383–389.

Zeerak Waseem, Thomas Davidson, Dana Warmsley, and Ingmar Weber. 2017. Understanding abuse: A typology of abusive language detection subtasks. In *Proceedings of the First Workshop on Abusive Language Online*, pages 78–84, Vancouver, BC, Canada, August. Association for Computational Linguistics.

Michael Wiegand, Melanie Siegel, and Josef Ruppenhofer. 2018. Overview. In Josef Ruppenhofer, Melanie Siegel, and Michael Wiegand, editors, *Proceedings of the GermEval 2018 Workshop*.

Ye Zhang and Byron Wallace. 2015. A sensitivity analysis of (and practitioners' guide to) convolutional neural networks for sentence classification. *arXiv preprint arXiv:1510.03820*.