UA at SemEval-2019 Task 5: Setting a Strong Linear Baseline for Hate Speech Detection

Carlos Perelló[♢] **David Tomás**[♢] **Alberto Garcia-Garcia**[♢]
Jose Garcia-Rodriguez[♢] **Jose Camacho-Collados**[♣]
[♢] University of Alicante, Spain
[♣] Cardiff University, United Kingdom
[♢]cpc69@alu.ua.es, dtomas@dlsi.ua.es, {agarcia, jgarcia}@dtic.ua.es,
[♣]camachocollados j@cardiff.ac.uk

Abstract

This paper describes the system developed at the University of Alicante (UA) for the SemEval 2019 Task 5: Multilingual detection of hate speech against immigrants and women in Twitter. The purpose of this work is to build a strong baseline for hate speech detection by means of a traditional machine learning approach with standard textual features, which could serve as a reference to compare with deep learning systems. We participated in both task A (Hate Speech Detection against Immigrants and Women) and task B (Aggressive behavior and Target Classification) for both English and Spanish. Given the text of a tweet, task A consists of detecting hate speech against women or immigrants in the text, whereas task B consists of identifying the target harassed as individual or generic, and to classify hateful tweets as aggressive or not aggressive. Despite its simplicity, our system obtained a remarkable macro-F1 score of 72.5 (sixth highest) and an accuracy of 73.6 (second highest) in Spanish (task A), outperforming more complex neural models from a total of 40 participant systems.

1 Introduction

Due to the massive rise of users in social media, the presence of verbal abuse, hate speech and bully-attitudes has increased over the years. A clear example is Twitter, where users find ways to anonymously harass and offend other individuals or collectives. This is especially troublesome as hate speech and hate crime are strongly related. Therefore, an early detection of hate speech could help prevent the subsequent hate crime. Online platforms like Twitter have been seeking to combat hate speech on their site, but it still requires a lot of manual work because there is not a reliable automatic method that can correctly identify hate speech behaviour. Building such automatic (or semi-automatic) systems is therefore essential to effectively fight this problem.

Hate speech detection is still a challenging task due to a number of reasons. First, hate speech content tends to be ambiguous and context-dependant (Chatzakou et al., 2017). Moreover, hate speech can cross sentence boundaries and be present in sarcastic comments in the same voice as the people that were producing abusive languages. These and other issues for detecting hate speech are discussed in more detail in Nobata et al. (2016).

In order to deal with these issues, over the past few years several techniques to detect hate speech and abusive language online have been proposed.[1] Previous works made use of heterogeneous features such as bag of words, n-grams, punctuation, as well as lexical features and user-related features (Chatzakou et al., 2017). In addition to these features, previous approaches showed the effectiveness of using word embeddings to detect abusive language in social media (Djuric et al., 2015) and exposed how sentiment analysis can also contribute to hate speech and offensive language detection (Nahar et al., 2012).

In this paper we build on these earlier works and propose a comprehensive framework to develop a traditional machine learning-based approach to hate speech detection, with the purpose of serving as a strong baseline for future systems using deep learning techniques. Our framework will be based on a linear classifier with standard textual features. As we will show throughout the paper, n-grams provide a reliable starting point when facing hate speech classification, and the performance can be further improved when combined with word em-

[1] Although related, it is important to distinguish between hate speech and abusive or offensive language. While the former is used to express hatred towards a targeted group based on characteristics such as race, ethnicity, gender, and sexual orientation, the latter can be used in the usual language of some users without being hateful (Davidson et al., 2017).

beddings and sentiment analysis features.

In particular for this work, we focus on hate speech against women and immigrants, following the Task 5 of SemEval 2019 (Basile et al., 2019). Indeed, race and gender hate speech has become an increasingly important issue in social media, as it stands for 50% of the targets of hate speech in Twitter (Silva et al., 2016). Code and pre-trained models are available at https://github.com/CPerelloC/UA-SemEval.

2 Hate Speech Detection System

In this section we present our hate speech detection model. The main goal of our model is to identify hate speech given a piece of text, in this case a tweet. A high-level overview of our model is presented in Section 2.1 and the set of features that are used in our model are described in Section 2.2.

2.1 Model

Our model consists of a linear classifier based on Support Vector Machines (SVM), which have proved to provide competitive results in text categorization since their conception (Joachims, 1998). The SVM classifier is trained on tweets containing hate speech annotations. During training, the model is fed with features relevant to hate speech detection. Then, in the test phase the goal of our model is to classify unannotated tweets with the categories learned during the training phase. In the following section we describe the main features used in our SVM classifier.

2.2 Features

For the main task A (hate speech detection), we distinguish three groups of features[2]:

- *Bag-of-n-grams*: Bag-of-n-grams features, which have been already used for hate speech detection (Chatzakou et al., 2017), are often reported to be highly predictive and can be combined with other features to improve performance. We make use of unigrams, bigrams and trigrams, represented in the feature vectors by their frequency in a tweet.

- *Sentiment analysis*: Hate speech and sentiment analysis are closely related, and we can assume that negative sentiment usually pertains to a hate speech message (Schmidt and Wiegand, 2017). To integrate this feature into our model, we simply add the output of a pre-trained sentiment analysis classifier.

- *Word embeddings*: Word embeddings are low-dimensional vector representations of words and are used extensively in natural language processing (Goldberg, 2016). In particular, Bayot and Gonçalves (2016) showed that word embeddings provide a useful generalization signal in text classification when used in a similar setting. In our case, we add the average of the embeddings in a tweet as an additional feature in our SVM classifier.

For task B, we use two simple extra features with specific information about each subtask:

- For *target classification* (individual or group), we use the count of the plural nouns in the tweet as a feature.

- For *aggressive behaviour*, we use the count of the insults in the tweet as a feature. We hypothesize that a high level of insults may involve violent behaviour. To this end, we filter a database from insults collected at https://hatebase.org/.

2.3 Feature selection

One of the main issues in text classification is the high dimensionality of the feature space (Yang and Pedersen, 1997). For instance, over 300K and 150K features were initially obtained using the bag-of-n-grams features alone on, respectively, the English and Spanish training sets from Task A (see Section 3.1). Besides the computational cost to train a model with such a large amount of features, an additional issue is the noise that could be introduced by including many irrelevant features. Thus, it is generally desirable to reduce the feature space, without sacrificing classification accuracy.

The feature selection method used in our system is based on word frequency, understanding a word as an n-gram. The system first delimits the n-grams by a frequency number to significantly reduce the feature space before preparing the vectors for the SVM. Then, highly sparse features (i.e. containing zero in more than 99.9% of the samples) are removed.[3]

[2] We use an extra standard feature to these three groups, the *length of the tweet* in words.

[3] This was achieved by leveraging the *VarianceThreshold* tool from *scikit-learn* (Pedregosa et al., 2011): https://scikit-learn.org/stable/modules/feature_selection.html

3 Evaluation

In this section we describe the experimental setup (Section 3.1) of our system along with the results obtained (Section 3.2), including a brief analysis of errors detected in the evaluation phase (Section 3.3).

3.1 Experimental setup

In the following we present the datasets considered, details about the text preprocessing and feature selection procedures, the pre-trained models used as part of our model, and how parameter tuning was performed.

Datasets. We used the two datasets made available as part of the SemEval-2019 Task 5 competition: one for English and another for Spanish. The datasets consist of training, development and test splits. For English, the number of tweets for each split is 9100, 1000 and 2971 for training, development and test, respectively. Conversely, the Spanish splits contain 4600, 500 and 1600 tweets.

Preprocessing. Each tweet is tokenized using the *spaCy* NLP library[4]. We experimented with various preprocessing variants and decided to work with raw words as tokens (i.e., without applying lemmatization), removing punctuation and URLs but keeping emojis and stopwords (pronouns and articles can be relevant in the context of hate speech classification).

Feature selection. As explained in Section 2.3, a feature selection procedure is applied on the n-gram features to reduce their noise and size. After the feature selection is performed for the bag-of-n-grams features, the featured space is reduced from 336,669 to 4,605 in English task A, and from 177,003 to 4,217 in Spanish task A.

Pre-trained models. Regarding sentiment analysis, we used as features the polarity [-1.0, 1.0] and the subjectivity [0.0, 1.0] of a tweet according to *TextBlob*[5] (Loria et al., 2014). Note that Textblob is only optimized for English input and was not used for the Spanish tasks. We leave the exploration of Spanish sentiment analysis systems for future work.

As far as word embeddings are concerned, we made use of Spanish and English 100-dimensional FastText word embeddings (Bojanowski et al., 2017) trained on two large Twitter corpus from Spain and United States, respectively (Barbieri et al., 2016).

Parameter tuning. We experimented with several kernels and parameter configurations to train the Support Vector Machines, including polynomial and linear kernels. Since our system is trained with a large amount of features, it is hard to find an optimal parameter configuration for the polynomial kernel. Therefore, we decided to use a linear kernel, as the SVM training was faster and implied tuning less parameters. We fine-tuned the C parameter of the SVM using as validation the development set of the task. This parameter tuning was performed using bag-of-n-grams as features and on the Spanish dataset only. The value of C that achieved the highest accuracy in the development set was $C = 2^{-5}$ for Task A and Task B-target classification, and $C = 3$ for Task B-aggressive behaviour, which were fixed across all experiments.

3.2 Results

In the following we present our results for Task A (Section 3.2.1) and Task B (Section 3.2.2).

3.2.1 Task A

Task A consists of detecting hate speech (HS) against women or immigrants in the text. Systems were evaluated according to standard classification metrics such as accuracy and macro-F1 score.

Table 1 shows our Task A results in the development and evaluation sets comparing different sets of features described in Section 2.2. As can be observed in the table, the highest accuracy and macro-F1-score obtained in the development phase were, respectively, 78.4 and 77.9 using all features (i.e., n-grams, tweet length and word embeddings for Spanish) and 72.8 and 72.0 with n-grams for English (the same features including sentiment analysis in this case). The sentiment analysis feature provided a small improvement when combined with n-grams on the English development set, but had a negligible influence on the test set. In general, except for the word embeddings which seem to generalize better, all features performed close to a random baseline in English. A further analysis should be required to explain

[4] https://spacy.io/

[5] TextBlob is a public Python library for processing textual data that provides an API for common NLP tasks such as sentiment analysis: https://textblob.readthedocs.io/en/dev/index.html

Features	English				Spanish			
	Dev		Test		Dev		Test	
	Acc	F1	Acc	F1	Acc	F1	Acc	F1
All	**72.8**	**72.0**	50.1	48.1	**78.4**	**77.9**	**73.1**	**72.2**
N-grams	72.1	71.5	50.0	48.0	77.2	76.6	73.0	71.9
N-grams and sent. analysis∗	72.5	71.8	50.1	48.0	77.4	76.8	73.0	71.9
Word embeddings	65.3	60.1	**57.5**	**56.9**	63.6	56.3	65.9	55.0
SVC baseline	-	-	*49.2*	*45.1*	-	-	*70.5*	*70.1*
MFC baseline	-	-	*57.9*	*36.7*	-	-	*58.8*	*37.0*

Table 1: Task A results using different sets of features. The row marked with ∗ was submitted to the task.

the difference between development and test results, which affected most participating systems. Some possible explanations are discussed in the Analysis section (Section 3).

Unlike in English, in Spanish our system obtains the best result with the configuration that performed best in the development set. Our official submission (n-grams and tweet length as features) ranked sixth in terms of macro-F1 and second in terms of accuracy among all 40 participating systems. In the English task, with the addition of word embeddings as feature, our system would have ranked third in terms of macro-F1.

3.2.2 Task B

Task B consists of identifying the target harassed as individual or generic (TR), and to classify hateful tweets as aggressive or not aggressive (AG). In addition to the individual macro-F1 scores for these two subtasks (i.e., TR and AG), two global scores based on the average macro-F1 scores and Exact Match Ratio (EMR) (Kazawa et al., 2005) are reported. The EMR score measures the percentage of instances which are correctly labeled in all subtasks, i.e., HS (hate speech), TR (target) and AG (aggressiveness). As previously explained, our official submission consisted of n-grams and sentiment analysis features, with the addition of the two extra features mentioned in Section 2.2: a count of plurals in each tweet for TR and a count of insults for AG.

Table 2 displays the results of our system on Task B. As can be observed, results for TR are better for Spanish than English, which could be attributed to the fact that Spanish uses more plural forms than English. Regarding AG, the reason could be that the insult database was more accurately filtered for Spanish than English. These results, however, show the general trend of participating systems in the task.

Finally, we noted that training only on the portion of tweets where hate speech is present is beneficial. In our official submission we used all tweets for training, irrespective of whether they were hateful or not. Using all tweets for training was clearly adding a lot of noise to the training, and without it, a significant increase in the performance was obtained. Table 3 shows the results using the full training set and the training set including tweets considered as hateful. As an example, in the Spanish test set, the macro-F1-scores using only hateful tweets for training were 92.8 and 87.8, which means an absolute improvement of 16.9 and 14.3 percentage points for target classification and aggressive behaviour, respectively.

3.3 Analysis

When analyzing the errors of our system, we found a number of cases where irony was present. It is worth noting that sometimes hate speech is expressed through irony, and therefore does not imply an aggressive behaviour. Moreover, offensive language does not necessarily imply hate speech, which poses an additional challenge to these systems. Here are some sample tweets of hate speech without aggressive behaviour from the development set:

> "Say it loud, say it clear, illegal #immigrants are not welcome here."

> "Poland: our country is safe because we haven't taken in refugees"

Finally, given the disparity of results between English development and test sets, we analyzed possible causes for this behaviour. In Task A, we obtained the best performance by only using word embeddings. One of the reasons for these results

	F1(HS)	F1(TR)	F1(AG)	F1(avg)	EMR
English Dev	71.8	72.7	60.9	68.5	56.9
English Test	48.0	68.2	54.4	56.8	31.2
Spanish Dev	77.9	80.6	81.6	80.0	68.4
Spanish Test	72.2	75.9	73.5	73.9	62.9

Table 2: Task B results in the development and evaluation phases.

Training	**English Test**		**Spanish Test**	
	F1(TR)	F1(AG)	F1(TR)	F1(AG)
Full	68.2	54.4	75.9	73.5
Only hateful	**88.0**	**70.9**	**92.8**	**87.8**

Table 3: Macro-F1 results in Task B by using different types of training data.

could be that, in the development set 64.8% of the vocabulary of the test set was present in the training set, whereas in the test set only 54.8% of the vocabulary overlapped with the vocabulary of the training set. This reduction in the overlapping vocabulary between training and test handicaps the performance of n-gram based systems, which heavily relies in vocabulary overlap. Word embeddings are less affected by this condition since they can capture synonymy relations and therefore are able to generalize better. This could explain why using word embeddings alone attained the best performance in this experiment, as the n-grams were not helpful.

4 Conclusion and future work

In this paper we described our system presented at SemEval 2019 Task 5. The system follows a traditional machine learning approach based on feature engineering, making use of n-grams, sentiment analysis and word embeddings as its main features. The results obtained show how word embeddings, when combined with n-grams and sentiment analysis, can improve the performance of the system. In Spanish task A, our proposed system obtained a remarkable macro-F1 score of 72.5 (sixth highest) and an accuracy of 73.6 (second highest). In view of these results, we have achieved our objective of building a strong baseline for hate speech detection.

Future directions of this work include incorporating users' features to the model, studying how the pronouns and the context of the tweet may affect hate speech classification, and comparing the resulting system with deep neural network approaches, which have recently gained popularity in text classification tasks.

Acknowledgments

We would like to thank Miguel Camacho and the Hate Crime National Office in Spain for their support.

References

Francesco Barbieri, German Kruszewski, Francesco Ronzano, and Horacio Saggion. 2016. How cosmopolitan are emojis?: Exploring emojis usage and meaning over different languages with distributional semantics. In *Proceedings of the 2016 ACM on Multimedia Conference*, pages 531–535. ACM.

Valerio Basile, Cristina Bosco, Elisabetta Fersini, Debora Nozza, Viviana Patti, Francisco Rangel, Paolo Rosso, and Manuela Sanguinetti. 2019. Semeval-2019 task 5: Multilingual detection of hate speech against immigrants and women in twitter. In *Proceedings of the 13th International Workshop on Semantic Evaluation (SemEval-2019)*. Association for Computational Linguistics.

Roy Bayot and Teresa Gonçalves. 2016. Author profiling using svms and word embedding averages. In *Proceedeings of the International Conference on Software, Knowledge, Information Management and Applications (SKIMA)*. CEUR.

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2017. Enriching word vectors with subword information. *Transactions of the Association of Computational Linguistics*, 5(1):135–146.

Despoina Chatzakou, Nicolas Kourtellis, Jeremy Blackburn, Emiliano De Cristofaro, Gianluca Stringhini, and Athena Vakali. 2017. Mean birds: Detecting aggression and bullying on twitter. In *Proceedings of the 2017 ACM on web science conference*, pages 13–22. ACM.

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*.

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. ACM.

Yoav Goldberg. 2016. A primer on neural network models for natural language processing. *Journal of Artificial Intelligence Research*, 57:345–420.

Thorsten Joachims. 1998. Text categorization with support vector machines: Learning with many relevant features. In *European conference on machine learning*, pages 137–142. Springer.

Hideto Kazawa, Tomonori Izumitani, Hirotoshi Taira, and Eisaku Maeda. 2005. Maximal margin labeling for multi-topic text categorization. In *Advances in neural information processing systems*, pages 649–656.

Steven Loria, P Keen, M Honnibal, R Yankovsky, D Karesh, E Dempsey, et al. 2014. Textblob: simplified text processing. *Secondary TextBlob: Simplified Text Processing*.

Vinita Nahar, Sayan Unankard, Xue Li, and Chaoyi Pang. 2012. Sentiment analysis for effective detection of cyber bullying. In *Asia-Pacific Web Conference*, pages 767–774. Springer.

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.

Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, et al. 2011. Scikit-learn: Machine learning in python. *Journal of machine learning research*, 12(Oct):2825–2830.

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*, pages 1–10.

Leandro Silva, Mainack Mondal, Denzil Correa, Fabrício Benevenuto, and Ingmar Weber. 2016. Analyzing the targets of hate in online social media. In *Proceedings of the Tenth International AAAI Conference on Web and Social Media*.

Yiming Yang and Jan O Pedersen. 1997. A comparative study on feature selection in text categorization. In *Proceedings of the International Conference on Machine Learning*, volume 97, page 35.