Grunn2019 at SemEval-2019 Task 5: Shared Task on Multilingual Detection of Hate

Mike Zhang, Roy David, Leon Graumans, Gerben Timmerman
University of Groningen
Groningen, the Netherlands
{j.j.zhang.1, r.a.david,
l.r.n.graumans, g.h.r.timmerman}@student.rug.nl

Abstract

Hate speech occurs more often than ever and polarizes society. To help counter this polarization, SemEval 2019 organizes a shared task called the Multilingual Detection of Hate. The first task (A) is to decide whether a given tweet contains hate against immigrants or women, in a multilingual perspective, for English and Spanish. In the second task (B), the system is also asked to classify the following subtasks: hateful tweets as aggressive or not aggressive, and to identify the target harassed as individual or generic. We evaluate multiple models, and finally combine them in an ensemble setting. This ensemble setting is built of five and three submodels for the English and Spanish task respectively. In the current setup it shows that using a bigger ensemble for English tweets performs mediocre, while a slightly smaller ensemble does work well for detecting hate speech in Spanish tweets. Our results on the test set for English show 0.378 macro F1 on task A and 0.553 macro F1 on task B. For Spanish the results are significantly higher, 0.701 macro F1 on task A and 0.734 macro F1 for task B.

1 Introduction

The increasing popularity of social media platforms such as Twitter for both personal and political communication has seen a well-acknowledged rise in the presence of toxic and abusive speech on these platforms (Kshirsagar et al., 2018). Although the terms of services on these platforms typically forbid hateful and harassing speech, the volume of data requires that ways are found to classify online content automatically. The problem of detecting, and therefore possibly limit the hate speech diffusion, is becoming fundamental (Nobata et al., 2016).

Previous work concerning hate speech against immigrants and women such as Olteanu et al. (2018) observed that extremist violence tends to lead to an increase in online hate speech, particularly on messages directly advocating violence. Also, Anzovino et al. (2018) contributed to the research field by (1) making a corpus of misogynous tweets, labelled from different perspective and (2) created an exploratory investigations on NLP features and ML models for detecting and classifying misogynistic language.

Basile et al. (2019) proposed a shared task on the Multilingual Detection of Hate, where participants have to detect hate speech against immigrants and women in Twitter, in a multilingual perspective, for English and Spanish. The task is divided in two related subtasks for both languages: a basic task about hate speech, and another one where fine-grained features of hateful contents will be investigated in order to understand how existing approaches may deal with the identification of especially dangerous forms of hate, for example those where the incitement is against an individual rather than against a group of people, and where an aggressive behavior of the author can be identified as a prominent feature of the expression of hate.

Within this experiment, Task A is a binary classification task where our system has to predict whether a tweet is hateful or not hateful. For Task B, our system has to decide whether a tweet is aggressive or not aggressive, and whether that tweet targets an individual or generic group, to elaborate, a single human or group of people.

The paper is structures as follows. In section 2 our system setup is described. In section 3, the datasets together with the preprocessing steps are presented. In section 4, obtained results are detailed. Finally, in section 5 a discussion about the proposed system is outlined.

Proceedings of the 13th International Workshop on Semantic Evaluation (SemEval-2019), pages 391–395
Minneapolis, Minnesota, USA, June 6–7, 2019. ©2019 Association for Computational Linguistics

2 System Setup

In our approach, we trained multiple classifiers and combined their results into an ensemble model using majority vote.

English Ensemble Setup

The setup of our system optimized for the English classification tasks consisted of the following classifiers:

- Random Forest
- Support Vector Machine (1)
- Support Vector Machine (2)
- Logistic Regression
- BiLSTM

Spanish Ensemble Setup

Due to time restrictions, we used three classifiers for the Spanish tasks:

- Random Forest
- Support Vector Machine (1)
- Logistic Regression

These time restrictions occurred, because we decided in the last moment to run our system for the Spanish task too. However, we did not have hate speech specific word embeddings, nor trained a BiLSTM model for the Spanish task. Therefore, we decided to run only three classifiers for both the Spanish tasks.

2.1 Random Forest (RF)

For our RF model we executed a grid search starting with the following parameters: character n-grams with range: 2-3, 2-4, 1-3, 1-4; word n-grams with range 1, 1-2, 1-3, 1-4 and all combinations of them. In the end we used a tf-idf vectorizer with character n-grams with range 2-4. As for our parameters also following a grid search we used 400 estimators, entropy as our split criterion/estimator, balanced for our class weight and a random seed of 1337. Due to time restrictions, we used the same parameters for the Spanish tasks.

2.2 Support Vector Machine (SVM 1)

Within this subpart of our ensemble model, we used a SVM model from the scikit-learn library (Pedregosa et al., 2011). We used a *linear* `kernel`, and a *weighted* `class_weight`. This model used vectorized character n-grams in range 2-4 using a tf-idf vectorizer as its input.

2.3 Support Vector Machine (SVM 2)

We used a second SVM classifier within our ensemble model, but this time with word embeddings as its input. This choice is motivated by the hypothesis that introducing different predictions given by models trained differently could lead to more insights. We tested four pre-trained embedding representations, which are the following: the 300-dimensional `GloVe` embeddings and the 25-dimensional `GloVe Twitter` representations by Pennington et al. (2014); a 400-dimensional and 100-dimensional word embedding created from tweets (Van der Goot). Using the `GloVe` embeddings proved to be superior within our work. The results of each word embedding can be found in Table 6.

2.4 Logistic Regression (LR)

Following our grid search testing our LR model with a tf-idf vectorizer with the following parameters: character n-grams with range: 2-3, 2-4, 1-3, 1-4; word n-grams with range 1, 1-2, 1-3, 1-4 and all combinations of them, we got the best performance using a tf-idf vectorizer with character n-grams with range 2-4. Due to time restrictions, we used the same parameters for the Spanish tasks.

2.5 BiLSTM

Our BiLSTM classifier was only optimized for the English classification task. Hence we decided not to use it in our Spanish setup. In combination with the BiLSTM model we used an attention mechanism, as proposed by Yang et al. (2016).

LSTM models can handle input sequentially and therefore can take word order into account. We combine this with a bidirectional model, which allows us to process the tweets both forwards and backwards. For each word in the tweets, the LSTM model combines its previous hidden state and the current word's embedding weight to compute a new hidden state. After using dropout to shut down a percentage of neurons of the model, we feed the information to the at-

	Hate Speech		Target Range		Aggressiveness	
	0	1	0	1	0	1
Trial data	50	50	87	13	80	20
Train data	5217	3783	7659	1341	7440	1559
Dev data	573	427	781	219	796	204
Test data	3000					
Total	13100					

Table 1: Distribution of English data, labels of the test data were not specified.

	Hate Speech		Target Range		Aggressiveness	
	0	1	0	1	0	1
Train data	2643	1857	3371	1129	2998	1502
Dev data	278	222	363	137	324	176
Test data	1600					
Total	6600					

Table 2: Distribution of Spanish data. No trial data was available, and test data labels were not specified.

tention mechanism. This mechanism emphasizes the most informative words in the article and gives these more weight.

Our final model uses 512 units in the hidden layer of the BiLSTM, a batch size of 64, the Adam optimizer in combination with the default learning rate of 0.001 and a dropout of 0.4. We trained our model for 10 epochs, of which we saved the model with the lowest validation loss.

3 Data and Preprocessing

For this shared task, the data distribution is seen in Table 1 and Table 2 for the train and development data, we assumed the trial data to be train data too. After release of the test data, the distribution would be 69% train, 8% development, and 23% (3000 sentences) test data for the English task, and for the Spanish task 68% train, 8% development, and 24% (1600 sentences) test data. For final submission, we combined the train and development data to train our system on.

The meaning of the binary encoding is as follows, for Hate Speech (HS) and Aggressiveness (AG): 0 or 1, absent and present respectively. For Target Range (TR): 0 or 1, whole group and individual respectively. We notice that there is more data available for the English task than the Spanish one.

With regard to preprocessing, we did this in the following fashion:

- Tokenized with the NLTK TweetTokenizer.
- Replaced URLs with a placeholder suitable for our English embeddings.
- Replaced mentions with a placeholder suitable for the available English embeddings (van der Goot and van Noord, 2017).
- Converted words to lowercase.
- Filtered out stopwords using the stopwords from NLTK, either English or Spanish.
- Removing single characters, excluding emoji's.

For the BiLSTM, we did not do any preprocessing. We deemed this might affect the learning curve of the system, since a BiLSTM algorithm often performs well with lots of different data. So, without preprocessing there will be less loss of information and thus a better performing system.

We tested how the preprocessing affected our scores, results are in Table 3 and Table 4. We used the train and development data available to test the preprocessing. We started using all the preprocessing, and in a cumulative way, excluded a preprocess step one by one. So in the end, we would only have tokenization left.

Interesting is that the scores of the RF and SVM 1 model are higher, for both English and Spanish data, when we exclude preprocessing steps. At the step of replacing URLs and usernames with placeholders, we expected the scores to be higher if excluded. Because if the same URL or username occurs often in the training set, and that specific URL or username is always corresponding with a hateful or non-hateful message, our system could wrongly classify a comment in the development set containing that same URL or username. The scores also increase when we exclude lowercasing or remove single characters in addition to the placeholder steps. However, if we omit either lowercasing or characters alone, the scores do not get better than if we use all preprocessing. This also explains the higher score with the LR model, but if we only disregard the character preprocessing step, the score also does not get better.

4 Results

In this section, we state our results on the test set, as well as the results of our ensemble model and individual models on the development set. Our final system for the English task consists of all five models shown in the English Ensemble Setup, each given a result being either 0 or 1, and run a majority vote on it for a final result. For the Span-

	RF	SVM 1	SVM 2	LR	BiLSTM
All	74.2	73.9	72.7	74.0	-
- URL	74.5	73.8	68.4	73.9	-
- USERNAME	75.3	74.1	68.3	73.9	-
- Lowercase	75.8	73.8	69.2	73.5	-
- Stopwords	74.2	72.4	67.9	73.4	-
- Characters	75.6	73.0	68.9	75.2	-
No preprocess (only tokenization)	75.6	73.0	68.9	75.2	77.5

Table 3: Scores with changes in preprocessing for English, scores in bold means that it was higher than using all preprocessing of the respective system.

	RF	SVM 1	LR
All	78.2	79.9	77.8
- URL	78.7	80.1	77.4
- USERNAME	78.2	78.8	77.9
- Lowercase	75.9	79.6	76.6
- Stopwords	77.3	80.8	76.7
- Characters	75.7	81.0	77.7
No preprocess (only tokenization)	75.7	81.0	77.7

Table 4: Scores with changes in preprocessing for Spanish, csores in bold means that it was higher than using all preprocessing of the respective system.

ish task, the final system contains three models, described in the Spanish Ensemble Setup.

The results on the the various tasks we participated in are listed in Table 5. For the English task, we achieved a much lower accuracy and macro f1-score than for the Spanish task. Assuming the data has been distributed fairly for both languages, it could be that the quality of the test data is lower than the train and development data.

These scores were lower in comparison to our results of the individual classifiers on the development set which are listed in table 3 and 4.

5 Discussion

We compared multiple classification algorithms and combined them into an ensemble model to get a more robust and accurate system. Initially, our system performed reasonably well on the development set, but when tested on the final test set our performance dropped a fair bit. Overall, the drop in performance was to be expected. During the final evaluation of the test set our system predicted over 80% as hate speech. Looking at the data we thought a large part of the remaining 20% could also be classified as hate speech. Also the majority class baseline (Basile et al., 2019) ranked second for accuracy, supporting our expectations.

From our results we can conclude that using a bigger ensemble model for the English tweets performs mediocre in comparison to a smaller ensemble model for detecting hate speech in Spanish tweets.

In the future, we would like to try to improve the performance of our Spanish model, of which our development was cut short due to time restrictions. We would also like to test our models with more high quality data. It would be interesting to find out whether this helps to improve our models' performance.

English Task A	accuracy	macro F1
Fermi	0.653	0.651
Panaetius	0.572	0.571
YNU_DYX	0.560	0.546
Grunn2019	0.459	0.378
English Task B	EMR	macro F1
MFC baseline	0.580	0.421
ninab	0.570	0.467
CIC-1	0.568	0.551
Grunn2019	0.279	0.553
Spanish Task A	accuracy	macro F1
Atalaya	0.731	0.730
mineriaUNAM	0.734	0.730
MITRE	0.729	0.729
Grunn2019	0.708	0.701
Spanish Task B	EMR	macro F1
hammad.fahim57	0.705	0.755
CIC-1	0.675	0.649
gertner	0.671	0.772
Grunn2019	0.601	0.734

Table 5: Scores of our ensemble models on both subtasks and languages during testing phase, compared to the top three systems in that subtask.

	accuracy	macro f1-socre
GloVe 300d	0.726	0.727
Glove Twitter 25d	0.680	0.675
Twitter 100d	0.716	0.711
Twitter 400d	0.719	0.717

Table 6: Scores on the English development set of the Support Vector Machine (SVM 2) classifier using different word embeddings as input.

6 Acknowledgements

We would like to thank the organizers of SemEval-2019[1], the organizers of SemEval 2019 Task 5 - Shared Task on Multilingual Detection of Hate[2], and the reviewers for accepting our paper. We would also like to thank R.M. Van der Goot for his guidance and help where needed.

[1] `http://alt.qcri.org/semeval2019/`

[2] `https://competitions.codalab.org/competitions/19935`

References

Maria Anzovino, Elisabetta Fersini, and Paolo Rosso. 2018. Automatic identification and classification of misogynistic language on twitter. In *International Conference on Applications of Natural Language to Information Systems*, pages 57–64. Springer.

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.

Rob Van der Goot. http://www.let.rug.nl/rob/.

Rob van der Goot and Gertjan van Noord. 2017. MoNoise: Modeling noise using a modular normalization system. *Computational Linguistics in the Netherlands Journal*, 7:129–144.

Rohan Kshirsagar, Tyus Cukuvac, Kathleen McKeown, and Susan McGregor. 2018. Predictive embeddings for hate speech detection on twitter. *arXiv preprint arXiv:1809.10644*.

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.

Alexandra Olteanu, Carlos Castillo, Jeremy Boy, and Kush R Varshney. 2018. The effect of extremist violence on hateful speech online. In *Twelfth International AAAI Conference on Web and Social Media*.

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.

Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. Glove: Global vectors for word representation. In *Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543.

Zichao Yang, Diyi Yang, Chris Dyer, Xiaodong He, Alex Smola, and Eduard Hovy. 2016. Hierarchical attention networks for document classification. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1480–1489.