Atalaya at SemEval 2019 Task 5: Robust Embeddings for Tweet Classification

Juan Manuel Pérez
Universidad de Buenos Aires
CONICET
jmperez@dc.uba.ar

Franco M. Luque
Universidad Nacional de Córdoba
CONICET
francolq@famaf.unc.edu.ar

Abstract

In this article, we describe our participation in HatEval, a shared task aimed at the detection of hate speech against immigrants and women. We focused on Spanish subtasks, building from our previous experiences on sentiment analysis in this language. We trained linear classifiers and Recurrent Neural Networks, using classic features, such as bag-of-words, bag-of-characters, and word embeddings, and also with recent techniques such as contextualized word representations. In particular, we trained robust task-oriented subword-aware embeddings and computed tweet representations using a weighted-averaging strategy. In the final evaluation, our systems showed competitive results for both Spanish subtasks ES-A and ES-B, achieving the first and fourth places respectively.

1 Introduction

Hate speech against women, immigrants, and many other groups is a pervasive phenomenon on the Internet. On the early days of the World Wide Web, many academics adventured that prejudices and hatred would be removed in this space by the dissolution of identities (Lévy, 2001; Rheingold, 1993). Twenty years after this hypothesis, we can say that it has not been the case. The prevalence of racism in the "World White Web" has been studied in a number of works (Adams and Roscigno, 2005; Kettrey and Laster, 2014) and so has been the misogyny in the virtual world (Filipovic, 2007; Mantilla, 2013).

Racist and sexist discourse are a constant in social media, but peaks are documented after "trigger" events, such as murders with religious or political reasons (Burnap and Williams, 2015). Most social media companies are concerned about this issue and take actions against it; nonetheless, most of the efforts still need human intervention, making this task very expensive. Therefore, reducing human intervention is vital in order to have effective tools to avoid the escalation of hate speech.

HatEval (Basile et al., 2019) is a SemEval-2019 shared task aimed at the detection of hate speech towards immigrants and women in tweets. It comprises two subtasks, with datasets in English (EN) and Spanish (ES) for both of them, giving a total of four subtasks. Subtask A is the binary classification of tweets into hateful or not hateful (HS). Subtask B is a triple binary classification task where, in addition to HS, tweets are classified into aggressive or not aggressive (AG), and targets of hate speech are classified into single humans or groups of persons (TR).

In this article, we present our participation in HatEval as team Atalaya. We focused our efforts on subtask A for Spanish (ES-A) but also worked at subtask B in Spanish (ES-B) and subtask A in English (EN-A). Our systems are based on our participation in the polarity classification task of Spanish tweets TASS 2018 (*Sentiment Analysis at SEPLN*) (Martínez-Cámara et al., 2018; Luque and Pérez, 2018).

To represent tweets, we experimented with a mixed approach of bag-of-words, bag-of-characters and tweet embeddings, which were calculated from word vectors using different averaging schemes. We used *fastText* (Bojanowski et al., 2016) to get subword-aware representations specifically trained for sentiment analysis tasks.

These word representations are robust to noise since they can be computed for unseen words by using subword embeddings. Moreover, we trained them using a database of 90M tweets from various Spanish-speaking countries, giving wide domain-specific vocabulary coverage. We achieved additional robustness by doing preprocessing using several text-normalization and noise-reduction techniques.

Also, we experimented with *ELMo* (Peters

Proceedings of the 13th International Workshop on Semantic Evaluation (SemEval-2019), pages 64–69
Minneapolis, Minnesota, USA, June 6–7, 2019.

et al., 2018), a deep contextualized word representation that has drawn a lot of attention in the last months. Unlike *fastText*, *ELMo* returns context-dependent embeddings from a multi-layer bidirectional-LSTM language model. These representations improved the state-of-the-art of several NLP tasks.

For the neural approach, we used bidirectional LSTMs to combine the word embeddings. We also did experiments that mix sequential models with complementary representations such as bag-of-words.

The rest of the paper is as follows. Next Section presents the primary tools we used to build our systems. Section 3 presents the configuration and development of both linear and neural models. Section 4 briefly shows our results in the competition, and Section 5 concludes the work with some observations about our experience.

1.1 Previous Work

The detection of hate speech is a sentence classification task quite related to sentiment analysis and has been studied for several social media networks (Thelwall, 2008; Pak and Paroubek, 2010; Saleem et al., 2017). Regarding the detection of hateful content, Greevy and Smeaton (2004) used bag-of-words and SVMs to detect racist content in web pages. Following a similar approach, Warner and Hirschberg (2012) used unigrams and Brown clusters with SVMs to detect anti-semitic messages on Twitter.

Waseem and Hovy (2016) annotated a corpus and used character n-grams to detect hateful comments, and Badjatiya et al. (2017) used the same dataset to train deep learning models and fine-tuned embeddings along with Gradient Boosted Trees. Zhang et al. (2018) trained a deep neural network combining CNNs with Gated-recurrent units (Cho et al., 2014), outperforming previous systems in several datasets.

Anzovino et al. (2018) collected a corpus of misogynous tweets and proposed a taxonomy to distinguish them into different categories. The authors proposed a number of different techniques to classify them, showing that simple approaches (as using linear models along with token n-grams) achieve competitive performance on small-sized datasets.

Regarding shared tasks, Fersini et al. (2018a) presented a challenge on misogyny detection on Twitter –both in Spanish and English– whereas Fersini et al. (2018b) posed a similar challenge but in Italian and English. Bosco et al. (2018) proposed an automatic detection contest over Twitter posts and Facebook comments, comprising general hate speech.

2 Techniques and Resources

2.1 Preprocessing

Preprocessing is crucial in NLP applications, especially when working with noisy user-generated data. Here, we followed Luque and Pérez (2018), defining two levels of preprocessing: basic and sentiment-oriented preprocessing. We used one or the other, depending on the configuration.

Basic tweet preprocessing includes tokenization, replacement of handles, URLs, and e-mails, and shortening of repeated letters.

Sentiment-oriented preprocessing includes lowercasing, removal of punctuation, stopword, and numbers, lemmatization –using TreeTagger (Schmid, 1995)– and negation handling. For negation handling, we followed a simple approach: We find negation words and add the prefix `'NOT_'` to the following tokens. Up to three tokens are negated, or less if a non-word token is found.

2.2 Bags of Words and Characters

The simplest approach considered to build tweet representations was bag-of-words encoding. A bag-of-words (BoW) builds feature vectors for each token seen in training data. For a particular tweet, its BoW vector contains the number of occurrences of each token on it, resulting in high-dimensional and sparse vectors. Variations of BoW include counting not only single tokens but also n-grams of tokens, binarizing counts, and limiting the number of features.

Character usage in tweets may also hold useful information for sentiment analysis. Character n-grams –such as the presence and repetition of uppercase letters, emoticons, and exclamation marks– may indicate a strong presence of sentiment of some kind, where others may indicate a more formal writing style, and therefore an absence of sentiment.

To capture this information, we considered a bag-of-characters (BoC) representation that encodes counts of character n-grams for some values of n. These vectors are computed from original

texts of tweets, with no preprocessing at all. BoCs have the same variants and parameters as BoWs.

2.3 Word Embeddings

We used *fastText*, a subword-aware embeddings library (Bojanowski et al., 2016) to get context-independent word representations. Instead of using publicly available pre-trained vectors, we trained our own embeddings on a dataset of $\sim$ 90 million tweets from various Spanish-speaking countries. We prepared two versions of the data: one using only basic preprocessing, and the other using sentiment-oriented preprocessing (with the exception of excepting lemmatization). For these two datasets, skip-gram embeddings were trained using different parameter configurations, including a number of dimensions, size of word and subword n-grams, and size of context window.

2.4 Tweet Embeddings

Linear combinations were used to compute a representation for a single tweet. We followed two simple approaches: plain average and weighted average. In the second case, we used a scheme that resembles Smooth Inverse Frequency (SIF) (Arora et al., 2017), inspired by TF-IDF reweighting. Each word w is weighted with $\frac{a}{a+p(w)}$, where $p(w)$ is the word unigram probability, and a is a smoothing hyper-parameter. Big values of a mean more smoothing towards plain averaging.

2.5 Context-Dependent Embeddings

After the great leap forward that represented context-independent word embeddings, a new wave came in the last years. Instead of having vectors trained for each word, context-dependent representations are generated for each token given a sentence. For instance, McCann et al. (2017) used a deep LSTM encoder for Machine Translation to generate context-aware vectors.

ELMo (Peters et al., 2018) is one of these context-dependent approaches and is based on a deep bidirectional language model (biLM). The architecture of the language model consists of L layers of bidirectional LSTMs, plus a context-independent token representation. Hence, for each token in a sequence, we get $2L + 1$ vector representations. To obtain a final vector for each token, the authors suggest collapsing the layers into vectors by means of a linear combination.

In this work, we used the implementation and pre-trained models from Che et al. (2018). The Spanish model was trained with $L = 2$ layers and 1024 dimensions, and the linear combination was done using a simple average.

3 Models

In this section, we describe the models we used in the competition.

3.1 Linear Classifiers

The first set of models we trained were simple classifying models implemented with scikit-learn (Pedregosa et al., 2011).

We started from the optimal configuration from Luque and Pérez (2018), that combines bag-of-words (BoW), bag-of-characters (BoC) and tweet embeddings as follows:

- BoW: All unigrams and bigrams of words, with binarized counts and TF-IDF reweighting. For the Spanish training dataset, this encoding gives 53504 sparse features.
- BoC: All n-grams of characters for $n \leq 5$, with binarized counts and TF-IDF reweighting. For the Spanish training dataset, it gives 226156 sparse features.
- Tweet embeddings: Computed from *fastText* sentiment-oriented word vectors of 50 dimensions. Weighted averaging was done as described in Section 2.4, with a smoothing value of $a = 0.1$.

Here, the only parameters specifically optimized using the HatEval development set were the n-gram ranges considered for BoW and BoC.

Using this vectorial representation we trained logistic regressions and linear-kernel SVMs with different hyperparameter values. The best results are shown in the first block of Tab. 1, as LR_0 and SVM_0.

Next, to confirm the relevance of each of the three components, we performed ablation tests for each of them. Results are displayed as SVM_{BoW}, SVM_{BoC} and SVM_{emb} in Tab. 1. Drops in the performance show the relevance of all components, especially for BoW and BoC.

Next, we tried adding tweet representations computed from ELMo vectors. Full tweet vectors were obtained by doing simple un-weighted averaging. PCA was optionally used to reduce the dimension of final vectors. The best results were

Model	Acc	F1 (avg)
LR_0	0.84	0.84
SVM_0	**0.85**	**0.85**
SVM_{BoW}	0.81	0.81
SVM_{BoC}	0.81	0.81
SVM_{emb}	0.84	0.84
SVM_{ELMo}	0.84	0.84

Table 1: Experiments with logistic regressions (LRs) and SVMs on the Spanish development set. Models are described in Section 3.1. The best result is in bold.

obtained using PCA to reduce from the original 1024 to 100 dimensions.

Results are shown as SVM_{ELMo} in Tab. 1. It can be seen that, under this configuration, we are not able to improve our results using ELMo.

To participate in the Spanish subtask B (ES-B) we used a very naive approach. We didn't develop or tune a specific system for this subtask but instead used the same system and configuration that was found optimal for subtask A. To do this, we first mapped the triple classification problem to a 5-way classification problem for all the possible label combinations:

HS	AG	TR
0	0	0
1	0	0
1	0	1
1	1	0
1	1	1

Then, we simply trained the classifier using the Spanish subtask B training dataset.

3.2 Neural Models

The second set of models we trained are neural models. We trained Recurrent Neural Networks (RNNs) using pre-trained context-dependent representations for Spanish.

The first model considered was a bidirectional LSTM with a dense layer on top, consuming ELMo vectors; we call this model *LSTM-ELMo*. Also, we tried another model by adding a second input consisting of a bag-of-words, as illustrated in Figure 1. We call this model *LSTM-ELMo+BoW*. Using *fastText* embeddings (of dimension 300 and context window 5) instead of BoW was considered as suggested by Peters et al. (2018) but discarded as it had no positive impact in performance (in the development dataset).

The *biLSTM* layer consists of 256 units. The bag-of-words has the 3500 most-frequent n-grams (having document-frequency less than 0.65), followed by a 512-unit dense layer. The two last dense layers have 64 neurons.

We used Keras (Chollet et al., 2015) to implement and train our models. *Adam* (Kingma and Ba, 2014) was the chosen optimizer, with $lr = 35 * 10^{-5}$ and $decay = 0.01$. To regularize our models, we applied dropout with keep-prob of 0.2 on the first layer, and 0.45 on the second, and we also early-stopped the training monitoring the performance on the development dataset. The hyperparameters were chosen from a small random search, as training ELMo is computationally expensive.

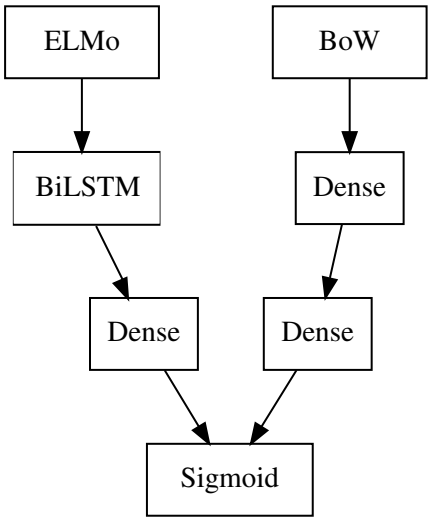

Figure 1: The *LSTM-ELMo+BoW* architecture. ELMo and BoW boxes represent inputs.

4 Results

Table 2 displays the evaluation results for the three classifiers trained for subtask A: SVM_0, and both neural models *LSTM-ELMo* and *LSTM-ELMo+BoW*. For Spanish, the best performing system was SVM_0. Despite its simplicity, it ranked first in terms of average F1 in the official results.

Among the neural models, *LSTM-ELMo+BoW* performed best, and ranked in position 17 for Spanish in terms of average F1.[1] We can observe that *LSTM-ELMo+BoW* performs better on the development set, although its performance decreases sharply in the test set. In spite of the applied regularization, we might have incurred in overfitting during model selection (Cawley and Talbot, 2010) as the chosen model has higher variance

[1] Results shown in Tab. 2 differ from the ones in the leaderboard as we couldn't exactly reproduce the experiments.

	Spanish				English			
	Dev		Test		Dev		Test	
Classifier	Acc	F1 (avg)	Acc	F1 (avg)	Acc	F1 (avg)	Acc	F1 (avg)
SVM_0	**0.850**	**0.850**	0.731	**0.730**	—	—	—	—
LSTM-ELMo	0.820	0.816	**0.732**	0.721	0.705	0.695	**0.508**	**0.471**
LSTM-ELMo+BoW	0.824	0.821	0.719	0.712	**0.743**	**0.738**	0.502	0.461

Table 2: Our evaluation results for subtask A on the development and test sets for Spanish and English. F1 (avg) is the average on positive and negative classes.

than *LSTM-ELMo*. This last model achieved similar results to SVM_0. This difference between the models was not seen in English.

For the Spanish subtask B (ES-B), the same SVM_0 system was used, achieving an average F1 of 0.758 and an EMR score of 0.657 over the test set (fourth place in terms of EMR).

5 Conclusion and future work

As in our previous experience with sentiment analysis, we found that linear models can be a match for neural models. Moreover, this time our SVM ranked in the first place in one of the subtasks.

We believe that –for this kind of challenges with small-sized datasets– preprocessing techniques, data normalization and robustness play a stronger role than model design and hyperparameter tuning. On the other hand, deep neural models are highly expressive and prone to overfitting, requiring being extremely careful with regularization.

Acknowledgments

We are grateful to Pablo Brusco for providing us with helpful comments. This material is based upon work supported by the Air Force Office of Scientific Research under award number FA9550-18-1-0026, and also by a research grant from SeCyT, Universidad Nacional de Córdoba.

References

Josh Adams and Vincent J Roscigno. 2005. White supremacists, oppositional culture and the world wide web. *Social Forces*, 84(2):759–778.

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.

Sanjeev Arora, Yingyu Liang, and Tengyu Ma. 2017. A simple but tough-to-beat baseline for sentence embeddings. In *International Conference on Learning Representations*.

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.

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.

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2016. Enriching word vectors with subword information. *arXiv preprint arXiv:1607.04606*.

Cristina Bosco, Dell'Orletta Felice, Fabio Poletto, Manuela Sanguinetti, and Tesconi Maurizio. 2018. Overview of the evalita 2018 hate speech detection task. In *EVALITA 2018-Sixth Evaluation Campaign of Natural Language Processing and Speech Tools for Italian*, volume 2263, pages 1–9. CEUR.

Pete Burnap and Matthew L Williams. 2015. Cyber hate speech on twitter: An application of machine classification and statistical modeling for policy and decision making. *Policy & Internet*, 7(2):223–242.

Gavin C Cawley and Nicola LC Talbot. 2010. On overfitting in model selection and subsequent selection bias in performance evaluation. *Journal of Machine Learning Research*, 11(Jul):2079–2107.

Wanxiang Che, Yijia Liu, Yuxuan Wang, Bo Zheng, and Ting Liu. 2018. Towards better UD parsing: Deep contextualized word embeddings, ensemble, and treebank concatenation. In *Proceedings of the CoNLL 2018 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*, pages 55–64, Brussels, Belgium. Association for Computational Linguistics.

Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning

phrase representations using rnn encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*.

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

Elisabetta Fersini, Maria Anzovino, and Paolo Rosso. 2018a. Overview of the task on automatic misogyny identification at ibereval. In *Proceedings of the Third Workshop on Evaluation of Human Language Technologies for Iberian Languages (IberEval 2018), co-located with 34th Conference of the Spanish Society for Natural Language Processing (SEPLN 2018). CEUR Workshop Proceedings. CEUR-WS. org, Seville, Spain*.

Elisabetta Fersini, Debora Nozza, and Paolo Rosso. 2018b. Overview of the evalita 2018 task on automatic misogyny identification (ami). *Proceedings of the 6th evaluation campaign of Natural Language Processing and Speech tools for Italian (EVALITA'18), Turin, Italy. CEUR. org*.

Jill Filipovic. 2007. Blogging while female: How internet misogyny parallels real-world harassment. *Yale JL & Feminism*, 19:295.

Edel Greevy and Alan F Smeaton. 2004. Classifying racist texts using a support vector machine. In *Proceedings of the 27th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 468–469. ACM.

Heather Hensman Kettrey and Whitney Nicole Laster. 2014. Staking territory in the "world white web" an exploration of the roles of overt and color-blind racism in maintaining racial boundaries on a popular web site. *Social Currents*, 1(3):257–274.

Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Pierre Lévy. 2001. *Cyberculture*, volume 4. U of Minnesota Press.

Franco M. Luque and Juan Manuel Pérez. 2018. Atalaya at TASS 2018: Sentiment analysis with tweet embeddings and data augmentation. In *Proceedings of TASS 2018: Workshop on Semantic Analysis at SEPLN, TASS@SEPLN 2018, co-located with 34nd SEPLN Conference (SEPLN 2018), Sevilla, Spain, September 18th, 2018.*, pages 29–35.

Karla Mantilla. 2013. Gendertrolling: Misogyny adapts to new media. *Feminist Studies*, 39(2):563–570.

Eugenio Martínez-Cámara, Yudivián Almeida Cruz, Manuel C. Díaz-Galiano, Suilan Estévez Velarde, Miguel Á. García-Cumbreras, Manuel García-Vega, Yoan Gutiérrez Vázquez, Arturo Montejo Ráez, André Montoyo Guijarro, Rafael Muñoz Guillena, Alejandro Piad Morffis, and Julio Villena-Román. 2018. Overview of TASS 2018: Opinions, health and emotions. In *Proceedings of TASS 2018: Workshop on Semantic Analysis at SEPLN (TASS 2018)*, volume 2172 of *CEUR Workshop Proceedings*, Sevilla, Spain. CEUR-WS.

Bryan McCann, James Bradbury, Caiming Xiong, and Richard Socher. 2017. Learned in translation: Contextualized word vectors. In *Advances in Neural Information Processing Systems*, pages 6294–6305.

Alexander Pak and Patrick Paroubek. 2010. Twitter as a corpus for sentiment analysis and opinion mining. In *LREC*, volume 10, pages 1320–1326.

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.

Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. *CoRR*, abs/1802.05365.

Howard Rheingold. 1993. *The virtual community: Finding commection in a computerized world*. Addison-Wesley Longman Publishing Co., Inc.

Haji Mohammad Saleem, Kelly P Dillon, Susan Benesch, and Derek Ruths. 2017. A web of hate: Tackling hateful speech in online social spaces. *arXiv preprint arXiv:1709.10159*.

Helmut Schmid. 1995. Improvements in part-of-speech tagging with an application to german. In *In Proceedings of the ACL SIGDAT-Workshop*, pages 47–50.

Mike Thelwall. 2008. Social networks, gender, and friending: An analysis of myspace member profiles. *Journal of the American Society for Information Science and Technology*, 59(8):1321–1330.

William Warner and Julia Hirschberg. 2012. Detecting hate speech on the world wide web. In *Proceedings of the Second Workshop on Language in Social Media*, pages 19–26. Association for Computational Linguistics.

Zeerak Waseem and Dirk Hovy. 2016. Hateful symbols or hateful people? predictive features for hate speech detection on twitter. In *Proceedings of the NAACL student research workshop*, pages 88–93.

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.