Hate Speech Detection in Portuguese with Naïve Bayes, SVM, MLP and Logistic Regression

Adriano dos S.R. da Silva[1], **Norton T. Roman**[1]

[1]Schoool of Arts, Sciences and Humanities – University of Sao Paulo (EACH-USP)

{adriano.santos.silva,norton}@usp.br

Abstract. *Even though social networks can provide free space for discussing ideas, people can also use them to propagate hate speech and, given the amount of written material in such networks, it becomes necessary to rely on automatic methods for identifying this problem. In this work, we set out to verify the use of some classic Machine Learning algorithms for the task of hate speech detection in tweets written in Portuguese, by testing four different models (SVM, MLP, Logistic Regression and Naïve Bayes) with different configurations. Results show that these algorithms produce better results (in terms of micro-averaged* $F1$ *score) than the LSTM used for benchmark, being also competitive to other results by the related literature.*

1. Introduction

Even though social networks can provide free space for discussing ideas, sometimes people can also use them to propagate hate speech. Defined as "language which attacks or demeans a group based on race, ethnic origin, religion, gender, age, disability, or sexual orientation/gender identity" [Nobata et al. 2016], hate speech is something taken so seriously as to be highlighted in some social network policy terms, such as Twitter's[1], where one reads that "You may not promote violence against or directly attack or threaten other people on the basis of race, ethnicity, national origin, caste, sexual orientation, gender, gender identity, religious affiliation, age, disability, or serious disease".

Due to the vast amount of users and publications in such networks, however, it is virtually impossible for a human agent to analyse all the content that will be published, thereby setting the need for an automated way of identifying hate speech. This need has, in turn, led the community to study the problem, mostly constrained to the English language [Fortuna and Nunes 2018]. Nevertheless, recent years have witnessed a profusion of tasks set up in competitions to detect hate speech in other languages, such as GermEval for German [Wiegand et al. 2018], EVALITA for Italian [Bosco et al. 2018], IberEval [Fersini et al. 2018b] for Spanish and PolEval [Ptaszynski et al. 2019] for Polish.

Hate speech in Portuguese, however, is still little explored and, in this work, we aim to help fill in this gap by comparing some classic machine learning algorithms (namely Support Vector Machines (SVM), Multilayer Perceptron Neural Network (MLP), Logistic Regression (LR) and Naïve Bayes (NB)) to identify hate speech in tweets written in Portuguese, also comparing them to a deep learning architecture from the related

[1]https://help.twitter.com/en/rules-and-policies/hateful-conduct-policy

literature. As it will be made clearer in the forthcoming sections, results show that most of these traditional algorithms produced results which are competitive to those in other languages, also outperforming our benchmark for Portuguese.

The rest of this article is organised as follows. In Section 2, we present some related initiatives for hate speech detection. Section 3, in turn, describes the methodology and tools used in our experiments, whereas Section 4 presents our main results. In Section 5, we discuss and compare our results with those by current research. Finally, in Section 6, we present our final considerations along with some venues for future research.

2. Related Work

Several strategies are currently being adopted to identify hate speech in text messages. In [Basile et al. 2019], for example, an SVM with RBF kernel was used to identify hate speech against immigrants and women in tweets written in English, achieving a macro-averaged $F1$ score of 0.65. This model was the winner at SemEval 2019, in their hate speech Identification task. Another winner, for the Spanish language, was an SVM model with a linear kernel, which delivered a macro-averaged $F1$ score of 0.73.

SVMs were also proposed to identify racism in Twitter messages in English, achieving an F1 score of 0.76 [Hasanuzzaman et al. 2017]. Apart from English, they have also been used to detect hate speech in tweets written in Arabic, with an $F1$ score of 0.82. They were, however, outperformed by a Naïve Bayes model, implemented for the same task, which obtained $F1 = 0.90$ [Mulki et al. 2019]. Logistic Regression was another model to be applied to hate speech identification in English, in this case focusing in hate speech towards women, with a reported accuracy of 0.70 [Saha et al. 2018]. This model has won at EVALITA 2018 competition [Fersini et al. 2018a], which counted with 10 competing teams running many different algorithms.

Finally, regarding hate speech identification in Portuguese, in [Fortuna et al. 2019] the authors apply an LSTM, with pre-trained word embeddings, to detect hate speech in a database of labeled tweets ('hate' vs. 'not hate speech'), obtaining a micro-averaged $F1$ score of 0.72. Since their corpus of labeled tweets is freely available for download, we have chosen it as a test bed for our models, also using their LSTM as the main benchmark for the classifiers tested in this research.

3. Materials and Methods

In this work we relied on a dataset of tweets in Portuguese [Fortuna et al. 2019], collected through Twitter's API from January to March 2017. To build the dataset, tweets were fetched using specific keywords, and then filtered so as to come from user accounts known to produce hate speech material (*i.e.* belonging to the so called "haters"). They were then manually labeled by two annotators (one of them an expert in the field of Social Psychology), reaching $\kappa = 0.72$ (Cohens' Kappa coefficient of agreement [Garmer et al. 2014]). The dataset comprises 5,668 tweets of which 1,228 are classified as hate speech.

During preprocessing, we followed [Fortuna et al. 2019] and removed stop words and punctuation marks using the NLTK (Natural Language Toolkit[2]) library and the python punctuation package, respectively. Text representations were built under the Bag

[2]https://www.nltk.org/

of Words (BOW) [Fan et al. 2008] and N-Gram [Collobert and Weston 2009] paradigms. BOW models were tested at the word level, whereas N-Gram models were tested both at the word and character level, with N ranging from 2 to 5. Within either paradigm, tests were made both using plain word frequencies (*i.e.* with no normalization) and with term frequency - inverse document frequency (tf-idf) [Rajaraman and Ullman 2011] normalization.

The dataset was randomly split in a training/validation set (with 90% of the data) and an out-of-sample test set (with 10% of the data). Models were trained in the training/validation set using 10-fold cross-validation [Han et al. 2011], whereby one further divides this set in 10 different subsets (folds), using 9 of them for training and the last one for validating the model. This procedure is repeated 10 times, with a different fold used for validation each time (while the remaining 9 are used for retraining the models).

Binary classifiers (hate $\times$ no hate) were then developed, trained and validated in the training/validation set, with the above mentioned different text representation strategies, both with and without the preprocessing step being applied to the dataset. Trained classifiers were Naïve Bayes, Logistic Regression, Support Vector Machine and Multilayer Perceptron Neural Network. In the end of the process, the best versions of each classifier (*i.e.* those with the highest mean F1 score, across the 10 folds, in the training/validation set) were run in the test set for out-of-sample comparisons.

4. Results

In what follows, results for each classifier are presented in detail. Mean $F1$ scores, across the 10 folds, are presented for the training/validation set. Out of sample $F1$ scores, measured at the test set, are also presented for all competing models. Even though we base our analysis only on the winning models during validation, along with their respective performance at the test set, we still present the performance of all tested configurations in this set, so as to allow for a better understanding of how winning models depart from their counterparts.

4.1. Naïve Bayes

Even though plain frequencies are discrete, the fact that tf-idf may result in real-valued vectors led us to choose a Gaussian Naïve Bayes model for this comparison. Mean F1 scores, over 10-folds, during training and validation, for every combination of language model (BOW $\times$ N-Gram, $2 \leq N \leq 5$), representation level (word $\times$ character), normalization strategy (none $\times$ tf-idf) and preprocessing (yes $\times$ no) are shown in Table 1, whereas the same results for the final test stage are shown in Table 2.

Best results were observed with BOW, at the word level, and with no normalization or preprocessing ($F1 \approx 0.46$), for the validation set[3] (Table 1). Overall results for this set were seen to vary significantly according to representation level (ANOVA: $F(1, 352) \approx 192.29, p << 0.001$) and language model (ANOVA: $F(4, 352) \approx 89.68, p << 0.001$). Significant differences could not be determined over preprocessing (ANOVA: $F(1, 352) \approx 0.05, p \approx 0.83$) and normalization strategy (ANOVA: $F(1, 352) \approx 1.38, p \approx 0.24$).

[3]From now on, we will refer to the training/validation set as validation set only.

Table 1. F1 result for Naïve Bayes in the validation set

		Without Preprocessing		With Preprocessing	
Level	**Lang. Model**	**No Norm.**	**tf-idf**	**No Norm.**	**tf-idf**
Word	**BOW**	**0.4647**	0.4503	0.4643	0.4494
	2-Gram	0.4530	0.4363	0.4338	0.4309
	3-Gram	0.4002	0.3973	0.3970	0.3971
	4-Gram	0.3786	0.3786	0.3753	0.3753
	5-Gram	0.3725	0.3725	0.3692	0.3692
Char	**2-Gram**	0.3987	0.3995	0.4000	0.4004
	3-Gram	0.3837	0.3809	0.4148	0.4172
	4-Gram	0.3109	0.3073	0.3138	0.3091
	5-Gram	0.3086	0.2954	0.3006	0.2875

Regarding the differences observed across language models, a Tukey multiple comparisons test showed them to be relevant, at $p << 0.001$ for almost all pairwise comparisons. Only differences between 2-Gram and BOW ($p \approx 0.68$) and between 4-Gram and 5-Gram ($p \approx 0.55$) were not found to be statistically significant.

Within the test set (Table 2), the best result was achieved with 2-Gram, at the word level, with no normalization or preprocessing ($F1 \approx 0.48$). Interestingly, the winning configuration during validation was not the same during out-of-sample testing, achieving $F1 \approx 0.46$, almost the same performance[4] as in the validation set. Overall results were also found to vary significantly according to representation level (ANOVA: $F(1, 28) \approx 143.59, p << 0.001$) and language model (ANOVA: $F(4, 28) \approx 63.15, p << 0.001$).

Here too significant differences could not be determined for preprocessing (ANOVA: $F(1, 28) \approx 0.12, p \approx 0.74$) and normalization strategy (ANOVA: $F(1, 28) \approx 0.48, p \approx 0.50$). Differences in pairwise comparisons of language models were found to be relevant ($p << 0.001$) for all pairs but between 2-Gram and BOW ($p \approx 0.1$), 3-Gram and BOW ($p \approx 0.17$), and 4-Gram and 5-Gram ($p \approx 0.38$)

Table 2. F1 result for Naïve Bayes in the test set

		Without Preprocessing		With Preprocessing	
Level	**Lang. Model**	**No Norm.**	**tf-idf**	**No Norm.**	**tf-idf**
Word	**BOW**	0.4560	0.4534	0.4541	0.4534
	2-Gram	**0.4829**	0.4768	0.4750	0.4634
	3-Gram	0.4084	0.4092	0.4076	0.4076
	4-Gram	0.3885	0.3885	0.3810	0.3810
	5-Gram	0.3780	0.3780	0.3787	0.3787
Char	**2-Gram**	0.4253	0.4217	0.4188	0.4154
	3-Gram	0.3768	0.3768	0.4107	0.3891
	4-Gram	0.2874	0.2787	0.2891	0.2891
	5-Gram	0.3209	0.3105	0.3292	0.3292

4.2. Logistic Regression

In this work, Logistic Regression was implemented with L2 regularization and limited memory BFGS optimization [Byrd et al. 1995]. F1 scores in the validation set are shown in Table 3, whereas results for the test set are presented in Table 4. During validation, the

[4]Before rounding.

best result ($F1 \approx 0.69$) was observed with 4-Grams, at the character level representation, without normalization and without preprocessing.

An analysis of the influence of each experimental variable in $F1$ scores revealed that only preprocessing did not seem to have significantly influenced the results (ANOVA: $F(1, 352) \approx 1.04, p = 0.31$). All other variables were found relevant, at $p << 0.001$ (ANOVA: $F(1, 352) \approx 396.22$ for normalization, $F(4, 352) \approx 383.04$ for language model, and $F(1, 352) \approx 2200.30$ for representation level).

Table 3. F1 results for Logistic Regression in the validation set

		Without Preprocessing		With preprocessing	
Level	**Lang. Model**	**No Norm.**	**tf-idf**	**No Norm.**	**tf-idf**
Word	**BOW**	0.6690	0.5193	0.6658	0.5348
	2-Gram	0.4023	0.0716	0.3422	0.0801
	3-Gram	0.1620	0.0037	0.1012	0.0052
	4-Gram	0.1160	0	0.0804	0
	5-Gram	0.0921	0	0.0709	0
Char	**2-Gram**	0.5594	0.4940	0.5432	0.4939
	3-Gram	0.6591	0.5375	0.6417	0.5411
	4-Gram	**0.6889**	0.4779	0.6843	0.4958
	5-Gram	0.6881	0.3892	0.6799	0.4245

Regarding language model, a Tukey multiple comparisons test showed that only differences between 5-Gram and 4-Gram ($p \approx 0.20$) and between 4-Gram and 3-Gram ($p \approx 0.79$) were not found to be statistically significant. All other pairwise differences were found significant at $p << 0.001$. Within the test set, the configuration that produced the highest $F1$ score was that under the 5-Gram model, at the character level, with no preprocessing or normalization ($F1 \approx 0.72$), as shown in Table 4.

Table 4. F1 results for Logistic Regression in the test set

		Without Preprocessing		With Preprocessing	
Level	**Lang. Model**	**No Norm.**	**tf-idf**	**No-Norm.**	**tf-idf**
Word	**BOW**	0.6759	0.5257	0.6729	0.5310
	2-Gram	0.4268	0.0465	0.3289	0.0312
	3-Gram	0.1582	0.0157	0.1044	0.0178
	4-Gram	0.1459	0	0.0757	0
	5-Gram	0.0902	0	0.0610	0
Char	**2-Gram**	0.5898	0.4804	0.5740	0.4719
	3-Gram	0.6772	0.5139	0.6446	0.5303
	4-Gram	0.6929	0.4523	0.6968	0.4588
	5-Gram	**0.7214**	0.3046	0.70	0.3544

Again, an analysis of the influence of each experimental variable in $F1$ scores revealed that only preprocessing did not seem to have significantly influenced the results (ANOVA: $F(1, 28) \approx 0.34, p = 0.57$). All other variables were found relevant at $p << 0.001$. An analysis of language model, across all other variables, showed that only differences between BOW and all N-Gram models ($2 \leq N \leq 5$) were significant ($p << 0.001$) in the test set.

4.3. Support Vector Machine

For each combination of language model (BOW $\times$ N-Gram) and level (word $\times$ character), the best SVM hyper-parameters were determined through grid search [Bergstra and Bengio 2012]. Tests were performed with the RBF, linear, poly and sigmoid kernels, with regularisation values of 0.01, 0.1, 1 and 10. Tables 5 and 6 present the results (in terms of F1 score), both in the validation and test sets, respectively, of the best classifiers for each of the proposed scenarios.

During validation (Table 5), the best performance ($F1 \approx 0.69$) was achieved with 5-Gram, without normalization, with preprocessing, at the character level. An analysis of the influence of each experimental variable in $F1$ scores revealed that only language model (ANOVA: $F(4, 352) \approx 234.07, p << 0.001$) and representation level (ANOVA: $F(1, 352) \approx 1310.24, p << 0.001$) produced statistically relevant differences, whereas normalization (ANOVA: $F(1, 352) \approx 1.66, p \approx 0.20$) was not found significant, with preprocessing being borderline (ANOVA: $F(1, 352) \approx 3.87, p \approx 0.05$).

Table 5. F1 results for SVM in the validation set

		Without Preprocessing		With Preprocessing	
Level	**Lang. Model**	**No Norm.**	**tf-idf**	**No Norm.**	**tf-idf**
Word	**BOW**	0.6798	0.6581	0.6717	0.6713
	2-Gram	0.4924	0.4534	0.4300	0.4029
	3-Gram	0.2430	0.2280	0.1873	0.1836
	4-Gram	0.1606	0.1653	0.1354	0.1416
	5-Gram	0.1378	0.1336	0.1083	0.1068
Char	**2-Gram**	0.6063	0.6031	0.6141	0.5983
	3-Gram	0.6494	0.6379	0.6458	0.6340
	4-Gram	0.6811	0.6712	0.6738	0.6682
	5-Gram	0.6882	0.6659	**0.6946**	0.6728

As for language model, a Tukey test showed non-significant differences between 4-Gram and 3-Gram, 5-Gram and 3-Gram, and 5-Gram and 4-Gram representations, with all other pairwise combination being significant at $p << 0.001$. Differently from the validation set, SVM's best result in the test ($F1 \approx 0.72$) was fount at the word level, with BOW, no normalization or preprocessing (Table 6). Within this set, only representation level ($F(1, 28) \approx 129, 26, p << 0.001$) and language model ($F(4, 28) \approx 23.75, p << 0.001$) were found to significantly correlate with $F1$ scores and, within language model, only differences between BOW and N-Gram ($2 \leq N \leq 5$) were significant ($p << 0.001$). Differences between the remaining variables turned out to be non-significant.

4.4. Multilayer Perceptron Neural Network

As our last model, we built an MLP Neural Network, with three hidden layers containing 8000, 5000 and 2000 neurons each, respectively, and weights updated according to the Adam optimization algorithm [Kingma and Ba 2014]. Neurons were activated by ReLU, and the learning rate was set to 0.001. Tables 7 and 8 present $F1$ results (both in the validation and test sets, respectively) of the best classifiers for each combination of language model, representation level, normalization strategy and preprocessing.

During training and validation (Table 7), the best performance configuration was that under a 5-Gram language model, at the character level, no preprocessing and tf-idf normalization ($F1 \approx 0.66$). Of the four analysed variables, only normalization

Table 6. F1 results for SVM in the test set

		Without Preprocessing		With Preprocessing	
Level	**Lang. Model**	**No Norm.**	**tf-idf**	**No Norm.**	**tf-idf**
Word	**BOW**	**0.7234**	0.6725	0.6986	0.6902
	2-Gram	0.5538	0.4571	0.4615	0.4285
	3-Gram	0.2432	0.1857	0.1985	0.1985
	4-Gram	0.1726	0.1726	0.1594	0.1726
	5-Gram	0.1594	0.1594	0.1323	0.1459
Char	**2-Gram**	0.6311	0.5970	0.6697	0.6169
	3-Gram	0.6842	0.648	0.6981	0.666
	4-Gram	0.6935	0.7017	0.6638	0.6725
	5-Gram	0.7124	0.6933	0.7130	0.6995

did not presented significant influence on $F1$ scores. Influence by the remaining variables was found significant (ANOVA: $F(1, 352) \approx 13.16, p < 0.001$ for preprocessing, $F(1, 352) \approx 740.89, p << 0.001$ for representation level, and $F(4, 352) \approx 132.02, p << 0.001$ for language model).

Table 7. F1 results for the MLP-ANN in the validation set

		Without Preprocessing		With Preprocessing	
Level	**Lang. Model**	**No Norm.**	**tf-idf**	**No Norm.**	**tf-idf**
Word	**BOW**	0.6371	0.6206	0,6282	0.6129
	2-Gram	0.5096	0.5144	0.4927	0.4246
	3-Gram	0.3895	0.3840	0.2526	0.2197
	4-Gram	0.2090	0.2192	0.1490	0,1428
	5-Gram	0.1582	0.1453	0.1174	0.1176
Char	**2-Gram**	0.5624	0.5527	0.5862	0.5577
	3-Gram	0.6217	0.6382	0.6267	0.6253
	4-Gram	0.6528	0.6472	0.6442	0.6192
	5-Gram	0.6376	**0.6621**	0.6410	0.6525

When comparing the results for pairwise combinations of language models, a Tukey multiple comparisons test showed non-significant differences only between 4-Gram and 5-Gram representations ($p \approx 0.71$), with all other combinations showing significant differences ($p < 0.01$). In the out-of-sample test set (Table 8), the best performance was found with 3-Gram, no preprocessing or normalization, at the character level ($F1 \approx 0.71$).

Table 8. F1 results for the MLP-ANN in the test set

		Without Preprocessing		With Preprocessing	
Level	**Lang. Model**	**No Norm.**	**tf-idf**	**No Norm.**	**tf-idf**
Word	**BOW**	0.6602	0.6719	0.6792	0.6637
	2-Gram	0.5586	0.5224	0.4986	0.4684
	3-Gram	0.4226	0.4108	0.2585	0.2112
	4-Gram	0.1971	0.2206	0.2237	0.1726
	5-Gram	0.1971	0.1843	0.1459	0.1594
Char	**2-Gram**	0.6283	0.5462	0.5862	0.5257
	3-Gram	**0.7127**	0.6218	0.6381	0.6093
	4-Gram	0.6431	0.7029	0.6864	0.6371
	5-Gram	0.6920	0.6911	0.6533	0.6801

Within this set, only representation level (ANOVA: $F(1, 28) \approx 61.69, p << 0.001$) and language model ($F(4, 28) \approx 61.69, p << 0.001$) were found to significantly correlate with $F1$ results. Regarding this last variable, a Tukey test showed differences between language models to be significant only between BOW and N-Gram ($2 \leq N \leq 5, p = 0.01$), with other combinations producing non-significant differences.

5. Discussion and Model Comparison

When analysing the influence each variable had in $F1$ scores by the tested models (Tables 9 and 10), we see significant differences[5] regarding representation level and language model (both in validation and test sets). The influence by specific types of language model, however, varied substantially across classifiers. During validation, best models were seen at the character level, with only Naïve Bayes producing a better result at the word level. Over the test set, best results were seen at both representation levels.

Table 9. Variables comparison across classifiers in the validation set

	Variable's Influence		
Classifier	**Significant**	**Non-Significant**	**Best Configuration**
NB	Representation Level Language Model	Preprocessing Normalization	BOW, word level, no normalization, no preprocessing
LR	Representation Level Language Model Normalization	Preprocessing	4-G, char level, no normalization, no preprocessing
SVM	Representation Level Language Model	Preprocessing Normalization	5-G, char level, no normalization, with preprocessing
MLP	Representation Level Language Model Preprocessing	Normalization	5-G, char level, tf-idf, no preprocessing

Table 10. Variables comparison across classifiers in the test set

	Variable's Influence		
Classifier	**Significant**	**Non-Significant**	**Best Configuration**
NB	Representation Level Language Model	Preprocessing Normalization	2-G, word level, no normalization, no preprocessing
LR	Representation Level Language Model Normalization	Preprocessing	5-G, char level, no normalization, no preprocessing
SVM	Representation Level Language Model	Preprocessing Normalization	BOW, word level, no normalization, no preprocessing
MLP	Representation Level Language Model	Normalization Preprocessing	3-G, char level, no normalization, no preprocessing

At the downside, normalization did not seem to have produced significant differences in any classifier, both during validation and test. The only exception to this rule

[5]Even when applying Bonferroni correction.

can be found with Logistic Regression, where we observe a negative effect of tf-idf normalization on $F1$ scores across all other variables, both in the validation and test sets. Similarly, the only observable influence of preprocessing was found with MLP, during validation. For all other models, both at validation and test (including MLP during testing), this variable was not found to have influenced the results.

By comparing the $F1$ results (across all 10 folds) of the winning configuration in each tested model (Table 11), one sees the observed differences to be significant (ANOVA: $F(3, 36) \approx 55.01, p << 0.001$). However, when making a pairwise comparison of classifiers, only differences between Naïve Bayes and its counterparts turn out to be significant (Tukey $p << 0.001$), with all other combinations producing non-significant results. As it seems, our assumption about data normality (implicit in the Gaussian implementation) may have played a role in this model's low performance.

Table 11. Best configuration scores, for all classifiers along the 10 folds

	$F1$ **Score**			
Fold	**SVM**	**LR**	**MLP**	**NB**
1	0.6417	0.6526	0.6063	0.4828
2	0.6557	0.6023	0.5889	0.4574
3	0.6961	0.6762	0.6639	0.5328
4	0.7170	0.7512	0.7330	0.5224
5	0.7263	0.6702	0.6415	0.4309
6	0.7128	0.6952	0.6304	0.4911
7	0.7374	0.7725	0.7196	0.3727
8	0.6919	0.7273	0.6872	0.4409
9	0.6264	0.6358	0.6733	0.4716
10	0.7407	0.7059	0.6770	0.4444

Interestingly, the winning configuration at validation was not the same as during out-of-sample testing for any of the tested models. Moreover, $F1$ scores for these winning configurations increased when they were run in the test set (Table 12), with SVM standing out as the best performance model. This increase was noticed for all models but Naïve Bayes, which performed worse in the test set. Even though these results were nonetheless found to be statistically non-significant (Wilcoxon $W = 6, p \approx 0.69$), they still increase our confidence that models might generalise well over unseen data.

Table 12. $F1$ **scores at validation and testing, for best trained models**

Model	**Validation**	**Test**
NB	0.4647	0.4560
LR	0.6889	0.6929
SVM	0.6946	0.7130
MLP	0.6621	0.6911

Finally, Table 13 presents our results in terms of accuracy, $F1$ score[6] and micro-averaged $F1$ score. These values are furnished to allow for a better comparison to the related literature, since there is still no proper standardisation of metrics for assessing hate speech classifiers. In this table, we present the results, in the test set, of the best ranked models at the validation set.

[6]Which, in our case, also matches macro-averaged $F1$, since we are dealing with two categories only.

Table 13. Alternative measures for the best-ranked models.

Classifier	Accuracy	F1 Score	Micro F1
SVM	0.8836	0.7130	0.8836
LR	0.8765	0.6929	0.8765
MLP	0.8519	0.6911	0.8518
NB	0.7601	0.4560	0.7601

As it turns out, results for our SVM ($F1 = 0.71$) were similar to those by [Basile et al. 2019] for Twitter posts in English and Spanish, which delivered $F1 = 0.65$ and $F1 = 0.73$, with RBF and Linear kernels, respectively. It performed worse, however, when compared to [Hasanuzzaman et al. 2017], which achieved $F1 = 0.76$, and to [Mulki et al. 2019], with $F1 = 0.82$. These, however, work with other languages (English and Arabic, respectively), which might have influenced the results.

Also, instead of trying to identify hate speech in general, as we did in our work, the work by [Basile et al. 2019] and [Hasanuzzaman et al. 2017] focused on hate speech against immigrants and women, and racism, respectively. This narrowing of possible types of hate speech may have reduced variance, and so increased their classifier's score.

Logistic Regression, on the other hand, performed better than its counterpart by [Saha et al. 2018], who report a 70.4% accuracy running on tweets written in English. With 87.7% accuracy, our model outscored theirs by almost 25%. Still, their focus on hate speech against women, instead of hate speech in general, along with the difference in the language of the data, might be responsible for this difference.

At the downside, our Naïve Bayes implementation was outrun by that of [Mulki et al. 2019], who report $F1 \approx 0.90$ in their analysis of tweets in Arabic, when identifying hate speech in general. Being, in our analyses, the only model to significantly differ from its counterparts, this is another indication of this model's non-appropriateness, at least as we have implemented it, to this task. Still, as it might have also been the case for SVM, results could be partially due to differences between Arabic and Portuguese.

Finally, regarding hate speech recognition amongst tweets in Portuguese, our results for the MLP (micro-averaged $F1 = 0.85$) outscored those by [Fortuna et al. 2019], who report micro-averaged $F1 = 0.72$ with LSTM. At this point, it is worth stressing the fact that both models were run in the same corpus, as mentioned in Section 3, thereby reducing the influence of external variables, such as data source and language.

These results by [Fortuna et al. 2019] were actually worse than those by any of the models we tested, Naïve Bayes included, with the largest difference being observed against SVM (which delivered a 22.2% higher micro-averaged $F1$ score than LSTM's). This is somewhat surprising, specially in light of the fact that, with the exception of MLP, all other models are simpler to develop, and (much) faster to train and run. Still, it might be the case that the amount of data (5,668 tweets) may not have been enought for the LSTM to converge.

6. Conclusion

Identifying Hate Speech in tweets is no simple task, since users may try to disguise their comments to prevent algorithms from detecting them. To add complexity to the problem,

data are imbalanced, in the sense that one finds much more comments without the use of hate speech than comments with it. Even though this is good news for everybody, it becomes a problem when designing classifiers to this end.

In this work, we set out to verify the usefulness of some classic Machine Learning algorithms for this task, by testing four different models (NB, LR, SVM and an MLP) with different configurations. Amongst other findings, results showed there to be a significant influence of representation level (word $\times$ character based representations) and language model (BOW $\times$ N-Gram, $2 \leq N \leq 5$) on $F1$ scores, even though specific configurations for these variables varied across models. Other studied variables (normalisation and preprocessing) did not seem to have significantly influenced the results.

Also, significant differences could not be observed between any of our models, except for comparisons with Naïve Bayes, which performed significantly worse than its peers in this research. Most importantly, perhaps, was the fact that all our models were found to be competitive against those by the related literature, sometimes outperforming them, and sometimes being outperformed by them. Differences in the analysed language, along with the type of hate speech to be detected (whether in general, or towards some specific groups), might have influenced these results nonetheless.

When comparing our results to those in the literature for the same task (*i.e.* detecting hate speech in tweets written in Portuguese) and the same corpus, one sees our models outperforming the reported LSTM by a good margin (up to around 22%, with SVM). This, in turn, is an indicative that it is worth looking at more classic models, which usually are much faster to train and run, for this type of task. Still, the use of micro-averaged F1[7], which might not be the best choice for unbalanced data, along with the size of the data set, may have played a role in this result.

Finally, regarding venues for future improvement, we think it is important to analyse the usefulness of other language models, such as word embeddings for example, for the detection of hate speech in Portuguese. Also, there are plenty of other classifiers to be tested, such as Random Forests, Decision Trees, and even other types of Naïve Bayes. These would most certainly help identify which techniques are better suited for this task.

References

Basile, V., Bosco, C., Fersini, E., Nozza, D., Patti, V., Rangel Pardo, F. M., Rosso, P., and Sanguinetti, M. (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*, Minneapolis, USA.

Bergstra, J. and Bengio, Y. (2012). Random search for hyper-parameter optimization. *Journal of machine learning research*, 13:281–305.

Bosco, C., Felice, D., Poletto, F., Sanguinetti, M., and Maurizio, T. (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.

[7] A choice made by [Fortuna et al. 2019] which we followed to allow for a comparison to be made.

Byrd, R. H., Lu, P., Nocedal, J., and Zhu, C. (1995). A limited memory algorithm for bound constrained optimization. *SIAM Journal on scientific computing*, 16(5):1190–1208.

Collobert, R. and Weston, J. (2009). Deep learning in natural language processing. *Tutorial at NIPS*.

Fan, R.-E., Chang, K.-W., Hsieh, C.-J., Wang, X.-R., and Lin, C.-J. (2008). Liblinear: A library for large linear classification. *Journal of machine learning research*, 9:1871–1874.

Fersini, E., Nozza, D., and Rosso, P. (2018a). Overview of the evalita 2018 task on automatic misogyny identification (ami). *EVALITA Evaluation of NLP and Speech Tools for Italian*, 12:59.

Fersini, E., Rosso, P., and Anzovino, M. (2018b). Overview of the task on automatic misogyny identification at ibereval 2018. In *IberEval@ SEPLN*.

Fortuna, P., da Silva, J. R., Wanner, L., Nunes, S., et al. (2019). A hierarchically-labeled portuguese hate speech dataset. In *Proceedings of the Third Workshop on Abusive Language Online*.

Fortuna, P. and Nunes, S. (2018). A survey on automatic detection of hate speech in text. *ACM Computing Surveys (CSUR)*, 51(4):1–30.

Garmer, M., Lemon, J., Fellows, I., and Singh, S. (2014). Various coefficients of interrater reliability and agreement.

Han, J., Pei, J., and Kamber, M. (2011). *Data mining: concepts and techniques*. Elsevier.

Hasanuzzaman, M., Dias, G., and Way, A. (2017). Demographic word embeddings for racism detection on twitter.

Kingma, D. P. and Ba, J. (2014). Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Mulki, H., Haddad, H., Ali, C. B., and Alshabani, H. (2019). L-hsab: A levantine twitter dataset for hate speech and abusive language. In *Proceedings of the Third Workshop on Abusive Language Online*.

Nobata, C., Tetreault, J., Thomas, A., Mehdad, Y., and Chang, Y. (2016). Abusive language detection in online user content. In *Proceedings of the 25th international conference on world wide web*.

Ptaszynski, M., Pieciukiewicz, A., and Dybała, P. (2019). Results of the poleval 2019 shared task 6: First dataset and open shared task for automatic cyberbullying detection in polish twitter. *Proceedings ofthePolEval2019Workshop*.

Rajaraman, A. and Ullman, J. D. (2011). *Mining of massive datasets*. Cambridge.

Saha, P., Mathew, B., Goyal, P., and Mukherjee, A. (2018). Hateminers : Detecting hate speech against women. *CoRR*, abs/1812.06700.

Wiegand, M., Siegel, M., and Ruppenhofer, J. (2018). Overview of the germeval 2018 shared task on the identification of offensive language.