Supervised Classifiers to Identify Hate Speech on English and Spanish Tweets

Sattam Almatarneh[1,2], Pablo Gamallo[1], Francisco J. Ribadas Pena[2], and Alexey Alexeev[3]

[1] Centro Singular de Investigación en Tecnoloxías Intelixentes (CiTIUS), Universidad de Santiago de Compostela, Spain
sattam.almatarneh@usc.es, pablo.gamallo@usc.es
[2] Computer Science Department, University of Vigo
Escola Superior de Enxeñaría Informática, Campus As Lagoas, Ourense 32004, Spain.
[3] ITMO University, Saint-Petersburg, Russia.

Abstract. Consistently with social and political concern about hatred and harassment through social media, in recent years, automatic hate-speech detection and offensive behavior in social media are gaining a lot of attention. In this paper, we examine the performance of several supervised classifiers in the process of identifying hate speech on Twitter. More precisely, we do an empirical study that analyzes the influence of two types of linguistic features (n-grams, word embeddings) when they are used to feed different supervised machine learning classifiers: Support Vector Machine (SVM), Gaussian Naive Bayes (GNB), Complement Naive Bayes (CNB), Decision Tree (DT), Nearest Neighbors (KN), Random Forest (RF) and Neural Network (NN). The experiments we have carried out show that CNB, SVM, and RF are better than the rest classifiers in English and Spanish languages by taking into account all features.

Keywords: Hate speech, Sentiment Analysis, linguistic features, Classification, Supervised Machine Learning

1 Introduction

Hate speech is defined as the language that attacks or diminishes, that incites violence or hate against groups, based on specific characteristics such as physical appearance, religion, descent, national or ethnic origin, sexual orientation, gender identity or other, and it can occur with different linguistic styles, even in subtle forms or when humor is used [10]. Hate speech identification is the sub-field of natural language processing that studies the automatic inference of offensive language and hate speech from textual data. The motivation behind studying this sub-field is to possibly limit the hate speech on user-generated content, particularly, on social media. One popular social media platform for researchers to study is Twitter, a social network website where people "tweet" short posts.

In machine learning, there are two main methods, unsupervised and supervised learning. Unsupervised learning approaches do not depend on the domain

and topic of training data. Supervised learning approaches use labeled training documents based on automatic text classification. A labeled training set with pre-defined categories is required. A classification model is built to predict the document class based on pre-defined categories. The success of supervised learning mainly depends on the choice and extraction of the proper set of features used to identify the target object: hate speech. Even though there are still few approaches to hate speech identification, there are many types of classifiers for sentiment classification, which is the most similar task to hate speech detection.

The main objective of this article is to examine the effectiveness and limitations of supervised classifiers to identify Hate Speech detection in Twitter focused on two specific targets, women and immigrants in two languages: English and Spanish. The main contribution of this paper is to report on a broad set of experiments aimed at evaluating the effectiveness of the most influence linguistic features in a supervised classification task. Besides, we compare some supervised classifiers, namely Support Vector Machine (SVM), Gaussian Naive Bayes (GNB), Complement Naive Bayes (CNB), Decision Tree (DT), Nearest Neighbors (KN), Random Forest (RF) and Neural Network (NN) for our binary classification task.

The rest of the paper is organized as follows. In the following section (2), we introduce some related work. Then, Section 3 describes the method. Experiments are introduced in Section 4, where we also describe the evaluation and discuss the results. We draw the conclusions and future work in Section 5.

2 Related Work

Early studies on hate speech detection focus mainly on lexicon-based approaches [14,12]. As well, some researchers deal with the problem by employing feature (e.g., N-gram, TF-IDF) based supervised learning approach using SVM and Naive-Bayes classifier [11,22].

Bag of words is often reported to be highly predictive and most evident information to employ, unigrams and larger n-grams are included in the feature sets by a majority of text classification task studies such as [7,5]. In many works, n-gram features are combined with a large selection of other features. For example, in their recent work, [18,2] report that while token and character n-gram features are the most predictive single features in their experiments, combining them with all additional features further improves performance. [1] compared different supervised machine learning classifiers: Naive Bayes (NB), Decision Tree (DT), and Support Vector Machine (SVM). The experiments show that SVM clearly outperforms NB and DT on the task of Search for Very Negative Opinions.

Table 1 lists the main components of some published studies: Reference, Year, Features, and Techniques Utilized.

Table 1. The main components of some published studies of supervised learning for Hate speech detection.

Reference	Year	Features	Techniques Utilized
[13]	2004	BOW, N-grams, POS	SVM
[14]	2013	N-grams	NB
[6]	2014	BOW, dictionary, typed dependencies	SVM, NB, DT
[5]	2015	N-gram, typed dependencies	RF, DT, SVM, Bayesian Logistic Regression Ensemble
[21]	2016	Dectionary	SVM
[2]	2019	TF-IDF, N-grams, Word embedding, Lexicon	SVM

3 The Method

To compare different classification algorithms, we build the corresponding classifiers by making use of the same training data in a supervised strategy. The characteristics of tweets are encoded as features in vector representation. These vectors and the corresponding labels feed the classifiers.

3.1 Features

Linguistic features are the most important and influential factor in increasing the efficiency of classifiers for any task of text mining. In this study, we included a number of linguistic features for the task of determining hate speech in tweets. The main linguistic features we will use and analyze are the following: N-grams and word embeddings features.

N-grams Features: we deal with n-grams based on the occurrence of unigrams and bigrams of words in the document. Unigrams (1g) and bigrams (2g) are valuable to detect specific domain-dependent (opinionated) expressions. We assign a weight to all terms by using two representations: Term Frequency-Inverse Document Frequency (TF-IDF) and CountVectorizer.

CountVectorizer transforms the document into token count matrix. First, it tokenizes the document and according to a number of occurrences of each token, a sparse matrix is created. In order to create the matrix, all stopwords are removed from the document collection. Then, the vocabulary is cleaned up by eliminating those terms appearing in less than 4 documents to eliminate those terms that are infrequent.

Doc2Vec: To represent the tweets as dense matrices or embeddings, we make use of the *Doc2Vec* algorithm described in [15]. This neural-based model is efficient when you have to account for high-dimensional and sparse data [15,9]. Doc2vec learns corpus features using an unsupervised strategy and provides a

fixed-length feature vector as output. The output is then fed into a machine learning classifier. We used a freely available implementation of the Doc2Vec algorithm included in *gensim*, [4] which is a free Python library. The implementation of the Doc2Vec algorithm requires the number of features to be returned (length of the vector). Thus, we performed a grid search over the fixed vector length 100 [8,16,17].

4 Experiments

4.1 Dataset

The multilingual detection of hate speech (HatEval) task 5 at SemEval-2019 [3] provides a benchmark dataset. The proposed task focuses on two specific different targets, including immigrants and women in a multilingual perspective, for Spanish and English. The data for the task consists of 9000 tweets in English for training, 1000 for developing and 2805 for the test. For Spanish, 4469 tweets for training, 500 for developing, and 415 for the test. The data are structured in 5 columns: ID, Text, Hate Speech (HS), Target Range (TR) and Aggressiveness (AG). In our study, we consider only the first two columns to identify if the tweet is classified as hate speech or not.

4.2 Training and Test

Since we are dealing with a text classification problem, any existing supervised learning methods can be applied. We decided to utilize *scikit*[5], which is an open source machine learning library for Python programming language [19]. We chose SVM, GNB, CNB, DT, KN, RF, and NN as our classifiers for all experiments. Hence, in this study, we will compare, summarize and discuss the behavior of these learning models with the linguistic features introduced above. In order to provide a comprehensive comparison between classifiers, we adopted the default values for all classifiers on all experiments.

1. Support Vector Machines (SVM): SVMs are supervised learning methods used for classification and regression, working effectively in high dimensional spaces. SVM classifiers show excellent performance on the text classification task. In our experiments, we chose LinearSVC from the scikit-learn library.
2. Naive Bayes (NB): NB methods are a set of supervised learning algorithms based on applying Bayes theorem with the Naive assumption of conditional independence between every pair of features given the value of the class variable. We used two algorithms from the scikit-learn library:
   - Gaussian Naive Bayes classifier. The likelihood of the features is assumed to be Gaussian:

$$P(x_i \mid y) = \frac{1}{\sqrt{2\pi\sigma_y^2}} \exp\left(-\frac{(x_i - \mu_y)^2}{2\sigma_y^2}\right) \tag{1}$$

[4] https://radimrehurek.com/gensim/
[5] http://scikit-learn.org/stable/

The parameters σ_y and μ_y are estimated using maximum likelihood.

– The Complement Naive Bayes classifier.
As described in [20]. The complement naive Bayes (CNB) algorithm is an adaptation of the standard multinomial naive Bayes (MNB) algorithm which is convenient for imbalanced data sets. More precisely, to compute the model's weights, CNB utilizes statistics from the complement of each class. The parameter estimates for CNB is better than those for MNB. Also, CNB outperforms MNB on text classification tasks. The procedure for calculating the weights is by the following equation :

$$\begin{aligned} \hat{\theta}_{ci} &= \frac{\alpha_i + \sum_{j:y_j \neq c} d_{ij}}{\alpha + \sum_{j:y_j \neq c} \sum_k d_{kj}} \\ w_{ci} &= \log \hat{\theta}_{ci} \\ w_{ci} &= \frac{w_{ci}}{\sum_j |w_{cj}|} \end{aligned} \tag{2}$$

where the summations are over all documents j which are not in class c; d_{ij} is either the count or tf-idf value of term i in document j; α_i is a smoothing hyperparameter as that found in MNB; and $\alpha = \sum_i \alpha_i$.

3. Decision Trees (DT): DTs are a non-parametric supervised learning method for regression and classification where predicts the value of a target variable by learning simple decision rules inferred from the data features.
4. Nearest Neighbors (KN): The precept behind nearest neighbor methods is to find a predefined number of training samples closest in the distance to the new point and predict the label from these. The number of samples can be a user-defined constant (k-nearest neighbor learning) or vary based on the local density of points (radius-based neighbor learning).
5. Random Forest(RF): RF classifier is an ensemble algorithm where each tree in the ensemble is built from a sample drawn with replacement (i.e., a bootstrap sample) from the training set. We used RandomForestClassifier from the scikit-learn library, which combines classifiers by averaging their probabilistic prediction, instead of letting each classifier vote for a single class, in contrast to the original publication [4].
6. Neural Network (NN): We only examined one class of neural network algorithms from the scikit-learn library which is Multilayer perceptron (MLP). It is one of the more basic neural network methods.

4.3 Discusion

Table 2 shows that CNB, SVM, and RF are by far the best classifiers for identifying the hate speech in both languages English and Spanish. N-grams features achieve the highest F1 scores in almost all tests regardless of whether the representations are CountVectorizer or TF-IDF. The performance of Gaussian Naive Bayes differs greatly depending on the number of features used in classification (see Table 2). GNB works better with a small number of features, more precisely

Table 2. Hate speech classification results of all classifiers, in terms of F1 scores for English and Spanish languages with linguistic features. The best F1 are highlighted (in bold).

	SVM		GNB		CNB		DT		KN		RF		NN	
	Eng.	Span.	Eng.	Span.	Eng.	Span.	Eng.	Span.	Eng.	Span.	Eng.	Span.	Eng.	Span.
TFIDF 1g	**0.76**	**0.77**	0.56	0.58	0.75	**0.77**	0.73	0.71	0.63	**0.73**	**.75**	0.75	0.71	0.72
TFIDF 2g	0.70	0.72	0.60	**0.60**	0.73	0.74	0.64	0.68	0.56	0.60	0.70	0.71	0.68	0.69
Countvect 1g	0.73	0.75	0.53	0.56	**0.76**	0.76	**0.76**	**0.74**	**0.66**	0.70	**0.75**	**0.77**	**0.74**	**0.75**
Countvect 2g	0.68	0.71	0.58	0.59	0.74	0.73	0.67	0.71	0.63	0.66	0.69	0.72	0.68	0.70
Doc2Vec	0.70	0.58	**0.65**	0.51	-	-	0.61	0.55	0.65	0.59	0.65	0.59	0.71	0.65

the best scores are achieved when it only uses Doc2Vec. It is worth noting that the combination of heterogeneous features hurts the performance of this type of classifier. By contrast, the Complement Naive Bayes showed a good performance and consistent with the features of the bag of words, whether they are represented by TF-IDF or Countvectorizer. However, in MNB and CNB classifiers, the input value of features must be non-negative. Therefore, they cannot deal with Doc2Vec features, as they contains negative value which is useless if we normalize the value to (0,1) as shown in Table 2.

Concerning the linguistic features, the best performance of most classifiers is reached with TF-IDF and Countvectorizer, for both 1g or 2g.

The DT classifier has similar behavior to SVM in terms of stability, but its performance tends to be much lower than that of SVM on both languages.

5 Conclusions

In this article, we have compared different supervised classifiers for a particular task. More precisely, we examined the performance of the most influence features within supervised learning methods (using SVM, GNB, CNB, DT, KN, RF, and NN), to identify the hate speech on English and Spanish tweets.

Concerning the comparison between machine learning strategies in this particular task, Support Vector Machine, Complement Naive Bayes, and Random Forest clearly outperforms all the rest classifiers and show stable performance with all features.

In future work, we will compare other types of classifiers with more complex linguistic features, by taking into account the new deep learning approaches based on neural networks.

Acknowledgments

Research partially funded by the Spanish Ministry of Economy and Competitiveness through projects TIN2017-85160-C2-2-R, and by the Galician Regional Government under projects ED431C 2018/50.

References

1. Almatarneh, S., Gamallo, P.: Comparing supervised machine learning strategies and linguistic features to search for very negative opinions. Information 10(1) (2019), http://www.mdpi.com/2078-2489/10/1/16
2. Almatarneh, S., Gamallo, P., Pena, F.J.R.: CiTIUS-COLE at semeval - 2019 task 5: Combining linguistic features to identify hate speech against immigrants and women on multilingual tweets. In: the 13th international Workshop on Semantic Evaluation (2019)
3. Basile, V., Bosco, C., Fersini, E., Nozza, D., Patti, V., Pardo, F.M.R., Rosso, P., Sanguinetti, M.: 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. pp. 54–63 (2019)
4. Breiman, L.: Random forests. Machine learning 45(1), 5–32 (2001)
5. Burnap, P., Williams, M.L.: Cyber hate speech on twitter: An application of machine classification and statistical modeling for policy and decision making. Policy & Internet 7(2), 223–242 (2015)
6. Burnap, P., Williams, M.L.: Hate speech, machine classification and statistical modelling of information flows on twitter: Interpretation and communication for policy decision making (2014)
7. Chen, Y., Zhou, Y., Zhu, S., Xu, H.: Detecting offensive language in social media to protect adolescent online safety. In: 2012 International Conference on Privacy, Security, Risk and Trust and 2012 International Confernece on Social Computing. pp. 71–80. IEEE (2012)
8. Collobert, R., Weston, J., Bottou, L., Karlen, M., Kavukcuoglu, K., Kuksa, P.: Natural language processing (almost) from scratch. Journal of machine learning research 12(Aug), 2493–2537 (2011)
9. Dai, A.M., Olah, C., Le, Q.V.: Document embedding with paragraph vectors. arXiv preprint arXiv:1507.07998 (2015)
10. Fortuna, P., Nunes, S.: A survey on automatic detection of hate speech in text. ACM Computing Surveys (CSUR) 51(4), 85 (2018)
11. Gaydhani, A., Doma, V., Kendre, S., Bhagwat, L.: Detecting hate speech and offensive language on twitter using machine learning: An n-gram and tfidf based approach. arXiv preprint arXiv:1809.08651 (2018)
12. Gitari, N.D., Zuping, Z., Damien, H., Long, J.: A lexicon-based approach for hate speech detection. International Journal of Multimedia and Ubiquitous Engineering 10(4), 215–230 (2015)
13. Greevy, E., Smeaton, A.F.: 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. pp. 468–469. ACM (2004)
14. Kwok, I., Wang, Y.: Locate the hate: Detecting tweets against blacks. In: Twenty-seventh AAAI conference on artificial intelligence (2013)
15. Le, Q., Mikolov, T.: Distributed representations of sentences and documents. In: International conference on machine learning. pp. 1188–1196 (2014)
16. Mikolov, T., Chen, K., Corrado, G., Dean, J.: Efficient estimation of word representations in vector space. arXiv preprint arXiv:1301.3781 (2013)
17. Mikolov, T., Sutskever, I., Chen, K., Corrado, G.S., Dean, J.: Distributed representations of words and phrases and their compositionality. In: Advances in neural information processing systems. pp. 3111–3119 (2013)

18. Nobata, C., Tetreault, J., Thomas, A., Mehdad, Y., Chang, Y.: Abusive language detection in online user content. In: Proceedings of the 25th international conference on world wide web. pp. 145–153. International World Wide Web Conferences Steering Committee (2016)
19. Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., et al.: Scikit-learn: Machine learning in python. Journal of machine learning research 12(Oct), 2825–2830 (2011)
20. Rennie, J.D., Shih, L., Teevan, J., Karger, D.R.: Tackling the poor assumptions of naive bayes text classifiers. In: Proceedings of the 20th international conference on machine learning (ICML-03). pp. 616–623 (2003)
21. Tulkens, S., Hilte, L., Lodewyckx, E., Verhoeven, B., Daelemans, W.: A dictionary-based approach to racism detection in dutch social media. arXiv preprint arXiv:1608.08738 (2016)
22. Unsvåg, E.F., Gambäck, B.: The effects of user features on twitter hate speech detection. In: Proceedings of the 2nd Workshop on Abusive Language Online (ALW2). pp. 75–85 (2018)