JCT at SemEval-2020 Task 12: Offensive Language Detection in Tweets using Preprocessing Methods, Character and Word N-grams

Moshe Uzan[1] **Yaakov HaCohen-Kerner**[2]
[1]Computer Science Department, Bar Ilan University, Ramat-Gan 5290002, Israel
[2]Jerusalem College of Technology (Lev Academic Center), Jerusalem 9116001, Israel
uzanmacho@gmail.com, kerner@jct.ac.il

Abstract

In this paper, we describe our submissions to SemEval-2020 contest. We tackled subtask 12 - "Multilingual Offensive Language Identification in Social Media". We developed different models for four languages: Arabic, Danish, Greek, and Turkish. We applied three supervised machine learning methods using various combinations of character and word n-gram features. In addition, we applied various combinations of basic preprocessing methods. Our best submission was a model we built for offensive language identification in Danish using Random Forest. This model was ranked at the 6th position out of 39 submissions. Our result is lower by only 0.0025 than the result of the team that won the 4th place using entirely non-neural methods. Our experiments indicate that char ngram features are more helpful than word ngram features. This phenomenon probably occurs because tweets are more characterized by characters than by words, tweets are short, and contain various special sequences of characters, e.g., hashtags, shortcuts, slang words, and typos.

1 Introduction

Nowadays, social networks occupy an increasingly important place in our life. However, their rapid development has also allowed the development of increasingly offensive, hateful content that takes advantage of the anonymous nature of the cyber world. This bullying, trolling, and harassment content has become a growing concern for governments, organizations, and individuals. The use of machine learning (ML) and natural language processing (NLP) solutions to find this offensive content has been surprisingly useful. Still, the detection of offensive language from social media is not an easy research problem due to the different levels of ambiguities present in natural language and the noisy nature of social media language. In addition, social media subscribers come from linguistically diverse communities. Up to now, most research has focused on the detection of offensive language in English but works have also been written for other languages like Struß et al. (2019) for German and Basile et al. (2019) for Spanish. Task 12-A of SEMEVAL 2020 deals with the detection of offensive language in tweets in five languages including English. In this paper, we describe our models submitted to subtask 12-A for tweets written in four languages Arabic, Danish, Greek, and Turkish. The full description of task 12 is given in Zampieri et al. (2020). The structure of the rest of the paper is as follows. Section 2 introduces a background concerning offensive language, tweet classification, and data preprocessing. Section 3 describes subtask 12 and its datasets. In Section 4, we present the submitted models and their experimental results. Section 5 summarizes and suggests ideas for future research.

2 Background

2.1 Offensive language

In recent years, more and more workshops on offensive content, hate speech, and aggression have been organized. This phenomenon shows the growing interest in the field. Schmidt and Wiegand (2017) noted

Proceedings of the 14th International Workshop on Semantic Evaluation, pages 2017–2022
Barcelona, Spain (Online), December 12, 2020.

in their survey paper that for hate speech detection the most used approach is the supervised one with a focus on support vector machines (SVM) and recurrent neural networks (RNN) (Pavlopoulos et al., 2017). Zampieri et al. (2019) showed that n-grams can perform well for hate speech detection using SVMs with different surface-level features, such as surface n-grams, word skip-grams, and word representation n-grams induced with Brown clustering. They also noticed that these features reached their limits for more complex tasks, e.g., distinguishing profanity and hate speech. In such tasks, more in-depth linguistic characteristics may be required. Most of the existing work essentially concerns the English language (Xu et al. (2012), Burnap and Williams (2015), Davidson et al. (2017)). However, the discussed competition has provided datasets not only for English, but also for four other languages Arabic, Danish, Greek, and Turkish. Given the large size of the English dataset and the resources required to train models on it, we chose to focus on the four other languages. The datasets for these four languages are relatively small and unbalanced. Therefore, we prefer using classical ML methods and avoid using deep-learning ones such as RNN or CNN.

2.2 Tweet Classification

Tweet classification means the classification of a tweet into one of a set of predefined classes. The most common general method for automatically performing this task is through supervised ML methods. We also adopted this general method. Tweets are a challenge for classification as they are informal, short, and contain various misspellings, shortenings, and slang words (HaCohen-Kerner et al., 2017).

2.3 Text preprocessing

Text preprocessing is an important step in text mining and ML processes. In text documents in general and in social text documents in particular, it is common to find typos, emojis, slang, HTML tags, spelling mistakes, and repetitive letters. Analyzing text that has not been carefully cleaned or preprocessed might lead to misleading results. Not all of the preprocessing types are considered effective by all text classification (TC) researchers. For instance, Forman (2003), in his study on feature selection metrics for TC, claimed that stop words are ambiguous and therefore should be removed. However, HaCohen-Kerner et al. (2008) demonstrated that the use of word unigrams including stop words lead to improved TC results compared to the results obtained using word unigrams excluding stop words. In our system, we applied various combinations of preprocessing types depending on the language. For all the languages we convert the tweet's characters to lowercase and add a start token and an end token. We first tried to remove stop words but probably due to the already reduced size of the tweet datasets, this led to poorer results. We also tried to use a list of offensive words but in most cases, these words already appeared in the words (or sometimes even in the character n-grams) that we collected in the first step. For Arabic, given the different forms that many words can have, we convert each word to its standard form and remove the noise caused by different vowels. To do so we applied various python functions for developed by Maxim Romanov[1].

3 Task Description

SemEval-2020 Task 12 (Zampieri et al., 2020) addresses the problem of detecting offensive language in tweets.It is a based on the last edition: SemEval-2019 Task 6. (Zampieri et al., 2019). For Arabic (Mubarak et al., 2020), Danish (Sigurbergsson and Derczynski, 2020), English (Rosenthal et al., 2020), Greek (Pitenis et al., 2020), and Turkish (Çöltekin, 2020), task 12 deals with the identification of offensive language in tweets. That is to say, we have to classify each tweet as offensive (OFF) or not offensive (NOT). For English, there were two additional tasks: Automatic categorization of offense types and Offense target identification. As mentioned and explained above, we did not participate in tasks for the English language. A development dataset has been provided for Arabic but not for the other languages in which we split the provided training dataset and allocate 20% of this data for development. Table 1 presents the size and distribution of each subset we used for the applied languages.

[1] https://alraqmiyyat.github.io/2013/01-02.html

4 The Submitted Models and Experimental Results

The two co-authors of this paper are from different academic places: Bar-Ilan University (BIU) and Jerusalem College of Technology (JCT). Different models have been applied and submitted from two users BIU and JCT. In this paper, we will present the best submitted JCT models. We applied the following three supervised ML methods on the training datasets: Random Forest (RF), Support Vector Classifier (SVC), and Multi-Layer Perceptron (MLP).

Language	Train subset		Dev subset		Test Subset
	# of tweet	# of offensive one	# of tweet	# of offensive one	# of tweet
Arabic	6,839	1,411	1,000	179	2,000
Danish	2,368	309	592	75	329
Greek	6,994	1,976	1,749	510	1,544
Turkish	25,404	4,932	6,352	1,199	3,528

Table 1: Information about the subsets for the applied languages.

Random forest (RF) is an ensemble learning method for classification and regression (Breiman, 2001). Ensemble methods use multiple learning algorithms to obtain improved predictive performance compared to what can be obtained from any of the constituent learning algorithms. RF operates by constructing a multitude of decision trees at training time and outputting classification for the case at hand. RF combines Breiman's "bagging" (Bootstrap aggregating) idea in (Breiman, 1996) and a random selection of features introduced by Ho (1995) to construct a forest of decision trees.
SVC is a variant of the support vector machine (SVM) ML method (Cortes and Vapnik, 1995) implemented in SciKit-Learn. SVC uses LibSVM (Chang and Lin, 2011), which is a fast implementation of the SVM method. SVM is a supervised ML method that classifies vectors in a feature space into one of two sets, given training data. It operates by constructing the optimal hyperplane dividing the two sets, either in the original feature space or in higher dimensional kernel space.
Multi-layer perceptron (MLP) is a deep, artificial neural network (Pal and Mitra, 1992). This model is based on a network of the computational unit, called perceptron, interconnected in a feed-forward way. The network is composed of layers of perceptron that each one has directed connections to the neurons of the subsequent layer. Usually these units apply a sigmoid function, called the activation function, on the input they get and feed the next layer with the output of the function. This model is very useful especially when the data is not linearly separable. We preferred to use the Relu function as activation function which show better convergence performance than sigmoid (Krizhevsky et al., 2012) and logistic loss as loss function. We also Adam solver for weight optimization (Kingma and Ba, 2014) and a batch size of 200. These ML methods were applied using the following tools and information sources:

- The Python 3.7.3 programming language[2]
- Scikit-learn – a Python library for ML methods[3]
- Numpy – a Python library that provides fast algebraic calculous processing especially for multidimensional object[4].

Table 2 presents the Macro F1-Score results of our best JCT's models in task 12-A. On each language, we applied these three different supervised ML methods for various combinations of character and word n-gram features from unigram to 5-gram. We, then, selected the model that gives the best result and we try to combine them. Finally We only submit one model per language, so the results of the other models on the test are not available (marked as NA in the table). Our submitted models for at least three languages (Arabic, Danish, and Greek) show that the most helpful type of feature sets for the tweet classification

[2]https://www.python.org/downloads/release/python- 373/
[3]https://scikit-learn.org/stable/index.html
[4]https://numpy.org

Language	ML Method	Features	Result on Dev	Result of submission
Arabic	**Majority Baseline**		**0.821**	**0.4441**
	SVC	6,500 char trigrams 1,000 char bigrams 10,500 char 4-grams	0.8065	0.4959
	MLP	6,500 char trigrams 1,000 char bigrams 10,500 char 4-grams	0.8275	N/A
	RF	9,000 char 4-grams	0.805	N/A
Danish	**Majority Baseline**		**0.873**	**0.4668**
	RF	12,000 char 4-grams	0.7994	0.7741
	RF	4,500 char trigrams	0.7994	N/A
	SVC	7,500 char 4-grams	0.7328	N/A
	MLP	8,000 char trigrams	0.7233	N/A
Greek	**Majority Baseline**		**0.708**	**0.4575**
	MLP	9,500 char 4-grams 17,000 char 5-grams 4,500 char trigrams	0.7571	0.7568
	RF	9,500 char 4-grams	0.7851	N/A
	SVC	14,000 char 4-grams	0.762	N/A
Turkish	**Majority Baseline**		**0.811**	**0.4435**
	SVC	6,000 char trigrams 13,000 char 4-grams 14,000 word unigrams	0.7095	0.5099
	RF	16,500 char 5-grams 14,000 word unigrams	0.6618	N/A
	MLP	6,000 char trigrams 13,000 char 4-grams 16,500 char 5-grams 14,000 word unigrams	0.7257	N/A

Table 2: Macro F1-Score results of our best models in task 12-A.

tasks is the char ngram features. Specifically, the best char ngram features are usually thousands of bigram and/or trigram and/or 4-grams and/or 5-grams features. These findings are because tweets are more characterized by characters than by words, tweets are relatively short, and contain also emojis, hashtags, onomatopoeia, slang words, shortcuts, and typos. Only for the Turkish language, the two submitted models used word n-grams (in addition to char ngrams). However, the results of these two models are relatively low.

Like we can see with the results of the majority baseline for each language the presence of offensive tweet in the submission set is disproportionate compared to this one in the training and development set. This can explain the differences between the results obtained by some models on the development set and on the submission, like for example the SVC on Arabic that reached a F1 of 0.8065 on the result.

5 Conclusions and Future Research

In this paper, we describe our submissions to subtask 12 of SemEval-2020 contest. Like said before, we developed several models for each language for four languages: Arabic, Danish, Greek, and Turkish, after what, we select the two best models while trying to choose two models that use a different method of supervised learning and submit them. Our best submission was a model we built for offensive language identification in Danish using Random Forest. This model was ranked at the 6th position out of 39 submissions. Its Macro F1-Score result (0.7741) is lower by only 0.003 than the result of the team that

won the 4th place. It is interesting to note that this is the only model among those in the top which does not use a neural method which show that classical methods still remains relevant.

Many tweets contain acronyms that can be presented in different forms. These acronyms can lead to ambiguity. Future research may seek to lessen this ambiguity. Acronym disambiguation, e.g. HaCohen-Kerner et al. (2010a), will extend and enrich the tweet's text and might enable better classification. We also suggest examining the usefulness of skip character n-grams because they serve as generalized ngrams that allow us to overcome problems such as noise and sparse data (HaCohen-Kerner et al., 2017). Other ideas that may lead to better classification are to use stylistic feature sets (HaCohen-Kerner et al., 2010b), key phrases (HaCohen-Kerner et al., 2007), and summaries (HaCohen-Kerner et al., 2003). We may also apply advanced deep learning methods and especially RNN, which is more suitable for text applications.

References

Valerio Basile, Cristina Bosco, Elisabetta Fersini, Nozza Debora, Viviana Patti, Francisco Manuel Rangel Pardo, Paolo Rosso, Manuela Sanguinetti, et al. 2019. Semeval-2019 task 5: Multilingual detection of hate speech against immigrants and women in twitter. In *13th International Workshop on Semantic Evaluation*, pages 54–63. Association for Computational Linguistics.

Leo Breiman. 1996. Bagging predictors. *Machine learning*, 24(2):123–140.

Leo Breiman. 2001. Random forests. *Machine learning*, 45(1):5–32.

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.

Çağrı Çöltekin. 2020. A Corpus of Turkish Offensive Language on Social Media. In *Proceedings of the 12th International Conference on Language Resources and Evaluation*. ELRA.

Chih-Chung Chang and Chih-Jen Lin. 2011. Libsvm: A library for support vector machines. *ACM transactions on intelligent systems and technology (TIST)*, 2(3):1–27.

Corinna Cortes and Vladimir Vapnik. 1995. Support-vector networks. *Machine learning*, 20(3):273–297.

Thomas Davidson, Dana Warmsley, Michael Macy, and Ingmar Weber. 2017. Automated hate speech detection and the problem of offensive language. In *Eleventh international aaai conference on web and social media*.

George Forman. 2003. An extensive empirical study of feature selection metrics for text classification. *Journal of machine learning research*, 3(Mar):1289–1305.

Yaakov HaCohen-Kerner, Eylon Malin, and Itschack Chasson. 2003. Summarization of jewish law articles in hebrew. In *CAINE*, pages 172–177.

Yaakov HaCohen-Kerner, Ittay Stern, David Korkus, and Erick Fredj. 2007. Automatic machine learning of keyphrase extraction from short html documents written in hebrew. *Cybernetics and Systems: An International Journal*, 38(1):1–21.

Yaakov HaCohen-Kerner, Dror Mughaz, Hananya Beck, and Elchai Yehudai. 2008. Words as classifiers of documents according to their historical period and the ethnic origin of their authors. *Cybernetics and Systems: An International Journal*, 39(3):213–228.

Yaakov HaCohen-Kerner, Hananya Beck, Elchai Yehudai, and Dror Mughaz. 2010a. Stylistic feature sets as classifiers of documents according to their historical period and ethnic origin. *Applied Artificial Intelligence*, 24(9):847–862.

Yaakov HaCohen-Kerner, Ariel Kass, and Ariel Peretz. 2010b. Haads: A hebrew aramaic abbreviation disambiguation system. *Journal of the American Society for Information Science and Technology*, 61(9):1923–1932.

Yaakov HaCohen-Kerner, Ziv Ido, and Ronen Ya'akobov. 2017. Stance classification of tweets using skip char ngrams. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 266–278. Springer.

Tin Kam Ho. 1995. Random decision forests. In *Proceedings of 3rd international conference on document analysis and recognition*, volume 1, pages 278–282. IEEE.

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

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. 2012. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105.

Hamdy Mubarak, Ammar Rashed, Kareem Darwish, Younes Samih, and Ahmed Abdelali. 2020. Arabic offensive language on twitter: Analysis and experiments. *arXiv preprint arXiv:2004.02192*.

Sankar K Pal and Sushmita Mitra. 1992. Multilayer perceptron, fuzzy sets, classifiaction.

John Pavlopoulos, Prodromos Malakasiotis, and Ion Androutsopoulos. 2017. Deep learning for user comment moderation. *arXiv preprint arXiv:1705.09993*.

Zeses Pitenis, Marcos Zampieri, and Tharindu Ranasinghe. 2020. Offensive Language Identification in Greek. In *Proceedings of the 12th Language Resources and Evaluation Conference*. ELRA.

Sara Rosenthal, Pepa Atanasova, Georgi Karadzhov, Marcos Zampieri, and Preslav Nakov. 2020. A Large-Scale Semi-Supervised Dataset for Offensive Language Identification. In *arxiv*.

Anna Schmidt and Michael Wiegand. 2017. A survey on hate speech detection using natural language processing. In *Proceedings of the Fifth International workshop on natural language processing for social media*, pages 1–10.

Gudbjartur Ingi Sigurbergsson and Leon Derczynski. 2020. Offensive Language and Hate Speech Detection for Danish. In *Proceedings of the 12th Language Resources and Evaluation Conference*. ELRA.

Julia Maria Struß, Melanie Siegel, Josef Ruppenhofer, Michael Wiegand, Manfred Klenner, et al. 2019. Overview of germeval task 2, 2019 shared task on the identification of offensive language.

Jun-Ming Xu, Kwang-Sung Jun, Xiaojin Zhu, and Amy Bellmore. 2012. Learning from bullying traces in social media. In *Proceedings of the 2012 conference of the North American chapter of the association for computational linguistics: Human language technologies*, pages 656–666.

Marcos Zampieri, Shervin Malmasi, Preslav Nakov, Sara Rosenthal, Noura Farra, and Ritesh Kumar. 2019. Predicting the type and target of offensive posts in social media. *arXiv preprint arXiv:1902.09666*.

Marcos Zampieri, Preslav Nakov, Sara Rosenthal, Pepa Atanasova, Georgi Karadzhov, Hamdy Mubarak, Leon Derczynski, Zeses Pitenis, and Çağrı Çöltekin. 2020. SemEval-2020 Task 12: Multilingual Offensive Language Identification in Social Media (OffensEval 2020). In *Proceedings of SemEval*.