JCTICOL at SemEval-2019 Task 6: Classifying Offensive Language in Social Media using Deep Learning Methods, Word/Character N-gram Features, and Preprocessing Methods

Yaakov HaCohen-Kerner, Ziv Ben-David, Gal Didi,
Eli Cahn, Shalom Rochman, and Elyashiv Shayovitz
Department of Computer Science, Jerusalem College of Technology, Lev Academic Center
21 Havaad Haleumi St., P.O.B. 16031, 9116001 Jerusalem, Israel
kerner@jct.ac.il,benda1237@gmail.com,
galdd8@gmail.com, eli.cahn@gmail.com,
shal.rochman@gmail.com, elyashiv12@gmail.com

Abstract

In this paper, we describe our submissions to SemEval-2019 task 6 contest. We tackled all three sub-tasks in this task "OffensEval - Identifying and Categorizing Offensive Language in Social Media". In our system called JCTICOL (Jerusalem College of Technology Identifies and Categorizes Offensive Language), we applied various supervised ML methods. We applied various combinations of word/character n-gram features using the TF-IDF scheme. In addition, we applied various combinations of seven basic preprocessing methods. Our best submission, an RNN model was ranked at the 25th position out of 65 submissions for the most complex sub-task (C).

1 Introduction

Offensive language is frequent in social media. For instance, ScanSafe's monthly "Global Threat Report" reported that up to 80% of blogs contained offensive contents and 74% included porn in the format of the image, video, or offensive language (Cheng, 2007). There are people that take advantage of the perceived anonymity of computer-mediated communication, using this to write in behavior that many of them would not consider in real life.

Online news and social networking services, online communities, social media platforms, and various computer companies have been investing a lot of effort, time and money to cope with offensive language in order to prevent abusive behavior.

Computational methods are among the most effective strategies to identify various types of aggression, offense, and hate speech in user-generated content (e.g., comments, microblogs, posts, and tweets). Detection of offensive language has been investigated in recent years in various studies (Waseem et al. 2017; Davidson et al., 2017, Malmasi and Zampieri, 2018, Kumar et al. 2018) and various workshops such as ALW (Abusive Language Online) and TRAC (Trolling, Aggression, and Cyberbullying).

In this paper, we describe our submissions to SemEval-2019 task 6 contest. In Task-6, OffensEval, there are three different sub-tasks. Sub-task A deals with offensive language identification. Sub-task B deals with the automatic categorization of offense types. Sub-task C deals with offense target identification.

The report of the OffensEval task is described in Zampieri et al. (2019A) and the description of the OLID dataset that was used for the competition is in Zampieri et al. (2019B).

The structure of the rest of the paper is as follows. Section 2 discusses work related to offensive language in social media, tweet classification, and data preprocessing. Section 3 presents, in general, the task description. In Section 4, we describe the submitted models and their experimental results. Section 6 summarizes and suggests ideas for future research.

2 Background

2.1 Offensive Language in Social Media

In recent years, there has been an increase in the number of studies dealing with Offensive language in social media. Nobata et al. (2016) developed a machine learning based method to detect hate speech on online user comments from two domains. They also built a corpus of user comments annotated accordingly to three subcategories (hate speech, derogatory,

Proceedings of the 13th International Workshop on Semantic Evaluation (SemEval-2019), pages 645–651
Minneapolis, Minnesota, USA, June 6–7, 2019. ©2019 Association for Computational Linguistics

profanity). Waseem and Hovy (2016) introduced a list of criteria founded in critical race theory and used them to label a publicly available corpus of more than 16k tweets with tags about both racial and sexist offenses.

A survey on hate speech detection is presented by Schmidt and Wiegand (2017). The authors introduced various NLP methods that were developed in order to detect hate speech. Davidson et al. (2017) presented a multi-class classifier to distinguish between three categories: hate speech, offensive language, and none of these two. The analysis of the predictions and the errors show when we can reliably separate hate speech from other offensive language and when this differentiation is more difficult. Anzovino et al. (2018) built a labelled corpus containing 2,227 misogynous (hate speech against women) tweets and no-misogynous tweets and explored various NLP features and ML models for detecting and classifying misogynistic language.

2.2 Tweet Classification

Sriram et al. (2010) presented a new classification model that uses a small set of domain-specific features extracted from the author‟s profile and text. Experimental results showed that the classification accuracy of their model is better than the classification accuracy of the traditional Bag-Of-Words model. Batool et al. (2013) introduced a system that extracts knowledge from tweets and then classifies the tweets based on the semantics of knowledge contained in them. For avoiding information loss, knowledge enhancer is applied that enhances the knowledge extraction process from the collected tweets.

Stance classification of tweets was investigated by HaCohen-Kerner et al. (2017). Given test datasets of tweets from five various topics, they classified the stance of the tweet authors as either in FAVOR of the target, AGAINST it, or NONE. Their algorithm used a few tens of features mainly character-based features where most of them are skip char ngram features. The experimental results showed that this algorithm significantly outperforms the traditional 'bag-of-words' model.

2.3 Data preprocessing

Data preprocessing is an important step in data mining (DM) and ML processes. In tweets, it is common to find typos, emojis, slang, HTML tags, spelling mistakes, irrelevant and redundant information. Analyzing data that has not been carefully cleaned or pre-processed might lead to misleading results.

Not all of the preprocessing types are considered effective in the text classification community. For instance, Forman (2003), in his study on feature selection metrics for text classification, claimed that stop words occurring frequently and 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 text classification results compared to the results obtained using word unigrams excluding stop words in the domain of Hebrew-Aramaic Jewish law documents.

In our system, we applied various combinations of seven basic preprocessing types: C - spelling Correction[1] using a dictionary of containing 479k English words[2], L – converting to Lowercase letters, P – Punctuation removal, S – Stopwords Removal, R – Repeated characters removal, T – sTemming, and M - leMmatizion) in order to employ the best combination.

3 The Competition of Task 6

SemEval-2019 Task 6 consists of three subtasks:

1. Subtask A: Given a tweet, predict whether it contains offensive language or a targeted (veiled or direct) offense or it does not contain offense or profanity.
2. Subtask B: Given a tweet containing offensive language, predict whether it contains an insult or threat to an individual, a group, or others, or contains non targeted profanity and swearing.
3. Subtask C: Given a tweet containing an insult or threat, predict whether the target is an individual or a group of people considered as a unity due to the same ethnicity, gender or sexual orientation, political affiliation, religious belief, or something else, or does not belong to any of the previous two categories (e.g., an organization).

The dataset of Task 6 contains tweets that were annotated using crowdsourcing. The dataset of sub-task A contains 13,240 tweets: 4,404 OFF (Offensive language) tweets (about 33%) and 8,836 NOT (Not Offensive) tweets (about 67%).

[1] https://github.com/phatpiglet/autocorrec.

[2] https://github.com/dwyl/english-words.

The dataset of sub-task B contains 4,400 offensive tweets: 3,876 TIN (Targeted Insult) tweets (about 88%) and 524 UNT (Untargeted) tweets. The dataset of sub-task C contains 3,876 tweets: 2,407 IND (Individual) tweets (about 62%), 1,074 GRP (Group) tweets (about 28%), and 395 OTH (Other) (about 10%). The test data of sub-tasks A, B, and C contain 860, 240, and 213 unlabeled tweets, respectively.

4 The Submitted Models and Experimental Results

We have submitted 17 models: 6 models to task 6-A, 6 models to task 6-B, and 5 models to task 6-C. We applied the Python module called Scikit-learn (Pedregosa et al., 2011) using the TF-IDF scheme called TfidfTransformer[3] and we applied various supervised ML methods with various numbers of n-gram features, skip word/char n-grams (HaCohen-Kerner et al., 2017) and combinations of pre-processing types.

While all teams' submissions in all three sub-tasks of task 6 were ranked according to their F-Measure scores, we were wrong in all these sub-tasks in the sense that we submitted models according to their accuracy scores.

Most of our submitted models were RNN models. Each RNN model was a bidirectional RNN with 4 hidden layers, with different numbers of LSTMs, values of Dropout, and number of vectors of GloVe (Pennington et al., 2014). Additional explanations to our RNN models, which are given in the next paragraphs are mainly based on the explanations given by Nikolai Janakiev in "Practical Text Classification with Python and Keras"[4].

We used the Tokenizer utility class, which converts a text corpus into a list of integers. Each integer maps to a value in a dictionary that encodes the entire corpus, with the dictionary's keys being the vocabulary terms themselves.

We chose to use the Twitter-aware tokenizer, designed to be flexible and easy to adapt to new domains and tasks (e.g., for tweet processing).

We used the word embeddings method. This method represents words as dense word vectors, which are trained, unlike the one-hot encoding which is hardcoded. The word embeddings map the statistical structure of the language used in the corpus. Their aim is to map semantic meaning into a geometric space. This geometric space is then called the embedding space. This method would map semantically similar words close on the embedding space.

There are two options to get such a word embedding. One way is to train the word embeddings during the training of our neural network. The other way is to use a precomputed embedding space that utilizes a larger corpus. Among the most popular methods are GloVe (Global Vectors for Word Representation) developed by the Stanford NLP Group (Pennington et al., 2014) and Word2Vec developed by Mikolov et al. (2013).

GloVe applies a co-occurrence matrix and by using matrix factorization while Word2Vec applies neural networks. Word2Vec is more accurate and GloVe is faster to compute. We used the GloVe method for our model.

4.1 Results of Task 6-A

Table 1 presents the main characteristics and results of our six submitted models to task 6-A. The models are presented in descending order according to their F-measure score on the test set.

[3] https://scikit-learn.org/stable/modules/generated/sklearn.feature_extraction.text.TfidfTransformer.html#sklearn.feature_extraction.text.TfidfTransformer

[4] https://realpython.com/python-keras-text-classification/#author

The first name of the model authors	Pre-processing	Model				Score			
		ML Method	N-Gram Features	Additional Features (for RNN only)	FC Layer (for RNN only)	CV	Test Scores		
						Acc.	F-M	Acc.	Rank
JCTICOL-Ziv Ben-David	-	RNN	5000 word unigrams, 100 word bigrams	512 LSTMs, 0.2 Dropout. GloVe: 100d.	Logistic Regression Random Forest SVM-linear SVC (kernel=linear) SVC (kernel=rbf) SVC (DFS[5]=ovo) KNeighbors	0.85	0.74	0.81	49/ 103
JCTICOL-Eli Cahn	CLS	RNN	None	512 LSTMs, 0.2 Dropout. GloVe: 100d.	-	0.77	0.73	0.8	50/ 103
JCTICOL-Gal Didi	-	RNN	5000 word unigrams, 100 word bigrams	512 LSTMs, 0.3 Dropout. GloVe: 100d.	Logistic Regression Random Forest SVM-linear	0.85	0.73	0.79	52/ 103
JCTICOL-Shalom Rochman	L	RNN	None	512 LSTMs, 0.2 Dropout. GloVe: 100d.	-	0.75	0.73	0.81	59/ 103
JCTICOL-Elyashiv Shayovitz	-	RNN	5000 word unigrams, 100 word bigrams	512 LSTMs, 0.2 Dropout. GloVe: 100d.	SVC (kernel=rbf)	0.86	0.72	0.81	62/ 103
JCTICOL-Yaakov HaCohen-Kerner	-	SVM-linear	5000 word unigrams, 200 word bigrams, 100 words trigrams	-	-	0.72	0.72	0.78	67/ 103

Table 1: Results of our 6 models in task-A.

The main results and conclusions that can be derived from Table 1 are as follows:

- The best submitted model is an RNN model with F-measure of 0.74 and accuracy of 0.81 obtaining the 43rd position out of 103 submissions.
- The best combination of N-gram features for this model contains 5000 word unigrams and 100 word bigrams (without any word trigrams).
- In addition, this model used 512 LSTMs and in its FC layer, it used seven different ML methods.
- This model did not use any combination of pre-processing types.
- Simpler RNN models and non RNN models e.g. the SVM-linear model (last row in Table 1) as well as other models that were tested but not submitted, were less successful.

4.2 Results of Task 6-B

Table 2 presents the main characteristics and results of our six submitted models to task 6-B. The models are presented in descending order according to their F-measure score on the test set. It should be noted that the train set for sub-task b contains imbalanced sets of tweets. The number of tweets classified as UNT is 524 (about 12%) while the number of tweets classified as TIN is 3,876 (about 88%). The main results and conclusions that can be derived from Table 2 are as follows:

- Our best submitted model is a SVC - support vector classifier with F-measure of 0.49 and accuracy of 0.85 obtaining the 62nd position

[5] DFS – Decision Function Shape.

out of 75 submissions. This model used a combination of the MPR pre-processing types.

- This model used 10,000 char trigrams where for each character trigram we allow up to a maximum of 7 skipped characters in-between the chosen ones.
- As mentioned before, while the submitted models were ranked according to their F-Measure results, we were wrong and submit models according to their accuracy results.

User	Pre-processing	Model		Score			
		ML Method	N-Gram Features	CV	Test Score		
				Acc.	Macro-F1	Acc.	Rank
JCTICOL-Eli Cahn	MPR	SVC - Support vector classifier	10000 char trigrams with 7 skips	0.87	0.49	0.85	62 / 75
JCTICOL-Ziv Ben- David	L	MLP - Multilayer perceptron	10000 char unigrams with 4 skips	0.87	0.48	0.85	63 / 75
JCTICOL-Gal Didi	MPRS	SVC - Support vector classifier	7000 word bigrams with 0 skips	0.87	0.47	0.82	65 / 75
JCTICOL-Elyashiv Shayovitz	CMPR	LR - Logistic regression	10000 char bigrams with 7 skips	0.87	0.47	0.89	66 / 75
JCTICOL-Yaakov HaCohen-Kerner	CLS	SVC - Support vector classifier	1000 char trigrams with 9 skips	0.87	0.47	0.89	67 / 75
JCTICOL-Shalom Rochman	CMP	RF - Random forest	7000 word unigrams with 0 skips	0.87	0.47	0.81	69 / 75

Table 2: Results of our 6 models in task-B.

4.3 Results of Task 6-C

Table 3 presents the main characteristics and results of our six submitted models to task 6-C. The models are presented in descending order according to their F-measure score on the test set.

The main results and conclusions that can be derived from Table 3 are as follows:

- Our best submitted model is an RNN model with F-measure of 0.53 and accuracy of 0.64 obtaining the 25th position out of 65 submissions.
- The best combination of N-gram features for this model contains 5000 word unigrams and 200 word bigrams (without any word trigrams).
- In addition, this model used 512 LSTMs and in its FC layer it used only the SVC ML method.
- This model did not use any combination of pre-processing types.
- Simpler RNN models and non RNN models such as the SVC-linear model (last row in Table 3) as well as other models that were tested but not submitted to the competition, were less successful.

User	Pre processing	Model				Score			
		ML Method	N-Gram Features	Additional Features (for RNN only)	FC Layer (for RNN only)	CV	Test Scores		
						Acc.	F-M	Acc.	Rank
JCTICOL-Gal Didi	-	RNN	5000 word unigrams, 200 word bigrams	512 LSTMs, 0.3 Dropout. GloVe: 200d special for Tweeter	SVC (kernel=linear)	0.88	0.53	0.64	25 / 65
JCTICOL-Ziv Ben-David	-	RNN	8000 word unigrams, 200 word bigrams	512 LSTMs, 0.3 Dropout. GloVe: 200d special for Tweeter	Logistic Regression	0.89	0.51	0.64	33 / 65
JCTICOL-Elyashiv Shayovitz	-	RNN	5000 word unigrams, 200 word bigrams	512 LSTMs, 0.3 Dropout. GloVe: 200d special for Tweeter	-	0.68	0.50	0.67	40 / 65
JCTICOL-Yaakov HaCohen-Kerner	-	RNN	5000 word unigrams, 200 word bigrams	512 LSTMs, 0.3 Dropout. GloVe: 200d special for Tweeter	SVM-linear	0.88	0.49	0.62	42 / 65
JCTICOL-Shalom Rochman	-	SVC_ (kernel= linear)	5000 word unigrams, 200 word bigrams	-	-	0.64	0.42	0.54	58 / 65

Table 3: Results of our 6 models in task-C.

5 Conclusions and Future Research

In this paper, we describe our submissions to three sub-tasks of Task 6 of SemEval-2019 contest. Our system JCTICOL (Jerusalem College of Technology Identifies and Categorizes Offensive Language) includes 17 formal submissions: 6 for sub-task A, 6 for sub-task B, and 5 for sub-task C. We used the TF-IDF scheme and we applied various supervised ML methods with various numbers of n-gram features and combinations of pre-processing types. Our best submission was ranked at the 25th position out of 65 submissions for the most complex sub-task (C).

Future research proposals that may contribute to better classification are as follows. (1) Using additional feature sets such as stylistic feature sets (HaCohen-Kerner et al., 2010B) and keyphrases that can be extracted from the text corpora (HaCohen-Kerner et al., 2007); (2) Using acronym disambiguation (e.g., HaCohen-Kerner et al., 2010A), i.e., selecting the correct long form of the acronym depending on its context will enrich the tweet's text; and (3) Using other deep learning models.

Acknowledgments

This research was partially funded by the Jerusalem College of Technology, Lev Academic Center.

References

Rabia Batool, Asad Masood Khattak, Jahanzeb Maqbool, and Sungyoung Lee 2013. Precise tweet classification and sentiment analysis. In Computer and Information Science (ICIS), 2013 IEEE/ACIS 12th International Conference on (pp. 461-466). IEEE.

Jacqui Cheng. 2007. Report: 80 percent of blogs contain "offensive" content, in ars technica. vol. 2011.

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, pages 512-515.

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, 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, Ariel Kass, and Ariel Peretz. 2010A. HAADS: A Hebrew Aramaic abbreviation disambiguation system. Journal of the American Society for Information Science and Technology, 61(9), 1923-1932.

Yaakov HaCohen-Kerner, Hananya Beck, Elchai Yehudai, and Dror Mughaz. 2010B. 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, 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 (pp. 266-278). Springer, Cham.

Ritesh Kumar, Atul Kr. Ojha, Shervin Malmasi, and Marcos Zampieri. 2018. Benchmarking aggression identification in social media. In Proceedings of the First Workshop on Trolling, Aggression, and Cyberbullying (TRAC-2018) (pp. 1-11).

Shervin Malmasi and Marcos Zampieri. 2018. Challenges in discriminating profanity from hate speech. Journal of Experimental & Theoretical Artificial Intelligence, 30(2), 187-202.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. arXiv preprint arXiv:1301.3781.

Chikashi Nobata, Joel Tetreault, Achint Thomas, Yashar Mehdad, and Yi Chang. 2016. Abusive language detection in online user content. In: Proceedings of the 25th International Conference on World Wide Web, pp. 145–153. International World Wide Web Conferences Steering Committee.

Fabian Pedregosa, Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R. and Dubourg, V. and Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., and Duchesnay, E. 2011. Scikit-learn: Machine learning in Python. Journal of machine learning research, 12(Oct), 2825-2830.

Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. GloVe: Global Vectors for Word Representation.

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. Association for Computational Linguistics, Valencia, Spain, pages 1–10.

Bharath Sriram, David Fuhry, Engin Demir, Hakan Ferhatosmanoglu. 2010. Short text classification in twitter to improve information filtering. In Proceedings of the 33rd international ACM SIGIR conference on Research and development in information retrieval (pp. 841-842). ACM.

Zeerak Waseem, Thomas Davidson, Dana Warmsley, and Ingmar Weber. 2017. Understanding abuse: A typology of abusive language detection subtasks. arXiv preprint arXiv:1705.09899.

Marcos Zampieri, Shervin Malmasi, Preslav Nakov, Sara Rosenthal, Noura Farra, and Ritesh Kumar. 2019A. SemEval-2019 Task 6: Identifying and Categorizing Offensive Language in Social Media (OffensEval). In Proceedings of the 13th International Workshop on Semantic Evaluation (SemEval).

Marcos Zampieri, Shervin Malmasi, Preslav Nakov, Sara Rosenthal, Noura Farra, and Ritesh Kumar. 2019B. Predicting the Type and Target of Offensive Posts in Social Media. In Proceedings of NAACL.