# Emad at SemEval-2019 Task 6: Offensive Language Identification using Traditional Machine Learning and Deep Learning approaches

**Emad Kebriaei[1], Samaneh Karimi[2], Nazanin Sabri[3], Azadeh Shakery[4]**
[1234]School of Electrical and Computer Engineering, College of Engineering, University of Tehran, Iran
[4]School of Computer Science and Institute for Research in Fundamental Sciences (IPM)
{emad.kebriaei,samanekarimi,nazaninsabri,shakery}@ut.ac.ir

## Abstract

In this paper, the used methods and the results obtained by our team, entitled Emad, on the OffensEval 2019 shared task organized at SemEval 2019 are presented. The OffensEval shared task includes three sub-tasks namely Offensive language identification, Automatic categorization of offense types and Offense target identification. We participated in sub-task A and tried various methods including traditional machine learning methods, deep learning methods and also a combination of the first two sets of methods. We also proposed a data augmentation method using word embedding to improve the performance of our methods. The results show that the augmentation approach outperforms other methods in terms of macro-f1.

## 1 Introduction

With the growth of social networking platforms, the need for automatic methods that manages the emerging issues or facilitate using them is rising. One of the rising trends in social networks such as Twitter is offensive behavior that can cause the offended users leave their social network. Therefore, the need for effective automatic methods for identifying offensive language in textual data is important.

The OffensEval shared task has been organized in order to give a boost to computational methods for identifying and categorizing offensive content on social media. Three sub-tasks defined in the offensEval shared task are identification of offensive language(sub-task A), categorization of offense types(sub-task B) and identification of the offense target(sub-task C) (Zampieri et al., 2019b).

The main goal in sub-task A is to identify offensive tweets from non-offensive ones. By definition, a post is labeled as offensive if it contains any form of non-acceptable language (profanity) or a targeted offense, which can be veiled or direct.

This year, we participated in sub-task A. Our methods for this sub-task include two approaches. In the first approach, traditional machine learning methods, deep learning methods and also a combination method are employed for the task. In the second approach, a data augmentation method is proposed to improve the performance of the methods of the first approach.

## 2 Related Work

Offenseive language identification which is also known as aggression, cyberbullying, hate speech and abusive language has been widely studied in previous works(Davidson et al., 2017; Malmasi and Zampieri, 2017, 2018).

Based on a survey conducted by Fortuna and Nunes (2018), the majority of previous works are on English and the researchers mainly use machine learning for this task and most proposed methods on abusive content detection have modeled the problem as a binary classification task. Based on another survey (Schmidt and Wiegand, 2017), different types of features have been employed by previous works including surface features, word generalization features such as word embeddings, sentiment-based features, lexical features, linguistic features, knowledge-based features and multimodal information features. The methods utilized for offensive language identification are mainly supervised learning methods including SVM, Random Forest, Naive Bayes and also deep learning approaches (Gambäck and Sikdar, 2017) As an example, (Gambäck and Sikdar, 2017) proposed a model based on convolutional neural networks which takes word embedding vectors of a document as input and decides whether the document contains hate-speech content or not.



*Proceedings of the 13th International Workshop on Semantic Evaluation (SemEval-2019)*, pages 600–603
Minneapolis, Minnesota, USA, June 6–7, 2019. 

## 3 Methodology and Data

### 3.1 Datasets

The dataset used in this competition is available as part of the OffensEval 2019 Shared Task Zampieri et al. (2019a). The training set contains 13240 tweets and the test set contains 860 tweets. We have also employed two external datasets including TRAC-1 data (Kumar et al., 2018) and 50K tweets collected by (Founta et al., 2018) in our experiments.

### 3.2 Features

In our methods, we make use of the following features:

**Content-based Features** Tweet text contains words which are the most prominent features to convey feelings. Therefore, based on the content of each tweet, we extract the following features as content-based features: the number of mentions, the number of links, the number of hashtags, the average word length, the number of punctuation marks, the average sentence length(based on the number of words in a each sentence), the total number of words, the number of uppercase and the number of emoticons in each tweet.

**Sentiment-based Features** Usually hate speech has negative sentiment. Thus, using the sentiment information of tweets may improve the performance of our methods. We use three types of sentiment-based features including polarity, subjectivity and emotion. In order to find the emotion label, we trained a random forest classifier on an external dataset annotated for emotions and polarity which contains 40K tweets and 13 classes of emotion (such as happiness, sadness, and anger)[1].

**TF-IDF Features** TF-IDF is one of the most popular term-weighting approaches which shows the importance of a term in a document or a collection. We use this feature in combination with other features.

**Hate-based Feature** Hate-based dictionary is a lexicon that can be used to identify hate speech and offensive language (Davidson et al., 2017). We considered the number of hate words and the number of hate n-grams of length 1 to 4 as hate-based features. Hate-base lexicon is available at www.hatebase.org.

[1] https://www.crowdflower.com/

### 3.3 Methods

In this section, the methods employed by our team for sub-task A are explained. We used several methods including traditional machine learning methods such as SVM, Random Forest and Naive Bayes in additions to a deep learning method and a combination method. In addition to the methods mentioned above, we proposed an augmentation method in order to improve the performance of our methods.

#### 3.3.1 Traditional Machine Learning Methods

Traditional machine learning methods, in particular, supervised classification methods is known as the most effective approach for offensive language identification. Therefore, in our experiments we applied three classifiers including SVM, Naive-Bayes and Random Forest. Among the most recent methods in the literature, deep learning methods has shown to be an effective approach for offensive language detection. Hence, we employed CNN, as our deep learning solution.

#### 3.3.2 Combination Method

In this method, we employed majority voting rule to combine the results of our best performing systems on the training set. Precisely speaking, for each tweet we find the majority label of three systems which are SVM, CNN trained on over 50k + 13k tweets and another CNN which trained on 50k + 13k + 10k tweets. The results are shown in Table 1.

**CNN Architecture**: The word-level CNN model has 1D convolution layer with 150 filters and kernel size 6, dropout 0.2, cross entropy loss funtion and four dense layers with ReLU, tanh, sigmoid and softmax activation respectively.

#### 3.3.3 Data Augmentation Method

A common technique to enhance model generalization is data augmentation. In this method, we employed an external dataset containing 50K tweets labeled as hateful, aggressive, normal and spam, in two different ways as follows. In direct augmentation, we added all tweets to the training set such that the tweets labeled as hateful or aggressive are added as offensive and normal or spam labeled tweets as non-offensive.

In indirect augmentation, first of all, the average word embedding of each tweet in the training set is calculated. Then, the average of the embedding vectors in each class is calculated to

be used as the representative (or center) of offensive and non-offensive class. Finally, the average word embedding vector of each tweet in the external dataset is calculated and compared with the offensive and non-offensive representative vectors through cosine similarity computation between each tweet and two centers. We defined a threshold for labeling new tweets. If the absolute difference of the distances between tweet's vector and each of the class center is higher than the threshold, we assign tweet to the nearest class. Thus, the tweets of the external dataset are labeled as their most similar class and added to the training set. During the indirect augmentation process, we used word2vec pre-trained Google News model (GoogleNews-vectors-negative300) to calculate embedding vectors of tweets. The threshold is determined 0.03 by experiments.

## 4 Results

### 4.1 Models' Performance Evaluation

In this section, the performance of all methods explained in section 3.3 on the training set using 5-fold cross validation is reported. The results of all of the used models on the training set are shown in Table 1. According to table 1, SVM outperforms other two methods in terms of macro-F1. Comparing the results of SVM and CNN shows that these two methods have close performance on the training set.

| System | F1 (macro) | Accuracy |
|---|---|---|
| Naive Bayes | 0.54 (+/- 0.02) | 0.70 (+/- 0.02) |
| Random Forest | 0.64 (+/- 0.02) | 0.74 (+/- 0.01) |
| SVM | 0.68 (+/- 0.01) | 0.73 (+/- 0.01) |
| CNN | 0.67 | 0.74 |

Table 1: Results for all methods on the training set using 5-fold cross validation (the variance of the scores for each fold are shown in parentheses)

### 4.2 Features' Evaluation

In this section, the impact of using the features explained in section 3.2 on the performance of the SVM method is studied. The results are noted in Table 2.

We perform 5-fold cross-validation on the training set and report the results for SVM using different combinations of the feature sets. The first observation is that TF-IDF features outperform other three sets of features in the first section of table 2 which corresponds to using only one feature set. The combination of TFIDF and sentiment-based features, TFIDF and hate-based features and TFIDF, content-based and hate-based features equally show the best performance among all combinations.

### 4.3 Augmentation Method Evaluation

In this section, the impact of the augmentation method on the performance of our classifier is evaluated. Table 3 shows the results of SVM on the training set using 5-fold cross validation in three different settings; when no augmentation is done, when the external dataset is used directly and when the augmentation method (i.e. using the external data indirectly) is employed. As table 3 shows, the augmentation method produces the best results.

### 4.4 Results on the Test Set

In this section, the results of three systems that we submitted to OffensEval 2019 is reported. Table 4 shows the results of SVM using augmentation method with two external datasets (SVM-50k+13k), CNN using augmentation method with three external datasets (CNN-50k+13k+10k) and the majority voting method using the outputs of two mentioned methods on the test set. Furthermore, the results of two random baseline generated by assigning the same labels for all instances (all offensive and all non-offensive) are reported for comparison. According to the table, the combination of first two method using majority voting has the best performance.

| System | F1 (macro) | Accuracy |
|---|---|---|
| All NOT baseline | 0.4189 | 0.7209 |
| All OFF baseline | 0.2182 | 0.2790 |
| SVM-50k+13k | 0.7076 | 0.7884 |
| CNN-50k+13k+10k | 0.7155 | 0.7814 |
| Majority Vote | 0.7325 | 0.8186 |

Table 4: Results for Sub-task A.

## 5 Conclusion

In this paper, we address the challenge of automatically detecting offensive and non-offensive language in textual content spread in twitter. We conducted experiments with SVM with varying feature sets and CNN model. We also proposed an augmentation method to improve the performance our classifiers.

| System | F1 (macro) | Accuracy |
|---|---|---|
| TF-IDF features | 0.68 (+/- 0.02) | 0.74 (+/- 0.01) |
| Content features | 0.42 (+/- 0.01) | 0.73 (+/- 0.02) |
| Sentiment features | 0.50 (+/- 0.01) | 0.69 (+/- 0.01) |
| Hatebased features | 0.49 (+/- 0.01) | 0.69 (+/- 0.01) |
| TF-IDF + Content | 0.67 (+/- 0.03) | 0.73 (+/- 0.02) |
| TF-IDF + Sentiment | 0.71 (+/- 0.02) | 0.76 (+/- 0.01) |
| TF-IDF + Hatebased | 0.71 (+/- 0.02) | 0.76 (+/- 0.01) |
| Content + Sentiment | 0.54 (+/- 0.02) | 0.68 (+/- 0.01) |
| Content + Hatebased | 0.49 (+/- 0.01) | 0.68 (+/- 0.01) |
| Sentiment + Hatebased | 0.51 (+/- 0.04) | 0.70 (+/- 0.02) |
| TF-IDF + Content + Sentiment | 0.68 (+/- 0.02) | 0.73 (+/- 0.01) |
| TF-IDF + Content + Hatebased | 0.68 (+/- 0.03) | 0.74 (+/- 0.02) |
| TF-IDF + Content + Hatebased | 0.71 (+/- 0.02) | 0.76 (+/- 0.01) |
| Content + Sentiment + Hatebased | 0.56 (+/- 0.02) | 0.70 (+/- 0.01) |
| TF-IDF + Content + Sentiment + Hatebased | 0.68 (+/- 0.01) | 0.73 (+/- 0.01) |

Table 2: Results for SVM method using different sets of features on the training set using 5-fold cross validation (the variance of the scores for each fold are shown in parentheses)

| System | F1 (macro) | Accuracy |
|---|---|---|
| Without using the external dataset | 0.68 | 0.73 |
| With using the external dataset directly | 0.73 | 0.89 |
| With using the external dataset indirectly (augmentation method) | 0.76 | 0.88 |

Table 3: Results for SVM method on the training set without using the external dataset, with using the external dataset directly and with using the external dataset indirectly(the augmentation method)

## References


Thomas Davidson, Dana Warmsley, Michael Macy, and Ingmar Weber. 2017. Automated Hate Speech Detection and the Problem of Offensive Language. In *Proceedings of ICWSM*.

Paula Fortuna and Sérgio Nunes. 2018. A Survey on Automatic Detection of Hate Speech in Text. *ACM Computing Surveys (CSUR)*, 51(4):85.

Antigoni-Maria Founta, Constantinos Djouvas, Despoina Chatzakou, Ilias Leontiadis, Jeremy Blackburn, Gianluca Stringhini, Athena Vakali, Michael Sirivianos, and Nicolas Kourtellis. 2018. Large Scale Crowdsourcing and Characterization of Twitter Abusive Behavior. *arXiv preprint arXiv:1802.00393*.

Björn Gambäck and Utpal Kumar Sikdar. 2017. Using Convolutional Neural Networks to Classify Hate-speech. In *Proceedings of the First Workshop on Abusive Language Online*, pages 85–90.

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 Cyberbulling (TRAC)*, Santa Fe, USA.

Shervin Malmasi and Marcos Zampieri. 2017. Detecting Hate Speech in Social Media. In *Proceedings of the International Conference Recent Advances in Natural Language Processing (RANLP)*, pages 467–472.

Shervin Malmasi and Marcos Zampieri. 2018. Challenges in Discriminating Profanity from Hate Speech. *Journal of Experimental & Theoretical Artificial Intelligence*, 30:1–16.

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*, pages 1–10, Valencia, Spain.

Marcos Zampieri, Shervin Malmasi, Preslav Nakov, Sara Rosenthal, Noura Farra, and Ritesh Kumar. 2019a. Predicting the type and target of offensive posts in social media. In *Proceedings of NAACL*.

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