CyberTronics at SemEval-2020 Task 12: Multilingual Offensive Language Identification over Social Media

Sayanta Paul
Dept. of CSE
IIT Patna, India
1811cs16@iitp.ac.in

Sriparna Saha
Dept. of CSE
IIT Patna, India
sriparna@iitp.ac.in

Mohammed Hasanuzzaman
Dept. of CS
CIT Cork, Ireland
Mohammed.Hasanuzzaman@cit.ie

Abstract

The SemEval-2020 Task 12 (OffensEval) challenge focuses on detection of signs of offensiveness using posts or comments over social media. This task has been organized for several languages, e.g., Arabic, Danish, English, Greek and Turkish. It has featured three related sub-tasks for English language: sub-task A was to discriminate between offensive and non-offensive posts, the focus of sub-task B was on the type of offensive content in the post and finally, in sub-task C, proposed systems had to identify the target of the offensive posts. The corpus for each of the languages is developed using the posts and comments over *Twitter*, a popular social media platform. We have participated in this challenge and submitted results for different languages. The current work presents different machine learning and deep learning techniques and analyzes their performance for offensiveness prediction which involves various classifiers and feature engineering schemes. The experimental analysis on the training set shows that SVM using language specific pre-trained word embedding (*Fasttext*) outperforms the other methods. Our system achieves a macro-averaged F1 score of 0.45 for *Arabic* language, 0.43 for *Greek* language and 0.54 for *Turkish* language.

1 Introduction

Offensive language is very common in social media, now-a-days. Individual users frequently take advantage of the perceived anonymity of computer mediated communication, using this to engage in behavior that many of them would not consider in real life. The SemEval-2020 Task 12 (OffensEval) challenge focuses on prediction of presence of offensive language using the social media. The main goal of this task is to instigate discussion on the creation of reusable benchmarks for evaluating proposed algorithms by exploring issues of evaluation methodology and other processes related to the creation of test collections. The given corpora are developed using the posts and comments over *Twitter*, a popular social media. Zampieri et al. (2020) organized a multilingual offensive language classification task with a particular focus on Twitter posts. It has released different corpora for the individual languages, e.g., Arabic (Mubarak et al. (2020)), Danish (Sigurbergsson et al. (2020)), English (Rosenthal et al. (2020)), Greek (Pitenis et al. (2020a)) and Turkish (Coltekikin (2020)), and intended to identify and capture the offensive language. All the languages except English have only one task to be performed, e.g., sub-task A which is to identify offensive language. For English, the task has been divided in three different sub-tasks, namely, sub-task B for offense type categorization in which the offense type is categorized into either targeted or untargeted, and sub-task C focuses on identification of target offense.

In this paper, different machine learning and deep learning frameworks have been proposed to accomplish the given task. Each of the proposed systems has been implemented using language specific *fasttext*, pre-trained word vectors developed over crawling the web. Support Vector Machine (SVM) is widely used for text categorization as introduced by Tong et al. (2001). Fan et al. (2008) recommended the linear kernel for text categorization as it performs well when there exists a lot of features. Hence linear SVM has been used in our experiments. Convolutional Neural Network (CNN) had been invented by LeCun (1998)

Proceedings of the 14th International Workshop on Semantic Evaluation, pages 1925–1931
Barcelona, Spain (Online), December 12, 2020.

for extraction of local features, which later had been proven to be the standard choice for image processing tasks. Also, Hochreiter and Schmidhuber (1997) introduced Long Short Term Memory (LSTM) to capture implicit ordering of the sequence data in terms of words and sentences, which is widely used in various Natural Language Processing tasks. Subsequently, SVM as machine learning classifier and hybrid network of CNN with LSTM as deep learning classification system are proposed and those have been implemented for efficiently identifying the presence of offensive languages in text. The results on the test set submitted to the challenge suggest that these frameworks achieve reasonably good performance. However, there are some submissions for this task, whose performances are better than our proposed framework.

The paper is organised as follows. Related literature reviews have been provided in Section 2. The corpora used in these experiments is described in Section 3. The proposed machine learning and deep learning frameworks are explained in Section 4. Section 5 describes the experimental evaluation. The conclusion is presented in Section 6.

2 Related work

Identifying offensive language over social media has been an increasingly trending issue over the past few years. Fortuna et al. (2018) showed a survey of different text mining approaches for effectively identifying these issues. Each existing language contains it's own language rule which comprises of different syntactic and semantic guidelines. For this, to accomplish the same goal over different languages, it requires different methodologies and approaches.

2.1 Arabic language

Offensive language detection in Arabic language is bit challenging due to the lexical variations of different Arabic dialects. Mubarak (2017) proposed abusive language detection framework on Arabic social media. They have first extracted a list of offensive words and related hashtags using common patterns used in offensive and rude conversations and then classified Twitter users according to presence of these words or not in their tweets. Alakrot (2018a) presented a Arabic corpus collected and developed from YouTube comments to be used for the detection of offensive language. Correspondingly, authors have also presented a brief statistical analysis for predictive modelling. Alakrot (2018b) again introduced Support Vector Machine classifier with combinations of different features and a variety of preprocessing techniques in order to detect offensive language from Arabic text.

2.2 Greek language

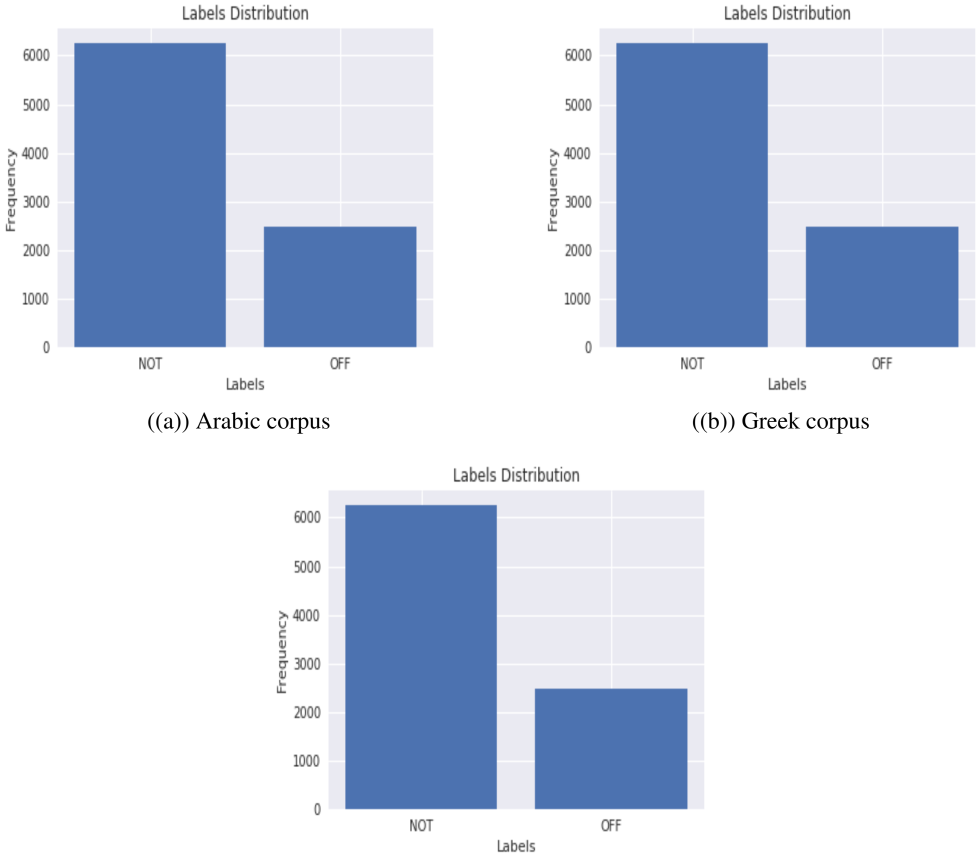
Greek language has a distinct writing system due to the Greek alphabet and an independent branch of the Indo-European family of languages. For this, there is a lack of available computational tools to analyze and process Greek language. Lekea et al. (2018) presented a methodology for automatically detecting the presence of hate speech within the Greek text. Again, Pitenis et al. (2020b) introduced the first Greek annotated corpus for offensive language detection. Authors have also shown a detailed data analysis.

2.3 Turkish language

Very small amount of research activities have been seen in several low-resource languages e.g., Turkish. S.A.Özel et al. (2017) introduced a text based approach for detecting cyberbullying from social media text. Authors have collected and developed corpus of this Turkish dataset from Instagram and Twitter messages written in Turkish. A few text classification algorithms have been used for predictive analysis.

3 Data Description

The corpora released as part of the SemEval 2020 (Zampieri et al.(2020)) are the collection of posts or comments from a set of users over Twitter. Each of the corpora is divided into two categories - Offensive(OFF) and Not-offensive(NOT). Figures 1 shows the distribution of the classes in the data provided for Arabic, Greek and Turkish languages, respectively. The distributions clearly show the imbalance in class labels. The overview of each of the corpora is presented in Table 1.



((a)) Arabic corpus

((b)) Greek corpus

((c)) Turkish corpus

Figure 1: Class distribution over different corpus

Language	**Training Set** NOT	OFF	**Test Set** NOT	OFF	**Total**
Arabic	5590	1410	1598	402	9000
Greek	6257	2486	1302	242	10287
Turkish	25625	6131	2812	716	35284

Table 1: Dataset statistics

4 Proposed Methodologies

The given corpus for each of the languages of SemEval-2020 Task 12 (Zampieri et al.(2020)) is further divided into two sets, namely, training set and validation set. The new training set is developed by randomly choosing 80% tweet samples from NOT and OFF categories; the rest 20% of these categories form the validation set. To train our models, support vector machine - a classical machine learning text classification algorithm is utilized. As a deep learning model, the hybrid model integrating networks of CNN with LSTM has been trained for accomplishing the task. Our proposed framework consists of three crucial layers: data preprocessing, feature extraction and text categorization. A high-level overview of our proposed framework and steps of producing clean text from raw text is shown in Figure 2. We have kept this architecture uniform for offensive language detection for all the three languages.

4.1 Data Preprocessing

In order to gain meaningful information from the available text data, it is important to remove noise from it to improve its quality before analyzing it. The steps of data preprocessing involve removal of stopwords, unnecessary URLs and punctuations and Twitter mentions. As the part of data normalization, we have

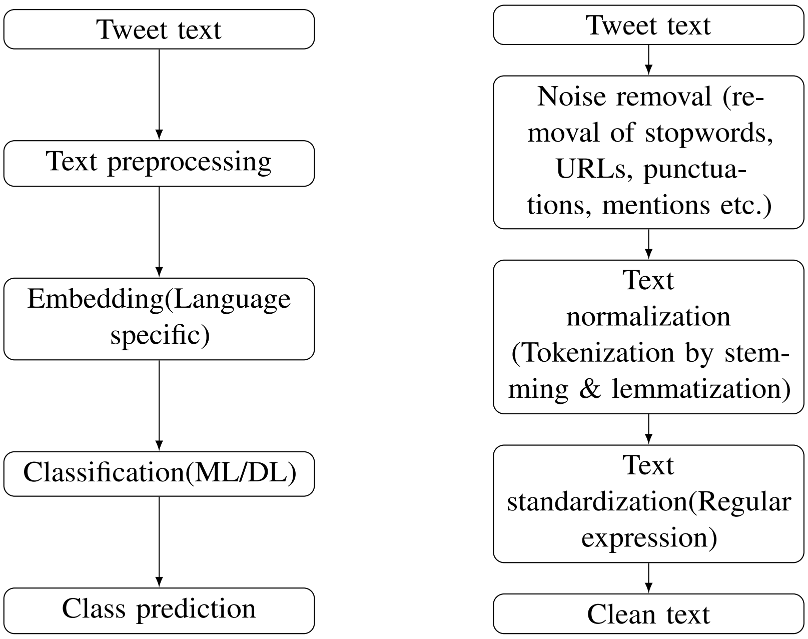

Tweet text
Text preprocessing
Embedding(Language specific)
Classification(ML/DL)
Class prediction
Tweet text
Noise removal (removal of stopwords, URLs, punctuations, mentions etc.)
Text normalization (Tokenization by stemming & lemmatization)
Text standardization(Regular expression)
Clean text

Figure 2: Overview of our proposed framework (left) & steps of data preprocessing

also performed Stemming and Lemmatization using Natural Language Tool Kit (NLTK) [1]. At the end of the step, clean text is generated.

4.2 Feature Engineering

Our proposed approach is entirely based on text features. It can be found in the literature that different linguistic features and quantitative features contribute significantly for fine-grained text classification, as described by Argamon et al.(2007) and HaCohen et al.(2010). Any classical machine learning or deep learning model can not take raw text as its input. Therefore we convert these clean text into the corresponding numerical feature vectors. We have used language specific word embedding models of embedding dimension 300 for each of the language corpora along with TF-IDF/Count models. The language specific embeddings are trained on Common Crawl and Wikipedia data using fastText [2]. These models were trained using CBOW (as described in Mikolov et al. (2013)) with position-weights, in dimension 300, with character n-grams of length 5 as described in Grave et al. (2018). In our case, a vector of a word is being predicted based on context words. For example, we want to predict the vector of a particular word w_0 based on its context words $w_{-n}, ..., w_{-1}, w_1, ..., w_n$. A vector representation h of this context is obtained by considering the average of the corresponding word vectors, which can be defined as:

$$h = \sum_{i=-n;i\neq 0}^{n} u_{w_i} \tag{1}$$

4.3 Classification

The classical machine learning model has been implemented using scikit-learn [3] and deep learning model has been implemented in Keras [4] on top of Tensorflow. We then fed the produced word vectors to the classifiers. As ML classifier, we have used SVM. As DL framework, we have proposed a hybrid framework of CNN with LSTM in order to identify offensive instances in twitter text. For SVM, the optimal set of parameters are as follows: regularization parameter (*C*) as 0.5, *class_weight* kept as *balanced* as our corpora is highly imbalanced, and *kernel=linear*. Similarly, for deep learning framework, the best set of

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

[2] https://fasttext.cc/

[3] https://scikit-learn.org/stable/

[4] https://keras.io/

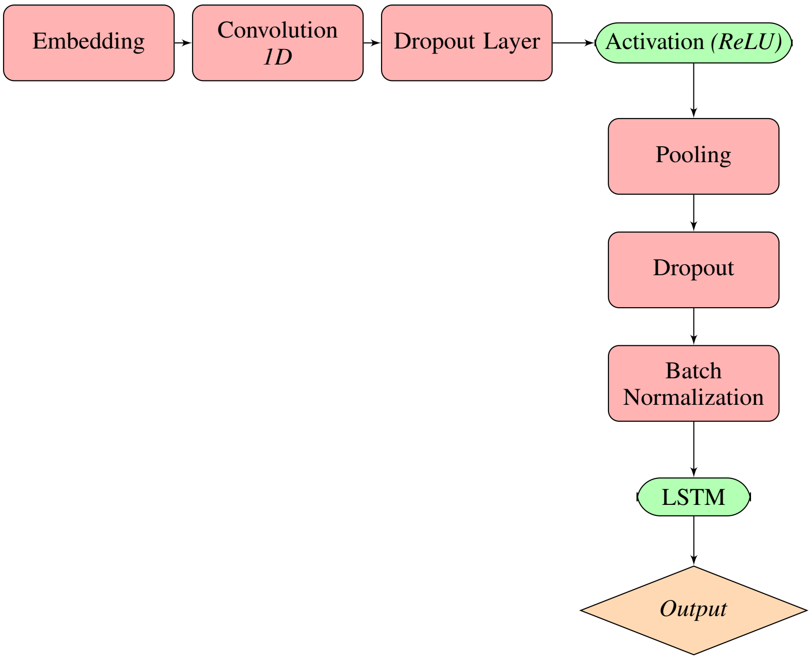


Figure 3: Diagram of CNN based LSTM network

hyperparameters are: batch_size is 16, dropout probability is 0.5 and the models are trained on 20 epochs. In order to find the optimal set of hyparparameter, we have used Bayesian optimization technique. The overview of our proposed deep neural network can be seen in Figure 3.

5 Experimental Results

We have reported the performance of both the machine learning and deep learning frameworks on our validation set for each of the corpora in Table 2, Table 3, Table 4, respectively, for Arabic, Greek and Turkish languages. It can be noted that the performance of these classifiers are measured in terms of F1 score. These results are useful in analyzing the performance of different proposed frameworks and subsequently the results are communicated. Classical machine learning model - Support Vector machine(SVM) outperforms the deep learning model deployed. Figure 4, 5 and 6 present the confusion matrices of our submission for Sub-task A for Arabic, Greek and Turkish languages, respectively.

Text Classifiers	Precision	Recall	F1 score
SVM	0.48	0.60	**0.54**
CNN+LSTM	0.49	0.46	0.48

Table 2: Performance of our proposed framework for **Arabic** language

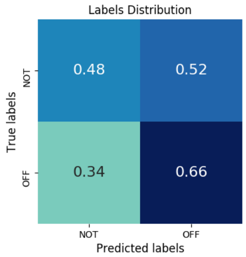


Figure 4: Confusion matrix for Arabic language

The final results of the challenges are presented in Table 5. It can be observed that the macro F1 score obtained for the test set is less when predicted with the training corpora. This may happen due to class imbalance problem which leads to poor generalization capability of classifiers.

6 Conclusion

In this paper, we have reported our proposed frameworks and their corresponding performances in Sub-task A of SemEval-2020 Task 12 OffensEval, for the languages of Arabic, Greek and Turkish: Multilingual

Text Classifiers	Precision	Recall	F1 score
SVM	0.59	0.60	**0.49**
CNN+LSTM	0.49	0.42	0.45

Table 3: Performance of our proposed framework for **Greek** language

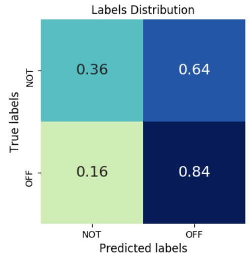


Figure 5: Confusion matrix for Greek language

Text Classifiers	Precision	Recall	F1 score
SVM	0.54	0.77	**0.64**
CNN+LSTM	0.40	0.50	0.44

Table 4: Performance of our proposed framework for **Turkish** language

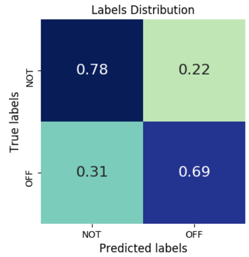


Figure 6: Confusion matrix for Turkish language

Language	F1 score
Arabic	0.45
Greek	0.43
Turkish	0.54

Table 5: Performances of our framework on test set

Offensive Language Identification in Social Media (OffensEval 2020). We have learned that our proposed frameworks did not perform well because of two possible issues which are as follows: firstly, all the language corpora are highly imbalanced, i.e., insufficient quantity of tweet samples in either of the classes and secondly, for the low resource language, e.g., Greek, the qualities of produced word vectors are much lower than those of other resourceful languages. We understand that an increase in the volume of data could add more potential to our framework, and use of external resources of data could be advantageous.

References

Azalden Alakrot, Liam Murray, and Nikola S Nikolov. 2018a. Dataset construction for the detection of anti-social behaviour in online communication in arabic. *Procedia Computer Science*, 142:174–181.

Azalden Alakrot, Liam Murray, and Nikola S Nikolov. 2018b. Towards accurate detection of offensive language in online communication in arabic. *Procedia computer science*, 142:315–320.

Shlomo Argamon, Casey Whitelaw, Paul Chase, Sobhan Raj Hota, Navendu Garg, and Shlomo Levitan. 2007. Stylistic text classification using functional lexical features. *Journal of the American Society for Information Science and Technology*, 58(6):802–822.

Çağrı Çöltekin. 2020. A Corpus of Turkish Offensive Language on Social Media. In *Proceedings of the 12th International Conference on Language Resources and Evaluation*, pages 6174–6184. ELRA.

Rong-En Fan, Kai-Wei Chang, Cho-Jui Hsieh, Xiang-Rui Wang, and Chih-Jen Lin. 2008. Liblinear: A library for large linear classification. *Journal of machine learning research*, 9(Aug):1871–1874.

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

Edouard Grave, Piotr Bojanowski, Prakhar Gupta, Armand Joulin, and Tomas Mikolov. 2018. Learning word vectors for 157 languages. In *Proceedings of the International Conference on Language Resources and Evaluation (LREC 2018)*.

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

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation*, 9(8):1735–1780.

Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. 1998. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324.

Ioanna K Lekea and Panagiotis Karampelas. 2018. Detecting hate speech within the terrorist argument: a greek case. In *2018 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM)*, pages 1084–1091. IEEE.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119.

Hamdy Mubarak, Kareem Darwish, and Walid Magdy. 2017. Abusive language detection on arabic social media. In *Proceedings of the First Workshop on Abusive Language Online*, pages 52–56.

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*.

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

Zeses Pitenis, Marcos Zampieri, and Tharindu Ranasinghe. 2020b. Offensive language identification in greek. *arXiv preprint arXiv:2003.07459*.

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

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.

Simon Tong and Daphne Koller. 2001. Support vector machine active learning with applications to text classification. *Journal of machine learning research*, 2(Nov):45–66.

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*.

S. A. Özel, E. Saraç, S. Akdemir, and H. Aksu. 2017. Detection of cyberbullying on social media messages in turkish. In *2017 International Conference on Computer Science and Engineering (UBMK)*, pages 366–370.