SU-NLP at SemEval-2020 Task 12: Offensive Language Identification in Turkish Tweets

Anıl Özdemir
Sabancı University
İstanbul, Turkey
aozdemir@sabanciuniv.edu

Reyyan Yeniterzi
Sabancı University
İstanbul, Turkey
reyyan@sabanciuniv.edu

Abstract

This paper summarizes our group's efforts in the offensive language identification shared task, which is organized as part of the International Workshop on Semantic Evaluation (Sem-Eval2020). Our final submission system is an ensemble of three different models, (1) CNN-LSTM, (2) BiLSTM-Attention and (3) BERT. Word embeddings, which were pre-trained on tweets, are used while training the first two models. BERTurk, which is the first BERT model for Turkish, is also explored. Our final submitted approach ranked as the second best model in the Turkish sub-task.

1 Introduction

With the growing popularity of social media all over the world, more and more people are using these platforms, and unfortunately, many people regularly face offensive language in these environments. Such offensive language can have devastating effects on users depending on their ages and psychological conditions (O'Keeffe et al., 2011). In the current big data era, manual filtering of such content is not possible; therefore, development of effective automated solutions is a necessity.

This important task of identifying the offensive language has been supported by SemEval organizers since last year, as part of the OffensEval task (Zampieri et al., 2019). Last year's task focused only on English and this year on its second run, the organizers aimed for a multilingual task and run the challenge for five languages, Arabic, Danish, English, Greek and Turkish (Zampieri et al., 2020). This year we specifically focused on Turkish, in which a high-performance automated system is very much needed, due to the widespread use of social media by Turkish-speaking people of all ages.

Within the framework of this problem, we initially analyzed different state-of-the-art neural network models and different word embeddings. Our final system consists of a weighted ensemble model combining the following 3 models; Convolutional Neural Network (CNN) with Long Short Term Memory (LSTM), Bidirectional Long Short Term Memory (BiLSTM) with Attention and the recently released Turkish BERT model. Additionally, word embeddings which are trained over Turkish tweets are explored for better domain and task adaptation, and for this particular task they provide much better performance compared to other available pre-trained embeddings. With our proposed system we achieved the 2nd place in Turkish sub-task. Our code is available online[1].

2 Task and Data Description

OffensEval task consists of three subtasks. This year for the four new languages, the organizers organize only the first sub-task A, which is the offensive language identification task. The goal of this sub-task is to predict whether a given content is offensive or not. In the task, a post is described as offensive if it includes insults, threats, profane language or swear words (Zampieri et al., 2019).

The data collection created by Çöltekin (2020), is used for OffensEval Turkish task. The collection consists of 34805 Turkish tweets which were annotated with the following labels:

- Offensive (OFF): Tweets that contain offensive language.

[1]https://github.com/SU-NLP/Multilingual-Offensive-Language-Identification-in-Social-Media

Proceedings of the 14th International Workshop on Semantic Evaluation, pages 2171–2176
Barcelona, Spain (Online), December 12, 2020.

- Not Offensive (NOT): Tweets that do not contain any form of offense.

This data collection is divided into two splits, as one is used for training and the other for test. The distribution of the labels after the split is presented in Table 1.

Label	**Training**	**Test**
OFF	6046	716
NOT	25231	2822
ALL	31277	3528

Table 1: The distribution of the labels in the data collection

3 Experimental Setup

Initially, common and Twitter specific pre-processing steps are applied to the data. These steps include converting all text to lowercase, removing punctuations, removing symbols like hashtags and Twitter specific tokens like @user. After this preprocessing, tweets are tokenized by the NLTK tokenizer.

During the development phase, we are only provided with the training data. Therefore, in order to evaluate the performance of the proposed approaches, we split the training data into two. 80% of the data is used for training and 20% for validation.

The provided training data collection is imbalanced as the size of the not-offensive tweets is around four times the size of offensive tweets. Machine learning algorithms may perform poorly with such imbalanced data collections. In order to handle this problem, we resample the training data by downsampling it. Random samples are chosen from the not-offensive observations. The number of non-offensive tweets in the training split is reduced by 40%. The average F1-score from different models, is used to choose this percentage value. Size of the validation set and training set before and after this downsampling is shown in Table 2.

Label	Training Before Downsampling	Training After Downsampling	Validation
OFF	4852	4852	1194
NOT	20174	11999	5057
ALL	25026	16851	6251

Table 2: Label distributions after re-sampling and train/validation split.

In order to analyze the effects of downsampling, we train two models for each neural network architecture, one with downsampled data and another one with all training data. Even after downsampling, the data collection is still imbalanced. Therefore, we use macro-averaged F1-score to compare the models in addition to the accuracy metric. F1-score is also the official evaluation measure for this task.

4 System Overview

Different neural network architectures are explored for this classification problem. Additionally, several pre-trained word embeddings which are trained on different collections, are used with these networks. Finally, the recently released pre-trained BERT model for Turkish is applied.

The following neural network architectures are explored for the task.

- **CNN-LSTM:** CNNs are good at detecting useful local features for classification tasks, especially when they are initialized with strong pre-trained word embeddings (Kim, 2014). On the other hand, LSTMs (Hochreiter and Schmidhuber, 1997) are good at keeping long distance dependencies within the text. Both of these skills can be useful in identification of offensive context in a text; therefore, our first architecture is a CNN-LSTM network. This network consists of two convolutional layers, one LSTM layer, one max-pooling layer and a fully connected layer.

- **BiLSTM-Attention:** As our second model, we explore a combination of BiLSTM model and attention mechanism. In BiLSTM models both past and future context is being used when processing any word. Attention mechanism is also useful for weighting words based on their importance for the task (Bahdanau et al., 2014). This network consist of a BiLSTM layer, attention layer and fully connected layer. The attention model used in this network was adopted from Raffel and Ellis (2015).

For all these models, all layers except for the embedding layer is trained from scratch. Pre-trained word embeddings are used for these models and the embedding layer is freezed during training in order to prevent possible overfitting.

Word Embeddings: Several pre-trained word embeddings are available for Turkish. In this particular task, publicly available Word2Vec (Mikolov et al., 2013a) embeddings[2] trained on 4.76 Million Turkish news articles (Erdinç and Güran, 2019) are used for the experiments. Erdinç and Güran (2019) trained several models with different pre-processing steps and compared their performances over various text classification tasks. In this work, we use the one trained after removing punctuation and stopwords, which is named as NDA. In their reported experiments, this particular embedding outperformed the others.

Embeddings trained on news articles is definitely useful for many classification tasks. However, in this particular task, the effects of such an embedding can be limited. Tweets are very different than news articles in terms of their writing styles and the used vocabulary. Furthermore, news articles do not contain profane language or swear words which are very important signals for identifying offensive language. Due to these reasons, an additional word embedding model is trained over tweets only. More than 24 Million tweets which were collected between June 2019 and February 2020 are used to train word embeddings. Continuous bag-of-words method (Mikolov et al., 2013b) is used to train 300 dimensional vectors. During the training, a window of 10 and minimum count of 5 is used as the hyperparameters.

BERT: In addition to the above architectures which were trained from scratch, pre-trained models are also explored to get a head start on the task. A pre-trained BERT model is fine-tuned for this task. BERT (Devlin et al., 2018), which is a transformer (Vaswani et al., 2017) based model, is capable of capturing the bidirectional representations of texts by jointly conditioning on both the left and the right context.

We use the publicly available BERTurk-Base cased model[3] which was trained over a combination of Turkish data collections like OSCAR corpus[4], Wikipedia and OPUS[5] corpora. BERTurk model, which consists of 12 transformer layers, is fine-tuned for offensive language identification task by using the provided training data.

5 Experiments

The described CNN-LSTM and BiLSTM-Attention models are trained from scratch by using either the word embeddings trained on news articles or our own embeddings trained on tweets. Additionally, BERTurk model is fine-tuned for the particular task. In order to analyze the effects of downsampling, all models are trained twice, one with the downsampled data and another one with the whole data. Later on, these models are applied to the validation data split and the results are presented in Table 3.

In Table 3, F1 Score and Accuracy of different architectures with different word embeddings and with or without downsampling are summarized. According to the results, models using word embeddings trained on tweets consistently outperform models using embeddings trained on news articles. This expected result is due to the domain similarity of the data collections used for training the word embeddings and training the final prediction model.

Even though using word embeddings trained on data originated from Twitter seems to be useful, BERTurk's results suggest that the architecture and how much information is being learnt and transferred, are still very much important. BERTurk was not trained on tweets but due to its better modeling of the context, it outperforms all other approaches.

[2]https://github.com/hakkiyagiz/SIU2019
[3]https://github.com/stefan-it/turkish-bert
[4]https://traces1.inria.fr/oscar/
[5]http://opus.nlpl.eu/

Model	Embedding	w/DS		w/oDS	
		F1-Score	Accuracy	F1-Score	Accuracy
CNN-LSTM	News W2V	0.731	0.850	0.711	0.856
	Twitter W2V	0.751	0.867	0.742	0.855
BiLSTM-Attention	News W2V	0.707	0.827	0.679	0.847
	Twitter W2V	0.748	0.870	0.763	0.859
BERTurk	-	0.789	0.866	0.814	0.888

Table 3: Results on Validation Data (w/DS: with Downsampling, w/oDS: without Downsampling)

According to Table 3, there is not a clear winner between training with downsampled data or with whole data. Even though they both return similar results, still with the downsampled data, risks which can arise due to the data imbalance problem, are lower. Therefore, we continue using the downsampled data for modeling.

5.1 Ensemble Models

In order to analyze how these individual systems perform with respect to each other, and whether they make the same errors or not, model prediction comparison chart is presented in Figure 1. This chart shows the actual and predicted labels of the observations in the validation data. The figure consists of two parts, the one on the left represents all data, while the one on the right focuses on the middle part of the chart on left, where models return different outputs with respect to each other or the gold standard label.

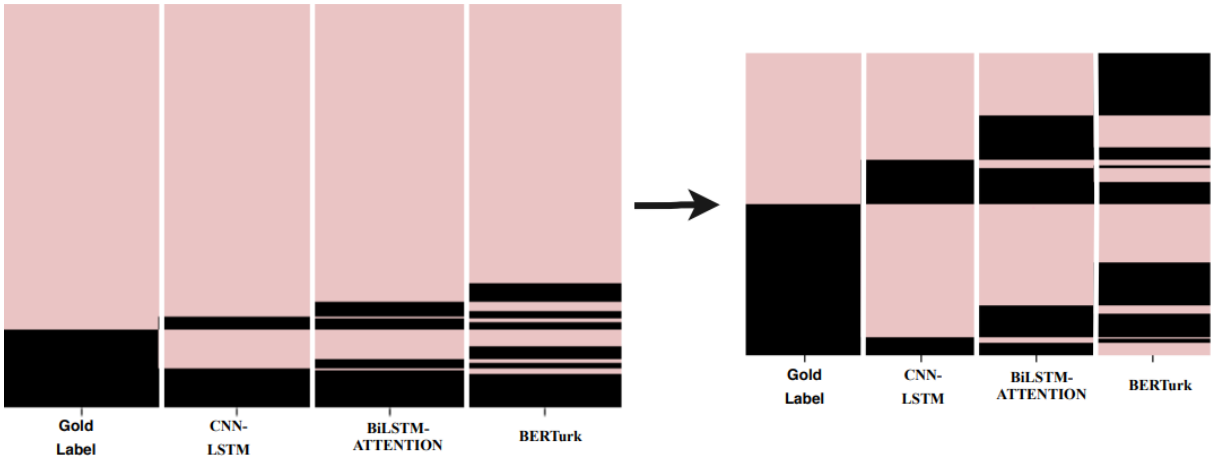

Figure 1: Individual Model Prediction Comparison Chart

In Figure 1, the regions in black represent offensive tweets, while the lighter parts represent non-offensive tweets. Validation data consists of 6251 tweets. For 373 of those, all three models predicted the same incorrect label, and for 4466 tweets all three models predicted correctly. For the rest of the 1412 tweets, three models did not return the same results. That part is presented in the right chart in more detail.

In order to improve the prediction performance specifically for those 1412 tweets, ensemble models are explored. The power of these three models, CNN-LSTM, BiLSTM with Attention and BERTurk, are combined using a weighted voting method. Several coefficients are experimented in order to find the optimum one. At the end, a weighting scheme which is based on BERTurk's prediction probability is used. Since BERTurk outperforms the other two models, during ensembling we use BERTurk as the main model and only use the prediction outcome of other models when BERTurk's prediction confidence is not high enough. The following rules are used to combine these models.

- Use only BERTurk if its prediction probability is higher than 0.85 or lower than 0.15. We trust BERTurk's performance on these highly confident predictions.

- If BERTurk's probability is in the range of $0.15 - 0.30$ or $0.70 - 0.85$, then use the following formula $0.6\times$BERTurk$+0.2\times$CNN-LSTM$+0.2\times$BiLSTMAttention

- If BERTurk's probability is in the range of $0.30 - 0.70$, then use the following formula $0.4\times$BERTurk$+0.3\times$CNN-LSTM$+0.3\times$BiLSTMAttention

This ensemble approach improved the performance across all metrics and experimental settings as shown in Table 4. This table presents the results on both validation split and test data. For the CNN-LSTM and BiLSTM-Attention model, Twitter W2V vectors are used.

Model	Validation				Test			
	w/DS		w/oDS		w/DS		w/oDS	
	F1	Acc	F1	Acc	F1	Acc	F1	Acc
CNN-LSTM	0.751	0.867	0.742	0.855	0.773	0.865	0.766	0.863
BiLSTM-Attention	0.748	0.870	0.763	0.859	0.760	0.868	0.781	0.871
BERTurk	0.789	0.866	0.814	0.888	0.808	0.873	0.806	0.877
Ensemble	0.809	0.881	0.822	0.896	**0.816**	**0.883**	0.813	0.887

Table 4: Results on Validation and Test Data (w/DS: with Downsampling, w/oDS: without Downsampling, Twitter W2V is used for the first two models)

According to Table 4, combining the power of three models provides consistent improvements across different settings. Even though using the whole training data for the ensemble model provides better results compared to the downsampled data on the validation set, our choice of staying safe and using the downsampled data in our final submission paid of with slightly higher F1 Score over the test data. The scores in bold belong to our submission which ranked in the second place.

Our weighted voting ensemble approach depends on BERTurk's output. BERTurk's confidence level is being used whether to consider other models' outputs or not. Analysis over 6251 instances of the validation set shows that in 4351 of them, BERTurk returns an output with either more than 0.85 probability or lower than 0.15. Among these 4351 instances, BERTurk has correctly predicted the outcome in 4128 of the instances, around 95% of them. This high percentage shows BERTurk's learning capacity. When it is very confident it is more likely to be right as well.

With the rest of the 1900 instances where weighted voting has been applied, only the output prediction of 324 instances were modified from BERTurk's prediction and 220 of these modifications resulted in correct predictions. Examples where BERTurk returned the wrong prediction while the ensemble method corrected, are analyzed in detail. For example in *"dm kutum senin beyninden daha boş"* case, BERTurk's output is not offensive, but it was in fact offensive and corrected during the weighted voting. In another example, *"kafamda saçma senaryolar kurup moralimi çok güzel bozarım"*, BERTurk predicted as offensive while the other CNN-LSTM and BiLSTM-Attention models, which used embeddings trained on tweets, predicted non-offensive with high confidence. One reason why BERTurk cannot perform as good as other simpler models in these cases, is that it was trained on formal data collections, and therefore does not know enough about the informal language used in Twitter or even the Twitter specific terminology.

6 Conclusion

Our proposed ensemble model of BERTurk, CNN-LSTM and BiLSTM-Attention provided to be effective in the identification of offensive language in Turkish tweets. It has been observed that, word embeddings trained on tweets has positive effect on the overall performance of the prediction model. Based on this observation, a future work will be to pre-train a BERT model over large set of tweets and then fine-tune it for this task.

References

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*.

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.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Hakkı Yağız Erdinç and Aysun Güran. 2019. Semi-supervised turkish text categorization with word2vec, doc2vec and fasttext algorithms. In *2019 27th Signal Processing and Communications Applications Conference (SIU)*, pages 1–4. IEEE.

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

Yoon Kim. 2014. Convolutional neural networks for sentence classification. *arXiv preprint arXiv:1408.5882*.

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

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

Gwenn Schurgin O'Keeffe, Kathleen Clarke-Pearson, et al. 2011. The impact of social media on children, adolescents, and families. *Pediatrics*, 127(4):800–804.

Colin Raffel and Daniel PW Ellis. 2015. Feed-forward networks with attention can solve some long-term memory problems. *arXiv preprint arXiv:1512.08756*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008.

Marcos Zampieri, Shervin Malmasi, Preslav Nakov, Sara Rosenthal, Noura Farra, and Ritesh Kumar. 2019. Semeval-2019 task 6: Identifying and categorizing offensive language in social media (offenseval). *arXiv preprint arXiv:1903.08983*.

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*.