Sonal.kumari at SemEval-2020 Task 12: Social Media Multilingual Offensive Text Identification and Categorization using Neural Network Models

Sonal Kumari
Samsung Research Institute, Bangalore, India
sonal.kumari@samsung.com

Abstract

In this paper, we present our approaches and results for SemEval-2020 Task 12, Multilingual Offensive Language Identification in Social Media (OffensEval 2020). The OffensEval 2020 had three subtasks: A) Identifying the tweets to be offensive (OFF) or non-offensive (NOT) for Arabic, Danish, English, Greek, and Turkish languages, B) Detecting if the offensive tweet is targeted (TIN) or untargeted (UNT) for the English language, and C) Categorizing the offensive targeted tweets into three classes, namely: individual (IND), Group (GRP), or Other (OTH) for the English language. We participate in all the subtasks A, B, and C. In our solution, first we use the pre-trained BERT model for all subtasks, A, B, and C and then we apply the BiLSTM model with attention mechanism (Attn-BiLSTM) for the same. Our result demonstrates that the pre-trained model is not giving good results for all types of languages and is compute and memory intensive whereas the Attn-BiLSTM model is fast and gives good accuracy with fewer resources. The Attn-BiLSTM model is giving better accuracy for Arabic and Greek where the pre-trained model is not able to capture the complete context of these languages due to lower vocab-size.

1 Introduction

The usage of offensive language in social media is very common nowadays. Sometimes it is used for fun and entertainment purposes, but sometimes it is an expression of user's aggression, hate, and/or offensive behavior. Identification of user's intention with an offensive content on social media requires domain-specific sentiment analysis at fined-grained level of granularity. To control the offensive behavior of the user's post on social media requires the categorization of hate speech problems into new notions like abusive, aggressive, and/or offensive hate speech. Such categorization helps law-enforcement agencies for the surveillance of social media.

Identification of hate, aggression, or offensive speech in user-generated content has attracted significant attention in the sentiment analysis research area recently. As evidenced, in recent publications (Waseem et al. 2016; Davidson et al., 2017, Malmasi and Zampieri, 2018, Kumar et al. 2018) and workshops such as AWL and TRAC and competitions such as HatEval 2019 (Basile et al. 2019), HASOC 2019, and OffensEval 2019 (Zampieri et al. 2019).

The exponential rise of users on social media and their posts on social media led to an enormous amount of data generation. Detection of hate, offensive speech, aggression, or abusive content on social media requires the involvement of algorithms to make decisions based on deep learning models such as LSTM, Bidirectional LSTM, pre-trained BERT (Bidirectional Encoder Representations from Transformers), CNN, or Stacked CNN, but not limited.

The OffensEval-2020 has been introduced as a Multilingual Offensive Language Identification by Zampieri et al. (2020) in which following 3 subtasks were given: (A) Offensive Language Identification

Proceedings of the 14th International Workshop on Semantic Evaluation, pages 2146–2154
Barcelona, Spain (Online), December 12, 2020.

for Arabic, Danish, English, Greek and Turkish, (B) Automatic Categorization of Offensive Type for English only, and (C) Offensive Target Identification for English only.

We participate in all the subtasks A, B, and C. Our approach is based on a pre-trained BERT model (Devlin et al., 2018) and Attention aware BiLSTM model (Att-BiLSTM) (Zhou et al., 2016). First, we clean the given tweet data and fine-tune the pre-trained BERT model for all the subtasks, A, B, and C. Next, we apply the BiLSTM model for the same. We observe that the BiLSTM model accuracy is better compare to the BERT model Arabic-A and Greek-A, and also, the BERT model is compute-intensive and memory-hungry compare to the BiLSTM model.

The rest of this paper is organized as follows. In Section 2, related work has been presented. Section 3 and Section 4 give the Methodology and Experimental analysis. Finally, Section 5 concludes the work.

2 Related Work

Recently many research works (Schmidt and Wiegand, 2017; Malmasi and Zampieri, 2017; Gamback and Sikdar, 2017; Lee et al., 2018; Fortuna and Nunes, 2018; Zhang et al. 2018; Basile et al., 2019; Liu et al., 2019) have been done for Artificial Intelligence-based offensive language detection in English text. These works can be classified into following four categories: 1) Convolutional Neural Network (CNN) based (Gamback and Sikdar, 2017; Seo et al. 2020), 2) Recurrent Neural Network (RNN) based (Lee et al., 2018; Seo et al. 2020), 3) Combination of CNN & RNN (Zhang et al. 2018), and 4) other approaches (Malmasi and Zampieri, 2017; Basile et al., 2019; Liu et al., 2019). Liu et al. (2019) applied a pre-trained BERT model (Devlin et al., 2018) to fine-tune the binary offensive language detection task and demonstrated outstanding accuracy. Lee et al. (2018) demonstrated that a bidirectional Gated Recurrent unit network applied to word-level features generated from Hate Speech Detection data having four categories of class-label outperforms. Seo et al. (2020) evaluated three CNN based and five RNN based deep learning models on 13 review datasets for both word-level and character-level input structures. They demonstrated that BiLSTM based model with word-level performs best across various benchmarked review datasets of the English language.

For multilingual abusive language detection, less work has been done which focuses Hindi, & English in (Kumar et al. 2018), English & Spanish in (Basile et al., 2019), English, Italian, & German in (Corazza et al., 2020), Danish & English in (Sigurbergsson and Derczynski, 2020). Some research has been done targeting specific foreign languages such as Arabic (Alomari et al., 2017), Dutch (Tulkens et al. 2016; Van Hee et al. 2018), German (Wiegand et al., 2018), Greek (Pitenis et al. 2020), Italian (Bosco et al., 2018), Spanish (Carmona et al., 2018), etc. Despite lots of research done for abusive language identification, most of these works focused on English Language and the rest of the languages are still not much explored in this field.

Moreover, the RNN model is a default choice for most of the text analytics-related applications as it outperforms over rest but its recurrent structure creates hurdle in learning long-term dependency because of vanishing or exploding gradient problem. Hochreiter and Schmidhuber (1997) introduced LSTM to avoid vanishing gradient problem and learn long-term dependency in the sequence model by inserting a gate Unit. BiLSTM with Attention (Zhou et al., 2016) has been proposed for relation classification which has been shown to outperform SVM, CNN, RNN, and LSTM models. From the literature, it is clear that the BERT model and the BiLSTM based model outperform the rest of the AI models for sentiment analysis (Lee et al., 2018; Liu et al., 2019; Seo et al., 2020).

3 Methodology

Given data sets for sub-task A, B, and C, contain user's tweet messages where graphical and non-graphical emoji are used and have grammatical errors. First, we perform text pre-processing to clean and normalize the text messages. In the pre-processing phase for all data sets, Emoji are replaced with equivalent English phrases, hashtags (single words) are separated into multiple words using heuristic techniques, all texts are lower-cased, and duplicate words, numerical digits, & punctuations are removed. For text normalization, we remove the repeated words from all the datasets. We delete the repeating consecutive characters, appearing more than two times, from the English dataset. For example, the words "Hhhhhaaaaaaatttteeeee", "lmaoooooo", "Wholesomeeeeee", "HEEEEEELLLLLPPPPPP", and

“Looool”, are substituted with the words “Hate”, “lmao”, “Wholesome”, “HELP”, and “Lol”, respectively. We also delete single character words from Danish, English, and Turkish datasets. Arabic text is normalized by removing diacritics, replacing the letters ‘آ’, ‘إ’ &, ‘أ’ with the letter ‘ا’, and replacing the letters ‘ى’, ‘ؤ’, ‘ئ’, ‘ة’, ‘گ’ with the letters ‘ي’, ‘ء’, ‘ء’, ‘ه’, ‘ك’ respectively. For Greek text normalization, letters ‘ά’, ‘έ’, ‘ή’, ‘ό’, ‘ώ’, are substituted with letters 'α', ‘ε’, ‘η’, ‘ο’, ‘ω’ respectively, the letters ‘ί’, ‘ΐ’, & ‘ϊ’ are replaced with the letter ‘ι’, and the letters ‘ύ’, ‘ΰ’, & ‘ϋ’ are replaced with the letter ‘υ’. For Turkish text normalization, letters ‘î’, ‘â’, ‘ö’, ‘ü’, ‘ğ’, ‘ç’, and ‘ş’ are converted into letters ‘i’, ‘a’, ‘o’, ‘u’, ‘g’, ‘c’, and ‘s’, respectively.

In the next step, we statistically analyze the given data sets and observe the presence of class imbalance in the given datasets. Also, the selection of an optimal threshold for mapping average confidence scores is not straight forward because of the high deviation in confidence scores obtained by different supervised models. To handle this, at the first level, we filter out the samples having a very high standard deviation (>=0.3) and then at the second level, assign penalty on confidence score based on standard deviation.

We choose, two model architectures for classification tasks for the given data sets: pre-trained BERT Model (Devlin et al., 2018) and BiLSTM with Attention (Att-BiLSTM) (Zhou et al., 2016). Besides, we also train a unified model that uses BiLSTM with Attention which is explained in subsection 3.3.

3.1 BERT-multi-cased (BERT_MC)

The BERT is a pre-trained bidirectional Transformer Encoder stack that is trained on large plain text corpus (Devlin et al. 2018). There are two multilingual BERT (Bidirectional Encoder Representations from Transformers) models each consist of an embedding layer with 12 encoders: 1) BERT-multi-cased (covers 104 languages) and 2) BERT-multi-uncased (covers 102 languages). We use the “BERT-multi-cased” model as it fixes the normalization issue in several languages and recommended to outperform for most of the languages. The BERT model is trained on Wikipedia and uses a common vocab of size 119,547 which is shared among 104 languages.

We adopt the BERT tokenizer for text encoding and feed the resulting encoded vector to the BERT model for training with 3 epochs. The BERT supports maximum sequence length (SEQ_LEN) up to 512 but Twitter messages are short, so we fixed it to 256 for Arabic-A & Danish-A and 128 for the rest of the subtasks. For Arabic-A, Danish-A, English-A, English-B, English-C, Greek-A, and Turkish-A, F1-score obtained using the BERT model is 0.8172, 0.7672, 0.9060, 0.6711, 0.6054, 0.8304, and 0.7481, respectively.

3.2 Attention aware BiLSTM Model: Attn-BiLSTM

Second, we apply the BiLSTM with an Attention layer on its top to capture the semantic context (see Figure 1). The model architecture consists of five components: 1) Input layer, 2) Embedding layer, 3) BiLSTM layer, 4) Attention layer, and 5) Output layer on the top. The input layer takes word tokens extracted from the pre-processed tweet messages and feeds to the embedding layer. Embedding layer maps the token index into a lower-dimensional space. BiLSTM layer tries to identify the higher-level features. Attention layer creates a weight vector and concatenates it with the word level features to get a sentence level feature vector. The output layer classifies the sentence level feature into a predefined set of class-labels.

Data cleaning layer: We apply data cleaning discussed at the beginning of Section 3.

Input layer: We apply word tokenizer to get a list of token (word) for the given twitter message. After that, we assign each token in the corpus a unique index and then apply padding to each token vector to get the fixed-length input vector. Suppose for the given sentence, the generated fixed-length vector is $\{x_1, x_2, \ldots, x_T\}$ where x_i represents a unique word index in the given train corpus, and T is the length of the input sentence.

Embedding layer: Now we transform x_i into a lower-dimensional space R^E where E is the size of the Embedding layer using an embedding matrix (W). We initialize the weights of the embedding layer randomly. Embedding matrix W is a parameter that is learned and E is a hyper-parameter chosen by the user. The x_i is transformed into e_i using the following operation:

$$e_i = W^E v^i$$

Here, v_i is a fixed-length vector of size $|V|$ (i.e., the total number of unique tokens in the corpus) which has value 1 at index e_i and 0 at rest. After applying the embedding, the transformed real-valued vector $\{e_1, e_2, \ldots, e_T\}$ is generated which are fed to the next layer. Here, T is the sentence length. For Arabic, we fixed E=128 and for the rest of the languages, it is fixed to 256.

Dropout layer: We apply dropout on the Embedding layer to avoid model overfitting. The optimal dropout value found to be 0.3 across all the subtasks.

BiLSTM layer: The traditional LSTM model processes the text sequences in temporal order and captures only the past context, whereas BiLSTM captures the past as well as the future context. We apply BiLSTM to exploit the context from both the directions. BiLSTM consists of two LSTM layers: forward LSTM and backward LSTM to compute forward ($\overrightarrow{h_i}$) and backward ($\overleftarrow{h_i}$) representations, respectively. The forward LSTM processes the input vector $\{e_1, e_2, \ldots, e_T\}$ from e_1 to e_T whereas the backward LSTM layer processes the text sequence in the opposite direction from e_T to e_1. These two representations are concatenated to get the i^{th} word representation $h_i = [\overrightarrow{h_i}; \overleftarrow{h_i}]$. We have used 128 forward and 128 backward LSTM units for the Arabic language subtask-A and 256 forward and 256 backward LSTM units for the rest of the subtasks.

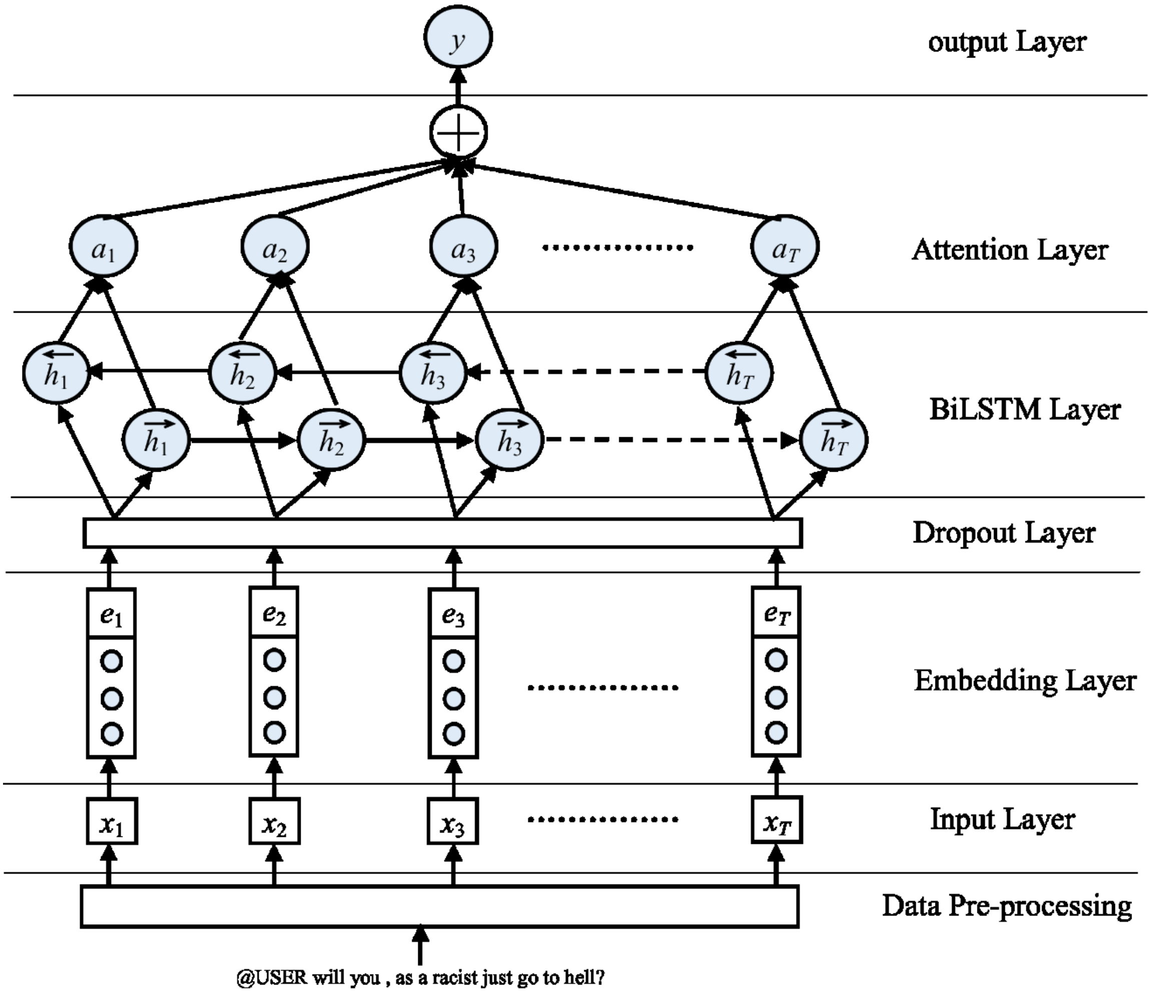


Figure 1: Architecture of the Attention based Bidirectional LSTM Model.

Attention layer: In offense analysis, all words do not equally contribute to understand the context. The attention mechanism is applied to capture the relative influence of various words used in the tweet messages by assigning a weight a_i to each word representation. Suppose H is a matrix that consists of BiLSTM layer output vectors $[h_1, h_2, \ldots, h_T]$ where T is the sentence length. The representation r is fixed across all the input sentences and computed using the below equations:

$$M = \tan(H)$$
$$\alpha = softmax(w^T M)$$
$$r = H\alpha^T$$

Here, w^T is the transpose of a trainable parameter vector, w. Suppose, H ∊ $R^{d \times T}$ where d is the dimension of word vector and T is the length of the input sentence. The dimensions of w, α, and r vectors are d, T, and d, respectively. The final sentence representation is computed using the following equation:

$$h^* = \tanh(r)$$

Output layer: We feed the h* representation to the fully connected softmax layer which outputs a probability distribution over all classes.

3.3 Unified multilingual BiLSTM Model: UML_BiLSTM

We also apply the BiLSTM model (discussed in subsection 3.2) for the combined Subtask-A in the multilingual environment (named as a unified multilingual BiLSTM viz. UML_BiLSTM). The benefit of the UML _BiLSTM model is twofold, first having one model reduces the overall model training time, and second, it reduces the manual effort of maintaining various models. Figure 2 depicts the UML_BiLSTM model pipeline.

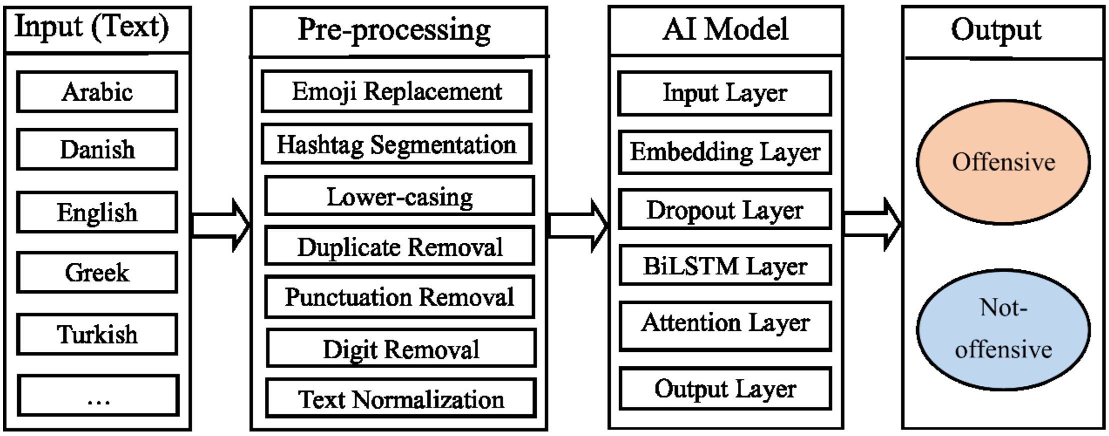


Figure 2: Block Diagram of unified multilingual BiLSTM Model.

4 Data Specification and Experimental Analysis

In this section, we present the data description and the experimental analysis for SemEval-2020 Task 12. We first report the submitted accuracy and then enhanced accuracy. We perform text normalization for foreign languages which results in improved accuracy. Also, we filter all the digits from the tweet messages as they are not contributing to offense analysis. We also combine all language data of subtask-A to train a single unified BILSTM model (described in subsection 3.3) for the multilingual environment. We report the accuracy in terms of macro-F1-score which gives equal weightage to all the classes and thus is a reliable metric in the presence of class-imbalance in the data.

4.1 Data Specification

The datasets, given by the organizers, contains five different languages for subtask-A and only the English language for subtasks-B & C. The datasets are collected from the social media (Zampieri, 2020). A complete description of the datasets collection and annotation techniques for Arabic, Danish, English, Greek, and Turkish languages can be found in (Mubarak et al., 2020), (Sigurbergsson and Derczynski, 2020), (Rosenthal et al. 2020), (Pitenis et al., 2020), and (Ç Çöltekin, 2020), respectively.

Table 1 shows the train-test data size and the vocab-size after pre-processing the train-set for each subtask. Large data samples for English subtask-A results in higher vocab-size compare to English subtask-B and English subtask-C. Vocab-size also depends on the richness of the language and thus might vary across different languages even though the sample size is the same which can be observed from the table.

Subtask	Train-size	Test-size	Train vocab-size
Arabic-A	8,000	2,000	33,987
Danish-A	2,961	329	9,836
English-A	9,089,140	3,887	401,164
English-B	188,974	1,422	77,798
English-C	188,973	850	67,103
Greek-A	8,743	1,544	23,480
Turkish-A	31,277	3,528	88,226

Table 1: Data Specification and vocab-size after applying text pre-processing.

4.2 Experimental Results and Analysis

We divide the train-set into train and the development sets for hyper-parameter-tuning for the BERT and BiLSTM models. For all the subtasks, we submit the best accuracy achieved on the development set.

In Table 2, we showcase the macro F1-score for all the subtasks. We report the F1-score of our submitted results and the respective applied model name. The submission results are using the BERT model for Greek-A and Turkish-A subtasks as it was giving better accuracy on the development set. We couldn't submit the results for the English-B subtask. The last column shows the best system accuracy of the participant who scored first rank in SemEval Task 12.

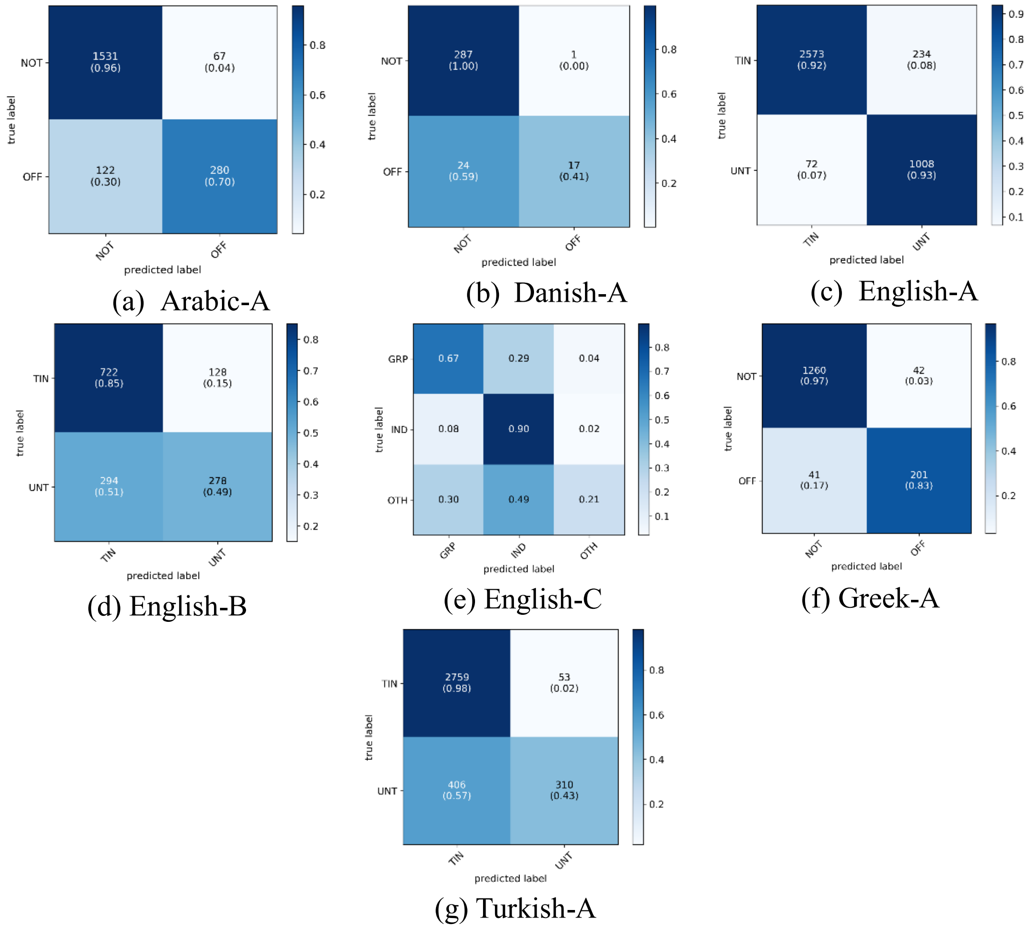
We also report the enhanced accuracy result for the BERT and BiLSTM models. The BiLSTM model outperforms on Arabic-A and Greek-A whereas the fine-tuned BERT model outperforms on the rest of the subtasks (see table 2). The BERT model uses a common vocab of size 119,547 (shared among 104 languages) in which Arabic and Greek vocab sizes are 4,873 and 1,566, respectively. But in the given datasets after removing redundant tokens, the vocab-size is 33,987 and 23,480 for Arabic-A and Greek-A, respectively. Our BiLSTM model learns better by using larger vocabulary, resulting in better accuracy on Arabic-A and Greek-A subtasks. Moreover, the pre-trained BERT model is resource-hungry and compute-intensive. For Greek subtask-A, our enhanced accuracy shows a significant improvement over the accuracy achieved by the top ranker.

The effect of text normalization varies across different languages. This is because the normalization techniques applied to different languages vary due to the morphological variation across these five languages. For Arabic-A, initially, we use the train-dev-test sets released by the organizers and submit the results obtained from the model trained on train-set only. But later we download the updated dataset in which train and development sets are combined by the organizers. So, the enhanced results are also computed on the updated dataset for the fair comparison with the top ranker score (assuming the top ranker scores are computed on the updated dataset). In Arabic-A, this is the major factor for the drastic accuracy improvement over the submitted accuracy.

Subtask	**Submitted**		**Further Enhancement**			**Top Ranker F1**
	F1-score	**Model**	**BERT F1**	**BiLSTM F1**	**UML_BiLSTM F1**	
Arabic-A	0.4536	BiLSTM	0.8172	0.8447	0.8192	**0.9017**
Danish-A	0.6710	BiLSTM	0.7672	0.7000	0.6543	**0.8120**
English-A	0.8900	BiLSTM	0.9060	0.8970	0.8775	**0.9222**
English-B	Not Applicable	BiLSTM	0.6711	0.6200	Not applicable	**0.7461**
English-C	0.5259	BiLSTM	0.6054	0.5774	Not applicable	**0.7145**
Greek-A	0.8020	BERT	0.8304	**0.8984**	0.7747	0.8520
Turkish-A	0.7421	BERT	0.7489	0.7254	0.6815	**0.8257**

Table 2: Macro F1-score on test-sets for all subtasks. For each subtask, our team submitted F1-score, the enhanced F1-score obtained by the BERT & the BiLSTM, F1-score obtained using the UML_BiLSTM model, and the top ranker submission score are reported (best in bold).

In Figure 3, we showcase the confusion matrix computed on test-sets of the respective enhanced model which achieves a higher F1-score.



(a) Arabic-A

(b) Danish-A

(c) English-A

(d) English-B

(e) English-C

(f) Greek-A

(g) Turkish-A

Figure 3: Confusion matrix on test-set for (a) Arabic-A, (b) Danish-A, (c) English-A, (d) English-B, (e) English-C, (f) Greek-A, (g) Turkish-A.

5 Conclusion

We used the BERT and the BiLSTM model for multilingual offensive language identification and compared the results. We demonstrated in the result analysis that our attention-aware BiLSTM model outperforms the BERT model for Arabic and Greek offensive language detection, while BERT performed better for Danish, English, and Turkish offensive language identification, English offense-type categorization, and English offense target identification. Further, we combined Arabic, Danish, English, Greek, Turkish multilingual offensive language identification data to train our UML_BiLSTM model. The obtained UML_BiLSTM model results were poor by -0.20 to 11.29% and 1.95% to 12.37% compared to the BERT and the BiLSTM, respectively. However, the UML_BiLSTM model training was 2.81 times and 1.34 times faster in comparison to the BERT and the BiLSTM models, respectively. UML_BiLSTM model can be further improved by introducing a data balancing mechanism before training the model. Our submitted F1-scores are 0.4536, 0.6710, 0.8900, 0.8020, 0.7421, and 0.5259 for Arabic subtask-A, Danish subtask-A, English subtask-A, Greek subtask-A, Turkish subtask-A, and English subtask-C, respectively. After submission, we further worked on enhancing the models for achieving better accuracy. We normalized the data for foreign languages before feeding to the BiLSTM model and got improved results. Especially for the Greek language, our achieved accuracy was significant compare to the OffensEval-200 top ranker F1-score. The top ranker accuracy was 0.8520 and our improved accuracy was 0.8984. Therefore, it has been proven that the normalization of social media posts is one of the important steps before applying model. The improved F1-scores are 0.8447, 0.7672, 0.9060, 0.8984, 0.7489, 0.6711, and 0.6054 for Arabic subtask-A, Danish subtask-A, English subtask-A, Greek subtask-A, Turkish subtask-A, English subtask-B and English subtask-C, respectively.

In the future, we also plan to evaluate our model on a wide range of languages.

Reference

Khaled Mohammad Alomari, Hatem M. ElSherif, and Khaled Shaalan. 2017. Arabic tweets sentimental analysis using machine learning. In *International Conference on Industrial, Engineering and Other Applications of Applied Intelligent Systems*, pages 602-610. Springer, Cham.

Valerio Basile, Cristina Bosco, Elisabetta Fersini, Debora Nozza, Viviana Patti, Francisco Rangel, Paolo Rosso, and Manuela Sanguinetti. 2019. Semeval-2019 task 5: Multilingual detection of hate speech against immigrants and women in twitter. In *Proceedings of the 13th International Workshop on Semantic Evaluation (SemEval-2019)*. Association for Computational Linguistics.

Cristina Bosco, Felice DellOrletta, Fabio Poletto, Manuela Sanguinetti, and Maurizio Tesconi. 2018. Overview of the EVALITA 2018 Hate Speech Detection Task. In *Proceedings of the 6th evaluation campaign of Natural Language Processing and Speech tools for Italian (EVALITA'18)*, Turin, Italy. CEUR.org.

Miguel Ángel Álvarez Carmona, Estefanía Guzmán-Falcón, Manuel Montes-y-Gómez, Hugo Jair Escalante, Luis Villaseñor Pineda, Verónica Reyes-Meza, and Antonio Rico Sulayes. 2018. Authorship and aggressiveness analysis in Mexican Spanish tweets. In *Proceedings of the 3rd Workshop on Evaluation of Human Language Technologies for Iberian Languages (IberEval'18) co-located with the 34th Conference of the Spanish Society for Natural Language Processing (SEPLN'18)*. 74–96.

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

Michele Corazza, Stefano Menini, Elena Cabrio, Sara Tonelli, and Serena Villata. 2020. A Multilingual Evaluation for Online Hate Speech Detection. *ACM Transactions on Internet Technology (TOIT)*, no. 2, pages 1-22.

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

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

Bjorn Gamback and Utpal Kumar Sikdar. 2017. Using Convolutional Neural Networks to Classify Hatespeech. In *Proceedings of the First Workshop on Abusive Language Online*, pages 85–90.

Cynthia Van Hee, Gilles Jacobs, Chris Emmery, Bart Desmet, Els Lefever, Ben Verhoeven, Guy De Pauw, Walter Daelemans, and Véronique Hoste. 2018. Automatic detection of cyberbullying in social media text. PloS one 13, no. 10.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. LSTM can solve hard long time lag problems. In *Advances in neural information processing systems*, pages 473-479.

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

Younghun Lee, Seunghyun Yoon, and Kyomin Jung. 2018. Comparative studies of detecting abusive language on Twitter. CoRR abs/1808.10245.

Ping Liu, Wen Li, and Liang Zou. 2019. Transfer learning for offensive language detection using bidirectional transformers. In *Proceedings of the 13th International Workshop on Semantic Evaluation. Association for Computational Linguistics*, pages 87–91. DOI:https://doi.org/10.18653/v1/S19-2011

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.

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. 2020. Offensive Language Identification in Greek. In *Proceedings of the 12th Language Resources and Evaluation Conference (LREC)*.

Sara Rosenthal, Pepa Atanasova, Georgi Karadzhov, Marcos Zampieri, and Preslav Nakov. 2020. A large-scale semi-supervised dataset for offensive language identification. *Computing Research Repository*, arXiv preprint arXiv:2004.14454.

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.

Seungwan Seo, Czangyeob Kim, Haedong Kim, Kyounghyun Mo, and Pilsung Kang. 2020. Comparative Study of Deep Learning-Based Sentiment Classification. IEEE Access 8 (2020): 6861-6875.

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 (LREC).

Zeerak Waseem. 2016. Are you a racist or am i seeing things? annotator influence on hate speech detection on Twitter. In *Proceedings of the 1st Workshop on Natural Language Processing and Computational Social Science*, pages 138–142.

Michael Wiegand, Melanie Siegel, and Josef Ruppenhofer. 2018. Overview of the GermEval 2018 Shared Task on the Identification of Offensive Language. In *Proceedings of GermEval*.

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 2019). In *Proceedings of the 13th International Workshop on Semantic Evaluation (SemEval)*.

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 the 13th International Workshop on Semantic Evaluation (SemEval)*.

Ziqi Zhang, Jonathan Tepper, and David Robinson. 2018. Detecting Hate Speech on Twitter Using a Convolution-GRU Based Deep Neural Network. In *Lecture Notes in Computer Science*. Springer Verlag.

Peng Zhou, Wei Shi, Jun Tian, Zhenyu Qi, Bingchen Li, Hongwei Hao, and Bo Xu. 2016. Attention-based bidirectional long short-term memory networks for relation classification. In *Proceedings of the 54th annual meeting of the association for computational linguistics (volume 2: Short papers)*, pages 207-212.