Proceedings of the 4th Workshop on Open-Source Arabic Corpora and Processing Tools, pages 66–70
with a Shared Task on Offensive Language Detection.
Language Resources and Evaluation Conference (LREC 2020), Marseille, 11–16 May 2020

ASU_OPTO at OSACT4 - Offensive Language Detection for Arabic text

Amr Keleg [1], Samhaa R. El-Beltagy [2], Mahmoud Khalil [1]
[1] Faculty of Engineering - Ain Shams University, [2] Optomatica
[1] Cairo, [2] Giza
Egypt
amr.keleg@eng.asu.edu.eg, samhaa@computer.org, mahmoud.khalil@eng.asu.edu.eg

Abstract

In the past years, toxic comments and offensive speech are polluting the internet and manual inspection of these comments is becoming a tiresome task to manage. Having a machine learning based model that is able to filter offensive Arabic content is of high need nowadays. In this paper, we describe the model that was submitted to the Shared Task on Offensive Language Detection that is organized by (The 4th Workshop on Open-Source Arabic Corpora and Processing Tools). Our model makes use transformer based model (BERT) to detect offensive content. We came in the fourth place in subtask A (detecting Offensive Speech) and in the third place in subtask B (detecting Hate Speech).

Keywords: Offensive speech, Hate speech, BERT

1. Background and task description

During the past decade, Social media platforms such as Facebook and Twitter have attracted millions of users from the Arab region. These platforms have given people the chance to express their ideas, beliefs and feelings. Unlike real life conversations, people tend to be more aggressive when they are communicating through this virtual online world. The aggression might also reach an extreme case where racist, violent and completely unacceptable words are shared online. Sites are trying to control the spread of these toxic comments by manually moderating and checking the reports that other users are filing. Moreover, Some services provide an automatic way to automatically filter offensive content. For example, Google Search has an option to use "SafeSearch Filters" which is allows filtering out any harmful or violent content before presenting the search results to the user.

All these facts have attracted researchers from all around the world to build different techniques that can be used to automatically detect offensive content. Various definitions and aspects have been used to tackle this task. Having a typology that can be clearly agreed upon by humans is of great importance. Mubarak et al. (2017) have used the term abusive speech to refer offensive text that contains profane content. On other hand, Hate speech (Toxic comments) is often used to refer to offensive text that is targeted towards a certain person or a group of people based on a common trait (race, ethnicity, religion, etc.) (Malmasi and Zampieri, 2017).

The competition is composed of two subtasks. Subtask A aims at differentiating between offensive and non-offensive text irrespective of the type of the offensive text (Hate Speech, Profanity, Cyber-bullying, etc). Substask B focuses on detecting text that contains targeted Hate Speech towards a person or a group of people.

2. Systems description

Lately, Fine-tuning large models using the idea of transfer learning such as: BERT (Devlin et al., 2019) and ULMFiT (Howard and Ruder, 2018) that are pre-trained on language modeling tasks reaches state-of-the-art results in multiple classification tasks. For this competition, we have focused on Subtask A and tested different models/ architectures keeping in mind that fine-tuning BERT based models should be among the top performing ones. The best performing model for subtask A was then adapted to work on subtask B as well. The following models were developed throughout our experiments[1]:

- Training a basic model using tf-idf (term frequency - inverse document frequency) and logistic regression. The tf-idf generates a sparse representation of the input text using character ngrams in range [1, 9] e.g: Some of the grams of the sentence (في غيابك الفرح والنور) are (غ، غي، غيا، غياب، غيابك، و، وا، وال، والن، والنو، والنور، ل، لن، لنو، لنور، ن، نو، نور) .
  This sparse feature vector is then fed to the logistic regression model to discriminate between the two classes (offensive and non-offensive). This model represents the baseline model for all other deep learning based architectures.

- Training a 1D Convolutional Layer using word embeddings from Aravec (Mohammad et al., 2017) as a 2D input array. At first, the line-feed token <LF>is replaced by a newline character \n. Then, the sentence is cleaned in the way that is used by the Aravec model. This step includes the removal of diacritics and fixing elongated words (Replacing any sequence of the same character of length two or more by a sequence of length two of the same character). Then, the sentence is tokenized using whitespaces. The tokens are mapped to their respective index in the word2vec model using 0 as the index for any unknown token.

[1] The source code for the developed models can be found through: `https://github.com/AMR-KELEG/offenseval-2020-ASU_OPTO`

The list of ids is then padded by the id 0 such that it has a fixed length of 75 ids. The list is truncated to have the length of 75 in case it had more than 75 tokens. The list of ids is then used to generate the respective word embeddings. The word embeddings are concatenated to form a 2D array of shape(75, 300) where 300 is the size of the word embedding for each token. 100 different 1D convolutional filters are then applied to the 2D array with kernel size of 3 and stride of 1 (e.g: the filter is applied to the word embeddings of all 3-consecutive tokens). A 1D max-pooling layer is then applied with a pool size of 4. Drop-out with probability of 0.5 succeeds the max pooling layer then a Dense layer of 1 neuron with a sigmoid activation function is used to predict the probability that the sentence is offensive or not. The model is trained for 2 epochs with L2 regularization (penalty factor is set to 0.0001). The used cost function is binary cross entropy and it's optimized using Adam (Kingma and Ba, 2014). The initial word vectors will also be fine-tuned during the training process to minimize the cost function.

- Training a Bi-directional LSTM using word embeddings from Aravec. Only the most occurring 300,000 words of the Aravec vocabulary are kept and fine-tuned as part of the model due to the limited GPU memory. After the embedding layer, a bidirectional LSTM layer of 64 cells is used followed by two dense layers of 64 neurons with relu activation function and 1 neuron with a sigmoid activation function.

- Fine-tuning multilingual BERT that is pre-trained on cased text of the top 104 languages with the largest Wikipedias (which includes Arabic). The text is tokenized using a word piece tokenizer (Wu et al., 2016) which is trained on large text in an unsupervised fashion to determine a set of word-pieces that form the words (e.g: the word **unaffable** might be split to (un, ##aff, ##able) according the word-pieces that were generated on training the tokenizer). After tokenizing the input text, the tokens are padded/truncated to the length of 75. BERT generates an embedding for the whole sentence using its self-attention layers. A Dense layer with softmax activation is then added to classify the sentence into offensive or not. The whole pretrained architecture in addition to the added dense layer are then fine-tuned using the tagged dataset. The model is fine-tuned for three epochs using a learning rate of 10^{-5} and with L2 regularization.

- Fine-tuning AraBERT (a publicly released BERT model trained on Arabic text [2]). The text is tokenized using Farasa (Abdelali et al., 2016) which is a segmenter that is developed to segment an Arabic word into its affixes. Then, the tokens are fed to the BERT model. The default values provided by the model's authors were used in the fine-tuning process. The training dataset was divided into batches of size 32, where each sample was tokenized to have a length of 64. Six epochs were used to fine-tune the pre-trained AraBERT model on the training dataset of 7000 samples with a learning rate of 10^{-5}.

Moreover, We have built a list of profanity words and used simple augmentation rules to generate the different forms of each word. Mubarak et. al (2017) have demonstrated the effectiveness of using a list of words to detect abusive content in text documents. They used a seed list of bad words and collected user data from twitter to find other candidate words that: 1) are used by those who have any of the seed words in their tweets. 2) aren't used by those who don't have any of the seed words in their tweets. We build on the same idea of having a list of profanity words to automatically mark some tweets as offensive irrespective of their context but we have used a morphological approach for augmenting our seed list of bad words. First, we used a list of bad words that is available online[3]. The list of bad words was manually augmented to include other common forms of an Arabic word by substituting ة (Taa-marbuta) with ه (Haa) and substituting ز (Zain) with ذ (Zaal). Then, the list was further augmented by other bad words that could be found in the training data-set using manual inspection. Finally, a list of prefixes and suffixes were used to generate the different morphological forms of each word. For example, if the word was a verb then the list of prefixes to be added would be (ا، ي، ت، ن، ه، سا) and the list of suffixes would be (ني، ك، ها، هم، كم، هن، نا). e.g.: For the verb هزم, 113 different morphological forms are generated. The following words represent a sample of these forms:
هزم، اهزم، اهزمك، اهزمكم، اهزمنا، اهزمني، اهزمها، اهزمهم، اهزمهن، يهزم، يهزمك، يهزمكم، يهزمنا، يهزمني، يهزمها، يهزمهم، يهزمهن

A seed list of 87 bad words was augmented to reach 5497 different words. Some combinations of the prefixes and suffixes might result in a word that is not linguistically valid but our intuition is since the word isn't part of the language then nobody will use it and thus considering a word that is impossible to be used to be a bad word won't affect the model's precision.
Throughout our experiments, we have faced problems with reproducing the results for models that are trained using GPUs among multiple runs given that we had used a random seed of value 42 in all our experiments. This seems like a problem that isn't widely discussed. The reproducibility problem can be partially mitigated by training the model multiple times while saving the trained weights for each training run and then choosing the best performing version of the model.

3. Results

Table 1 reports the accuracy and the macro-averaged precision, recall and F1 scores for the training and development datasets respectively on subtask A. Our best model

[2]The initial version of AraBERT can be found through: `https://github.com/zaidalyafeai/ARBML/issues/18#issuecomment-580924000`

[3]https://github.com/LDNOOBW/List-of-Dirty-Naughty-Obscene-and-Otherwise-Bad-Words

Table 1: Results of the developed models on the training and development datasets

	Training dataset				Development dataset			
Model name	*Accuracy*	*Precision*	*Recall*	*F1*	*Accuracy*	*Precision*	*Recall*	*F1*
tfidf + logistic regression	0.889	0.938	0.725	0.778	0.888	**0.921**	0.694	0.746
CNN + Aravec	0.982	0.985	0.959	0.971	0.928	0.906	0.838	0.867
BiLSTM	**0.999**	**0.998**	**0.998**	**0.998**	0.920	0.856	**0.884**	0.869
Multi-lingual BERT	0.978	0.975	0.956	0.965	0.905	0.855	0.805	0.826
AraBERT	0.998	0.998	0.994	0.996	**0.928**	0.881	0.871	**0.876**

Table 2: Effect of using the list of profane words on the fine-tuned AraBERT reported on the development dataset

Model name	*Accuracy*	*Precision*	*Recall*	*F1*
AraBERT	0.928	0.881	0.871	0.876
AraBERT + augmented list of profane words	**0.930**	**0.883**	**0.877**	**0.880**

for subtask A was the AraBERT based model which performed better than the cased multilingual BERT model that is trained using the dumps of the 104 most represented languages on wikipedia. Researchers focusing on langauges other than English have found that a BERT model trained specifically for a certain language such as: German, Greek and Dutch (de Vries et al., 2019) achieves better results than the multilingual BERT model that might under-represent some languages. Additionally, The results of the Offenseval 2019 (Zampieri et al., 2019) competition reported that 7 out of the top 10 teams have used BERT to build their models. Risch, et al. (2019) have also showed that using a BERT model that is trained using large German corpora performs better than all the other baseline models.
The AraBERT based model was also succeeded by a simple look-up search that marks a sentence as offensive if it contains any of the words in the augmented profanity words list irrespective of the prediction of the AraBERT model. Using this hybrid approach has improved the macro-averaged precision and recall and consequently improved the macro-averaged F1 score as shown in table 2. The official macro-averaged F1 score of this hybrid system on the test and development datasets is 0.896 which is much better than that of our second best system that is based on the Bidirectional LSTM which achieved an official score of 0.856.
For subtask B, We have fine-tuned AraBERT using the whole training dataset of 7000 tweets with the same configuration and hyperparameters that were used in subtask A. Our official macro-averaged F1 score is 0.807 which put our team in the third place on the scoreboard.

4. Error Analysis

One of the important steps to carry-out on training a machine learning model is to check the mis-classified samples and try to find reasonable explanations for such errors. This task might be hard for text data since one can't easily find relations between different samples unlike images for example. On checking a random sample of 50 mis-classified samples, we found that most of the errors were False Negatives (The sample is offensive yet it was classified as not offensive). Additionally, we found that all these samples contained the Arabic vocative article يا (Ya). This seemed like a really serious problem that needs to be fixed until we discovered that (6986 out of 7000) of the sentences in the training and (999 out of 1000) of the sentences in the development data-sets contain the article يا (Ya). The effect of such observation on the model needs more analysis but clearly this article was used by the data-set creators to query sentences (tweets) and it might limit the distribution of the corpus.

Table 3: Tweets containing bad words with mixed inconsistent labels

ID	Text	Label
2206	ده اللى كنت خايفة منه اصل تخيلى انزل صورى عادى و اجى الاقى الناس بتتبادل صورى و عليها رسم اسهم ودواير عشان نبين الفوتوشوب فين ا** ده انا هخاف و طبيعى همسحها مبقولش ان ده خداع و انه مش صح بس خوفى انى اجرحها يخلينى يا اسكت يا اما اعلق تعليق بسيط	NOT OFF
7177	لا ثانية واحدة كدة هى اسمها كان يا ما كان يا سادة يا كرام مش يا سعد يا إكرام ا***	OFF

4.1. Issues with the Annotation scheme

Human annotation is a tiresome task especially in the field of natural language processing since text might sometimes be ambiguous in a way that the same sentence might carry different meanings. In this section, we will shed the lights on different issues that we have spotted on performing error analysis.

Presence of a bad word in a non-negative context: The way people perceive and use bad words might depend on different factors such as: the dialect that they use or their society's culture. Some words might be accepted in some regions but are completely inappropriate in other regions.

Table 4: Tweets with offensive semantic meaning and sarcastic pragmatic meaning

ID	Text	Label
261	RT @USER: وشوشنى وديتها فين يا محول يا أبو عين واحده ؟! أول هام أسمها أعور .. لما تحب تهزئنى هزئنى صح URL	OFF
7868	It seems like تاني يا زكي يا تافه .. رحتلهم تاني عشان يضربوك وياخدوا هدومك تاني	NOT OFF

Table 5: Tweets containing Offensive content with incorrect labels

ID	Text	Label
7106	إلي بحط جدول المحاضرات بالجامعة اقسم بالله ما بكون صاحي ! يا غبي ٣ مواد تخصص بنفس الساعة كيييف؟؟ أكيد بكون يا نايم يا بحلم بذكرني بأركان وهو بكتب أحداث كارا دينيز	NOT OFF - NOT HS
7491	@USER @USER التهريج مش بكلام هشام التهريج بكلامك الاستزلامي و الانبطاحي و السوقية بتوجيهك الكلام الو. عادي الاستزلاميين هيدي لغتن مش انو شي جديد الزلمي بكلامو ما كان في اهانات و تجريح بعكس انتي يلي مفروض يا متعلمة يا بتوعت المدارس. عموما يعط	NOT OFF - NOT HS
7358	يا عيباه يا حسافاه اليمني يهان بكل مكان وهذا كله بسبب قيادته الملعونه المرتزقة والمشكله نحن كجنوبين والله ثم والله اننا نحترم ونقدر الشعب في الشمال وكل خلافنا سببة قادتهم السرق واعمالهم في الجنوب متى يصحى الشعب اليمني ويشوف الاهانات من كل البلدان متى ويبني علاقة حب مع الجنوبيين	NOT OFF - NOT HS

Additionally, Annotators might neglect the presence of a bad word if the context isn't offensive while others consider the whole sentence to be offensive if it contains a bad word. Table 3 demonstrates the disagreement problem between human annotators where the same bad word (with different forms) was found in a non-offensive context. Annotators have considered the first to be not offensive but marked the second one as offensive.

Usage of sarcastic speech quoting popular movie scenes: Our Arabic culture relies heavily on quoting conversations from popular movies. The semantic meaning of these words might be offensive but the pragmatic meaning will depend on the context in which they are used. Ambiguity is an issue that rises in almost all the systems that operate on linguistic data. Table 4 shows two examples where quotes from movies were used. Although the fact that the model can only depend on the semantic meaning of the sentence, we believe that annotators should pick a side and mark them as either offensive or not. The two sentences have offensive speech yet one of them was annotated as offensive and the other was annotated as non offensive.

Wrong annotations: Having errors in annotations generated by humans is a problem that is almost unavoidable especially if the dataset was of a large size (10,000 tweets) and annotators are asked to provide two different labels for each tweet (Offensive or not offensive and Hate speech or not hate speech). In table 5, we believe that all these samples should have marked as offensive and as hate speech.

5. Conclusion

Our experiments reveals that the contextualized word embeddings generated using BERT yield better classifiers for offensive text detection. A BERT model that is pre-trained on large text corpora achieves state-of-the-art results. On the other hand, multilingual BERT seemed to lack the ability to represent Arabic text. This might be attributed to the fact that Arabic text needs to be tokenized in a different way than the other languages that are supported by multilingual BERT. Additionally, using a hybrid approach improved our system that is used for subtask A. Relying on a manually prepared list to mark a sentence that contains a profane word as offensive is a logical solution to support machine learning based models.

6. Bibliographical References

Abdelali, A., Darwish, K., Durrani, N., and Mubarak, H. (2016). Farasa: A fast and furious segmenter for arabic. 06.

de Vries, W., van Cranenburgh, A., Bisazza, A., Caselli, T., van Noord, G., and Nissim, M. (2019). Bertje: A dutch bert model. *ArXiv*, abs/1912.09582.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (2019). BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota, June. Association for Computational Linguistics.

Howard, J. and Ruder, S. (2018). Universal language model fine-tuning for text classification. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 328–339, Melbourne, Australia, July. Association for Computational Linguistics.

Kingma, D. P. and Ba, J. (2014). Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Malmasi, S. and Zampieri, M. (2017). Challenges in discriminating profanity from hate speech. *Journal of Experimental & Theoretical Artificial Intelligence*, 30(2):187202, Dec.

Mohammad, A. B., Eissa, K., and El-Beltagy, S. (2017). Aravec: A set of arabic word embedding models for use in arabic nlp. *Procedia Computer Science*, 117:256–265, 11.

Mubarak, H., Darwish, K., and Magdy, W. (2017). Abusive language detection on Arabic social media. In *Proceedings of the First Workshop on Abusive Language Online*, pages 52–56, Vancouver, BC, Canada, August. Association for Computational Linguistics.

Risch, J., Stoll, A., Ziegele, M., and Krestel, R. (2019). hpidedis at germeval 2019: Offensive language identification using a german bert model. In *Preliminary proceedings of the 15th Conference on Natural Language Processing (KONVENS 2019). Erlangen, Germany: German Society for Computational Linguistics & Language Technology*, pages 403–408.

Wu, Y., Schuster, M., Chen, Z., Le, Q. V., Norouzi, M., Macherey, W., Krikun, M., Cao, Y., Gao, Q., Macherey, K., Klingner, J., Shah, A., Johnson, M., Liu, X., ukasz Kaiser, Gouws, S., Kato, Y., Kudo, T., Kazawa, H., Stevens, K., Kurian, G., Patil, N., Wang, W., Young, C., Smith, J., Riesa, J., Rudnick, A., Vinyals, O., Corrado, G., Hughes, M., and Dean, J. (2016). Google's neural machine translation system: Bridging the gap between human and machine translation.

Zampieri, M., Malmasi, S., Nakov, P., Rosenthal, S., Farra, N., and Kumar, R. (2019). SemEval-2019 task 6: Identifying and categorizing offensive language in social media (OffensEval). In *Proceedings of the 13th International Workshop on Semantic Evaluation*, pages 75–86, Minneapolis, Minnesota, USA, June. Association for Computational Linguistics.