BhamNLP at SemEval-2020 Task 12: An Ensemble of Different Word Embeddings and Emotion Transfer Learning for Arabic Offensive Language Identification in Social Media

Abdullah I. Alharbi
School of Computer Science
University of Birmingham, UK
aia784@cs.bham.ac.uk

Mark Lee
School of Computer Science
University of Birmingham, UK
m.g.lee@cs.bham.ac.uk

Abstract

Social media platforms such as Twitter offer people an opportunity to publish short posts in which they can share their opinions and perspectives. While these applications can be valuable, they can also be exploited to promote negative opinions, insults, and hatred against a person, race, or group. These opinions can be spread to millions of people at the click of a mouse. As such, there is a need to develop mechanisms by which offensive language can be automatically detected in social media channels and managed in a timely manner. To help achieve this goal, SemEval 2020 offered a shared task (OffensEval 2020) that involved the detection of offensive text in Arabic. We propose an ensemble approach that combines different levels of word embedding models and transfer learning from other sources of emotion-related tasks. The proposed system ranked 9th out of the 52 entries within the Arabic Offensive language identification subtask.

1 Introduction

While microblogging platforms can be used positively and productively, they are also frequently used for destructive purposes such as spreading offensive messages. People who wish to spread hate can use these channels to quickly and easily reach millions of people at the click of a button. To prevent this spread, there is a need for a system that can automatically identify messages that contain offensive language. A significant number of studies have focused on the detection of offensive text in English. However, there is a lack of research that focuses on detecting offensive text in Arabic (Mubarak and Darwish, 2019). To contribute to the emerging body of knowledge in this domain, a shared task 'OffensEval 2020 - Task 12' was conducted at the SemEval 2020 (Zampieri et al., 2020). One of the subtasks of this shared task is 'Offensive language identification'. The organisers released multilingual datasets for five languages, including Arabic, which is the focus of this paper.

The shared task involved some significant challenges. The distribution of the targeted classes was unbalanced: 19% of the tweets were labelled as offensive, while the remaining were labelled as inoffensive. Also, the tweets employed a variety of dialects and sub-dialects. Unlike Modern Standard Arabic (MSA), dialectical Arabic can take a variety of forms, and there is a general lack of rules and standards. To address these challenges, we proposed an ensemble approach that combines three different models (detailed in Section 3). We believe that integrating multiple kinds of classifiers can help overcome the weaknesses and realise the advantages of each. Our proposed system ranked 9th out of 52 participants for the Arabic Offensive language identification subtask.

The remainder of this paper is organized as follows. Section 2 provides a brief overview of the current literature on the detection of offensive Arabic text. Section 3 includes an overview of the methodology, including the experimental setup by which the proposed system was tested. Section 4 provides an evaluation, analysis and discussion of the outcomes. Finally, Section 5 concludes the paper with suggestions for future research.

Proceedings of the 14th International Workshop on Semantic Evaluation, pages 1532–1538
Barcelona, Spain (Online), December 12, 2020.

2 Related Work

The identification of offensive text in the English language has been studied in depth, and multiple offensive categories have been identified, including racist speech, religious hate speech, and sexism (Davidson et al., 2017; Malmasi and Zampieri, 2017; Kumar et al., 2018; Waseem et al., 2017; Zampieri et al., 2019). In contrast, only a few studies have focused on identifying offensive language in Arabic (Al-Hassan and Al-Dossari, 2019).

One of these few studies was performed by Mubarak et al. (2017) who produced an extensive corpus of Arabic tweets that were manually classified according to three categories: clean, obscene and offensive. A further contribution to the body of knowledge was offered by Alakrot et al. (2018), who developed a corpus that consisted of offensive comments that had been posted on YouTube in Arabic. The dataset consisted of 16,000 Egyptian, Libyan and Iraqi comments that were subsequently categorized into one of three classes: offensive, inoffensive and neutral. The researchers trained a support vector machine (SVM) classifier to detect the offensive comments. The results indicated that the use of an N-gram feature could enhance the accuracy of the classifier, while using a combination of an N-gram with stemming diminished system performance. Research that was performed by Albadi et al. (2018) was limited to the use of Arabic to share religious hate. They developed and ranked a lexicon of the religious hate terms that were in most common use and tested various classifiers.

More recently, Mubarak and Darwish (2019) extended their existing corpus of offensive Arabic words and used it to construct a large-scale training model that could be used to automatically detect offensive tweets. The researchers developed a character-level deep learning algorithm that was subsequently employed to classify each tweet as either offensive or inoffensive. Mubarak et al. (2020) created the largest Arabic-language dataset to date and employed specific tags for hate speech and vulgarity. They subsequently analyzed the dataset to identify the topics that were most commonly associated with offensive tweets and to obtain meaningful insights into the use of hate speech among members of the Arab community. Finally, they proved the effectiveness of the system by performing a range of large-scale experiments on the dataset using SVM techniques and generated promising results (F1 = 79.7).

3 Methodology

The proposed ensemble system consists of three different classifier models. We pre-processed the raw tweets as inputs (see Section 3.1) to the models. Sections 3.2, 3.3 and 3.4 describe the three different models which are then combined using an ensemble technique as described in Section 3.5.

3.1 Preprocessing

Pre-processing was undertaken following a procedure used by several researchers previously (Abu Farha and Magdy, 2019; Duwairi and El-Orfali, 2014). Firstly, any unknown symbols or other characters were removed, e.g. letters from other languages, punctuation, diacritics, etc. However, emojis were retained. We also normalised several letters which appeared in different forms in the original tweets; these were rendered into a single form. For example, the 'hamza' on characters (أ,إ) was replaced with the (ا), and the 't marbouta' (ة) was replaced with (ه). In addition, we noted that one of the most frequent ways of using offensive words in Arabic is to begin a phrase with (يا - ya), followed by the offensive word. Many social media posters do not insert a space inside this phrase, so it can be identified as a single word. This is a problem that even the most state-of-the-art tools such as Farasa (Abdelali et al., 2016) and MADAMIRA (Pasha et al., 2014) cannot treat. We dealt with this problem by employing RegEx to split any strings beginning with (ya) into two words. This method requires further improvement in order to be able to deal with words like 'Yasmine' or 'Yafa'.

3.2 Combined Character and Word Embeddings Model (CWE)

One central method that has been recently applied to Natural Language Processing tasks is word embeddings (Devlin et al., 2014; Zhang et al., 2014; Lin et al., 2015; Bordes et al., 2014). Word Embeddings are dense vector representations of text, which capture semantic similarity between words as proximity within the vector space. Encoding takes place for each word in a real-valued vector that has several hundred

dimensions. We employed three different word embedding models, which are detailed in the following subsections. A summary of the important information about each of these models, including their sizes and pre-trained corpus, is presented in Table 1.

Model	Level	Corpus	Size	Dimensions
Ara2Vec	Word	General - Twitter	77M tweets	300
Mazajak	Word	Sentiment - Twitter	250M tweets	300
Character Embeddings	Character	Emotion - Twitter	10M tweets	300

Table 1: Different pre-trained Arabic word embeddings used for our system

Word-level Embeddings: Two Arabic pre-trained word embeddings were employed: Ara2Vec (Soliman et al., 2017) and Mazajak (Abu Farha and Magdy, 2019). Ara2Vec is one of the common open-source word embeddings; it comprises six distinct Arabic word embedding models. The training data for this resource were extracted from Twitter, Wikipedia and web-page crawl data from Common Crawl. Mazajak was also employed, which is regarded as the largest resource for word-level embeddings. This model uses 250 million Arabic tweets to generate the language model. While both Ara2Vec and Mazajak have been trained with a substantial body of words, it is impossible for them to cover every word that may be found in the real world. The Out-Of-Vocabulary (OOV) problem, which renders such resources unable to identify words, represents one of the chief limitations to the word-level model.

Character-level Embeddings: As noted in the introduction, there is a wide variety of Arabic dialects, which can cause OOV problems. This means that effective tools and resources need a better understanding of the many Arab dialects when searching for offensive text. We observed that, while word-level embeddings seem to provide more importance to the semantic similarity, char-level embeddings are more likely to encode every variant of the morphology of the word more closely together within the embedded space. We used a pre-trained character-level language model that achieved outstanding results in different Arabic tasks (Alharbi and Lee, 2020). This model was generated by training the FastText algorithm (Bojanowski et al., 2017) on a large dataset (10 million tweets) containing a variety of emotional and sentimental words in a number of Arabic dialects. Thus it appears that a combination of both levels of embedding could lead to better results.

To this end, Long Short-Term Memory (LSTM) (Hochreiter and Schmidhuber, 1997) was used for the detection of offensive tweets. We concatenated the three pre-trained embeddings mentioned above, and then employed them for initializing the weights of the embedding layer. The embedding layer's weights were then updated during the training to be fine-tuned to the task. We then made a connection to the other layers in the network. We added two layers of 1-D convolutions with 128 filters and a kernel size of 3, followed by a pair of layers of the LSTM network, one with 256 and one with 128 filters. A series of 0.2 dropouts were induced and the 0.2 recurrent dropouts were used for the LSTM layers. Finally, a dense single-output layer was put in place through exploitation of sigmoid as an activation function. The network's optimization function was undertaken using the Adam optimizer.

3.3 Contextualized Embeddings Model (AraBERT)

Because the meaning of some offensive words depends on the context, contextualised language models would be helpful in this task. The Google research team's Bidirectional Encoder Representation from Transformer (BERT) (Devlin et al., 2019) has yielded excellent results for a wide variety of NLP tasks. To obtain better results, we selected the AraBERT model (Antoun et al., 2020) model which was specifically built for the Arabic language. This model achieved state-of-the-art results on the majority of Arabic NLP tasks. AraBERT was pre-trained on a large dataset containing 24 GB of text from a variety of media across the Arab world. We used this pre-trained model to generate embeddings for given tweets. The model employs 12 encoder blocks, 12 attention heads, 768 hidden dimensions, and has a maximum sequence length of 512. A maximum of 512 tokens are presented to the model which then returns a representation for that particular sequence. The initial token in the sequence is the [CLS], containing

specialist classification embedding. The final hidden state of the first token (the sentence embeddings) was passed as an input (768 hidden dimensions) for the classifier model, SVM.

3.4 Affect Model

By 'affect', we mean a range of emotion-related categories, including binary sentiment classification (positive or negative), the strength of sentiment (high positive to low positive) and emotional intensity (e.g. high anger to low anger). We believe that the task of offensive language detection can be further enhanced by transferring learning from other sources of different emotion-related tasks. For instance, some emotions with negative sentiment such as depression, sadness and worry do not increase the likelihood of offensive language, while others, such as anger with high intensity, are more likely to occur within offensive language. While some existing studies exploit sentiment analysis, no research, to the best of our knowledge, leverages emotional intensity for the specific purpose of enhancing the detection of offensive language.

In our system, we employed three affect tasks: sentiment classification, emotion classification and emotional intensity. We assumed that if a given tweet is predicted as negative sentiment and anger emotion with high intensity, the possibility that the tweet may contain offensive language is labelled as 'OFF'; otherwise, it will be labeled as 'NOT_OFF'. We only applied this assumption if the previous two models (CWE and AraBERT) are not agreed on the final prediction (described in the following subsection). We trained the Affect model on the three aforementioned tasks using datasets from SemEval 2018 task 1 (Affect in Tweets) (Mohammad et al., 2018). We selected these datasets due to the variety of affect tasks and Arabic dialects. Finally, we used the knowledge learned from the trained model to predict the affect labels for our target dataset (OffensEval 2020).

3.5 Ensemble

At this step, we now have three models: CWE, AraBERT and Affect. The final output of each model is a prediction for whether each tweet is offensive or not. Integrating multiple kinds of classifiers can help overcome the weaknesses and realise the advantages of each. We used an ensemble technique to combine the classifiers via a majority voting method. Each of these classifiers has a vote (class). Given that two classifers (CWE and AraBERT) were trained on the same task and dataset of OffensEval 2020 then the Affect model's vote only decides the classification is CWE and AraBERT disagree.

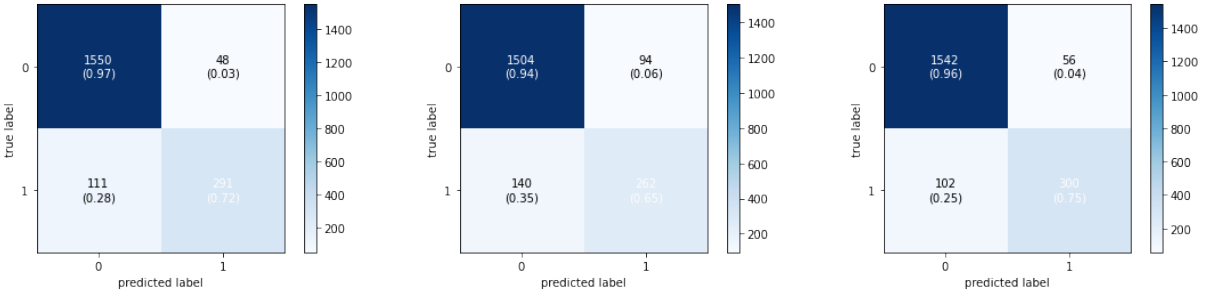
4 Experiment Results

Our models were trained and tested on the training and Dev sets released by the organisers. Details to the data can be found in (Zampieri et al., 2020). The official evaluation metrics for this shared task came from Macro-F1. The results of our models are presented in Table 2. It can be seen that our proposed ensemble model obtained the best performance: an F1-score of 0.90 for the Dev dataset and 0.87 for the Test dataset. From our experiments, the CWE model provided noticeably better results than AraBERT. We believe that this low performance was a result of only extracting the weighted embeddings of AraBERT without fine-tuning the model. Due to computational resources and some technical issues we left fine-tuning the model for future work. The confusion matrixes of the three models are shown in fig 1, which is another way to describe the results.

	Dev dataset				Test dataset			
Model	F1	Accuracy	Precision	Recall	F1	Accuracy	Precision	Recall
CWE	0.88	0,93	**0.90**	0.87	0.86	**0.92**	**0.89**	0.84
AraBERT	0.81	0.89	0.82	0.80	0.81	0.88	0.82	0.80
Ensemble	**0.90**	**0.94**	**0.90**	**0.88**	**0.87**	**0.92**	**0.89**	**0.86**

Table 2: Results for the three models with Dev and Test dataset

Based on analysing the predictions on the Dev set, we observed that each model has its advantages and weaknesses. That is why integrating the three models improved the results by 1%. For example, the



(a) CWE model (b) AraBERT model (c) Ensemble model

Figure 1: Confusion Matrix for the three models

tweet (زود كمان يا عبدالفتاح يا سيسي - give more O Abdel Fattah O Sisi) is incorrectly predicted as 'OFF' by the CWE model and 'NOT_OFF' by the AraBERT and Affect models. The word (Sisi) occurs in several 'OFF' tweets, which is why CWE could not distinguish the context. This is not the same for AraBERT which was better able to capture (Sisi) based on the context. In this example we could also see how the Affect model supported AraBERT to obtain the final correct prediction. In another example, the word (شيطان - devil) can be found in two different contexts (insulting and not insulting). AraBERT was again better able to make a correct prediction than CWE. In contrast, CWE did a lot better on many tweets containing words in a variety of dialects. We think this is because the training data for AraBERT was on MSA, while Mazajak and CE were mainly trained in different Arabic dialects.

In addition, we assessed three pre-trained word embeddings: two word-level models (Mazajak and Ara2Vec) and a char-level model. We engaged in a variety of evaluation exercises during which we compared the performance of each model individually and in combination. The char-level model and Mazajak obtained an F1-score of 0.87, outperforming Ara2Vec (0.86) on the Dev dataset. Although the char-level model was trained on a dataset that consisted of just 10 million tweets, it achieved a result comparable to Mazajak, which was trained on 250 million tweets. However, by combining the three models, we enhanced the accuracy of the results by around 1%. We noted that a char model can adequately address the OOV problem. For example, in Arabic, (الكلبو به - Alklbwbh), meaning female puppy, represents an offensive word. However, the Mazajak and Ara2Vec pre-trained embeddings did not detect this word as potentially offensive. The char-level model adequately captured its meaning by encoding this word close to other related words that either have the same semantic meaning or a predominantly different form of this word.

5 Conclusion

In this work, we proposed an ensemble approach that combines different levels of word embedding models and transfer learning from other sources of emotion-related tasks. The proposed system ranked 9th out of the 52 entries within the Arabic Offensive language identification subtask. In future studies, we hope to enhance the performance of our model by applying additional pre-processing techniques and more effectively exploiting a list of offensive words. Additionally, we will investigate various methods by which data can be augmented into our training datasets to make them more robust.

References

Ahmed Abdelali, Kareem Darwish, Nadir Durrani, and Hamdy Mubarak. 2016. Farasa: A fast and furious segmenter for Arabic. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Demonstrations*, pages 11–16, San Diego, California, June. Association for Computational Linguistics.

Ibrahim Abu Farha and Walid Magdy. 2019. Mazajak: An online Arabic sentiment analyser. In *Proceedings of the Fourth Arabic Natural Language Processing Workshop*, pages 192–198, Florence, Italy, August. Association for Computational Linguistics.

Areej Al-Hassan and Hmood Al-Dossari. 2019. Detection of hate speech in social networks: a survey on multilingual corpus. In *6th International Conference on Computer Science and Information Technology*.

Azalden Alakrot, Liam Murray, and Nikola S. Nikolov. 2018. Towards accurate detection of offensive language in online communication in Arabic. *Procedia Computer Science*, 142:315 – 320. Arabic Computational Linguistics.

N. Albadi, M. Kurdi, and S. Mishra. 2018. Are they our brothers? analysis and detection of religious hate speech in the Arabic twittersphere. In *2018 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM)*, pages 69–76.

Abdullah I. Alharbi and Mark Lee. 2020. Combining character and word embeddings for affect in Arabic informal social media microblogs. In Elisabeth Métais, Farid Meziane, Helmut Horacek, and Philipp Cimiano, editors, *Natural Language Processing and Information Systems*, pages 213–224, Cham. Springer International Publishing.

Wissam Antoun, Fady Baly, and Hazem Hajj. 2020. Arabert: Transformer-based model for Arabic language understanding.

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2017. Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, 5:135–146.

Antoine Bordes, Sumit Chopra, and Jason Weston. 2014. Question Answering with Subgraph Embeddings. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 615–620.

Thomas Davidson, Dana Warmsley, Michael Macy, and Ingmar Weber. 2017. Automated hate speech detection and the problem of offensive language. In *Eleventh international aaai conference on web and social media*, pages 512–515.

Jacob Devlin, Rabih Zbib, Zhongqiang Huang, Thomas Lamar, Richard Schwartz, and John Makhoul. 2014. Fast and robust neural network joint models for statistical machine translation. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 1370–1380.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 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. Association for Computational Linguistics.

Rehab Duwairi and Mahmoud El-Orfali. 2014. A study of the effects of preprocessing strategies on sentiment analysis for Arabic text. *Journal of Information Science*, 40(4):501–513.

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

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 Cyberbullying (TRAC-2018)*, pages 1–11, Santa Fe, New Mexico, USA, August. Association for Computational Linguistics.

Chu-Cheng Lin, Waleed Ammar, Chris Dyer, and Lori Levin. 2015. Unsupervised POS Induction with Word Embeddings. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1311–1316.

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 2017*, pages 467–472, Varna, Bulgaria, September. INCOMA Ltd.

Saif Mohammad, Felipe Bravo-Marquez, Mohammad Salameh, and Svetlana Kiritchenko. 2018. SemEval-2018 task 1: Affect in Tweets. In *Proceedings of The 12th International Workshop on Semantic Evaluation*, pages 1–17, New Orleans, Louisiana. Association for Computational Linguistics.

Hamdy Mubarak and Kareem Darwish. 2019. Arabic offensive language classification on twitter. In *International Conference on Social Informatics*, pages 269–276. Springer.

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*.

Arfath Pasha, Mohamed Al-Badrashiny, Mona Diab, Ahmed El Kholy, Ramy Eskander, Nizar Habash, Manoj Pooleery, Owen Rambow, and Ryan Roth. 2014. MADAMIRA: A fast, comprehensive tool for morphological analysis and disambiguation of Arabic. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, pages 1094–1101, Reykjavik, Iceland, May. European Language Resources Association (ELRA).

Abu Bakr Soliman, Kareem Eissa, and Samhaa R El-Beltagy. 2017. AraVec: A set of Arabic word embedding models for use in Arabic NLP. *Procedia Computer Science*, 117:256–265.

Zeerak Waseem, Thomas Davidson, Dana Warmsley, and Ingmar Weber. 2017. Understanding abuse: A typology of abusive language detection subtasks. In *Proceedings of the First Workshop on Abusive Language Online*, pages 78–84, Vancouver, BC, Canada, August. Association for Computational Linguistics.

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). In *Proceedings of the 13th International Workshop on Semantic Evaluation*, pages 75–86, Minneapolis, Minnesota, USA, June. Association for Computational Linguistics.

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*.

Jiajun Zhang, Shujie Liu, Mu Li, Ming Zhou, and Chengqing Zong. 2014. Bilingually-constrained phrase embeddings for machine translation. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 111–121.