ISSN 1846-6168 (Print), ISSN 1848-5588 (Online)
https://doi.org/10.31803/tg-20220305120018

Subject review
Received: 2021-07-16, Accepted: 2022-03-14

Detecting Arabic Offensive Language in Microblogs Using Domain-Specific Word Embeddings and Deep Learning

Khulood O. Aljuhani*, Khaled H. Alyoubi, Fahd S. Alotaibi

Abstract: In recent years, social media networks are emerging as a key player by providing platforms for opinions expression, communication, and content distribution. However, users often take advantage of perceived anonymity on social media platforms to share offensive or hateful content. Thus, offensive language has grown as a significant issue with the increase in online communication and the popularity of social media platforms. This problem has attracted significant attention for devising methods for detecting offensive content and preventing its spread on online social networks. Therefore, this paper aims to develop an effective Arabic offensive language detection model by employing deep learning and semantic and contextual features. This paper proposes a deep learning approach that utilizes the bidirectional long short-term memory (BiLSTM) model and domain-specific word embeddings extracted from an Arabic offensive dataset. The detection approach was evaluated on an Arabic dataset collected from Twitter. The results showed the highest performance accuracy of 0.93% with the BiLSTM model trained using a combination of domain-specific and agnostic-domain word embeddings.

Keywords: Arabic Natural Language Processing; Arabic Tweets; Offensive Language Detection; Offensive Language; Word Embeddings

1 INTRODUCTION

Warning: This paper tackles the problem of offensive Arabic language in microblogs. So, it may contain some examples that include offensive or vulgar words. These examples do not reflect the authors' perspective in any way.

Over the last years, online social media platforms have become an essential part that influences everyone's daily life. The widespread use of online social media platforms has changed the way people communicate with each other, exchange information, promote products, and evaluate services. Despite the significant benefits provided by social media platforms to individuals and businesses, they are still vulnerable to harmful activities. One of the most detrimental behaviors in social media platforms is offensive language [1]. These platforms enable their users to communicate online anonymously and to express their opinions without barriers, creating an environment where people have the freedom to misbehave and to use obscene words to offend. Therefore, each social media platform was keen to provide policies and guidelines to determine the content that is permitted to be published. Whenever a user posts any content that violates these policies and guidelines, the content will be deleted, or the user's account will be suspended.

However, some users might cross the limits to post content that may violate these policies and guidelines, such as posting intentionally misspelled offensive words, slang, emoticons, or uninformative words. Thus, to maintain the violations as low as possible and ensure that all users can communicate online freely and safely, social media platforms such as Facebook and Twitter have invested in people, processes, and technology to detect offensive and hateful content. But most social media platforms' legal efforts and policies in detecting harmful content and filtering offensive language still heavily depend on traditional channels of reporting misconduct and monitoring by moderators. The manual tracking of offensive and hateful content will be challenging, especially with the massive volume of content on social media platforms these days. So, the automatic detection of offensive language and hate speech on social media platforms has attracted the attention of many scholars. Several studies and competitions emerged in detecting offensive language domain, which clearly emphasizes the growing importance of this issue. However, most of the research on automatic detection of offensive language has focused on rich resource languages such as English, whereas research on this area in Arabic has been rather limited.

The Arabic language is among the most widely used languages on the Internet [2]. A report about the state of Arabic language in social media released in 2018 [3], stated that the number of Arabic users on the Internet reached 237 million users, 17 million tweets in Arabic daily, and 72% of the tweets in the Arab region are in Arabic. According to the New Media Academy Report [4], social media penetration in Arab countries reached 90% of the population in 2020. The report also showed that the average of social media users in the Arab region is represented by 8% of all social media accounts. In Saudi Arabia, for example, there are 25 million social media users, representing 72% of the population [5]. Further, The Arab Youth survey showed that around 90% of young Arab use at least one social media platform every day [6].

Thus, this paper aims to detect Arabic offensive language in Twitter based on textual and contextual features using deep learning models. This paper describes a new approach for constructing a sizeable Arabic corpus for offensive language collected from Twitter. Following this approach, we compiled a data corpus containing more than 500K tweets to extract a domain-specific word embedding. Using a subset of 30K tweets from the data corpus, we then constructed an Arabic offensive language dataset for classification. It also presents a deep learning-based approach to detecting Arabic offensive language using the Bidirectional Long Short-Term Memory (BiLSTM) model, domain-agnostic word embedding (AraVec), and domain-specific word embeddings extracted from an Arabic offensive corpus. To this end, we can summarize the contributions of this study to the field of offensive language detection on social media platforms as follow:

1) Present a large cross-domain and multi-dialect dataset up to date for Arabic tweets that embrace a broad range of offensive and non-offensive tweets for detecting offensive language on Twitter.
2) Propose a new approach for constructing and labeling an Arabic dataset collected from Twitter.
3) Build a domain-specific word embeddings extracted from an Arabic offensive language corpus. To the best of our knowledge, this is the first study that that introduces domain-specific word embeddings for an Arabic offensive language domain.
4) Develop a deep learning-based approach to detect offensive Arabic language, the approach combines the bidirectional Long Short-Term Memory (BiLSTM) model with domain-agnostic word embedding (AraVec) and our domain-specific word embeddings.

The following is how the rest of the paper is structured: Section 2 reviews the previous work on offensive language detection. The approach used in collecting and labeling the dataset is discussed in greater depth in Section 3. The proposed model and experiment implementation are described in sections 4 and 5, respectively. The results are discussed in Section 6 of the paper. Section 7 conclude the study and looks ahead to future works.

2 RELATED WORK

Offensive, as a standalone word, is generally understood, but as a concept, it is broad, complicated, and has different forms and types. As a concept, offensive can be described as discourteous and rude words or comments that lack respect and cause anger or harm [7]. Xu and Zhu [8] defined offensive language as any textual content that might be considered offensive on the grounds of religion, society, culture, or morals, such as sexual, racist, or aggressive content. Jay and Janschewitz [9], identified three categories of offensive language. These three categories are ‘vulgar’ which includes explicit and rude references of a sexual nature, ‘hate,’ which includes offensive comments or words that attack a group of people based on their race, religion, or nationality, and ‘pornographic’.

As aforesaid, the widespread of offensive language in online communication has become an issue especially with the massive increase in using online social communication. Most of this research focused on the issue of offensive language in English and Arabic. However, it's important to note that the research works addressing this problem in the Arabic language are still limited.

2.1 English Studies

Since detecting offensive language in online social platforms is challenging, many supervised machine learning approaches have addressed this problem. Most of these methods extract various types of information from text. Some research [7-14] employed lexical features such lexicon, Bag of words (BoW), N-gram, and Term Frequency-Invert Document Frequency (TF-IDF). For instance, Vandersmissen [7] used BoW and N-gram to detect just two categories of offensive language – sexual and racist – using the techniques of query expansion and text classification. Query expansion is used to increase the overall efficiency of retrieving the relevant messages from the dataset, while text classification separates inoffensive from offensive (sexual or racist) messages. Naive Bayes (NB) and Support vector machine (SVM) classifiers were applied to detect offensive messages. The SVM method outperformed the Naive Bayes algorithm on the validation set, obtaining a precision of 62% versus 9% for the Naive Bayes. In addition, Gaydhani et al. [10] used supervised machine learning to distinguish and detect offensive language and hate speech on Twitter. To detect offensive and hateful tweets, our method used N-gram and TF-IDF characteristics to train three classifier models (Logistic Regression, SVM, and NB). The results revealed that the Logistic Regression model, with an accuracy of 95.6 percent, outperformed the other two models. Shende and Deshpande [11] also used N-gram, TF-IDF, and tokenization to extract features from a dataset collected from Twitter and Facebook. This study proposed a system to recognize offensive material and identify the potential offensive users using SVM and NB classifiers. The SVM system achieved 91.75% accuracy whereas the NB reached 90%.

Although the lexicon-based approaches disregard the syntactical structure of the entire offensive sentence, they performed well in detecting foul language and showed promising results. However, Razavi et al. [12] designed and implemented a novel approach for automatic flame detection, which applies multi-level machine learning classifiers to extract features, boosted by the lexicon of abusive and insulting words. This study demonstrated that the semantic features without the syntactical structure in detecting offensive messages fail to identify the exact insulting comments in different arrangements. Moreover, Davidson et al. [13] proved that using lexical-based methods to detect and separate offensive language and hate speech in online social networks tends to have low accuracy.

Chen et al. [14] suggested a method for detecting offensive content and identifying probable abusive users based on lexical-syntactic characteristics architecture. Because the offensive message cannot be discovered unless it comprises comparable words or expressions that originated from a dictionary, this approach focuses on lexicon to see unacceptable information. It also distinguishes between the use of derogatives and obscenities in detecting offensive content. Moreover, Chen et al. proposed approach integrates style, structure, and context features to detect the user's potential to post abusive messages. The approach showed a higher precision in detecting offensive content by 94.34% and detecting offensive users by 90.2%.

To identify what offensive content should be removed from user messages, Xu and Zhu [8] proposed a technique for semantic filtering based on words' grammatical relations. This method was only concerned with filtering rather than exact identification of offensive remarks. The inflammatory term in the phrase was detected using a word matching algorithm and a vast dictionary of harsh words.

2.2 Arabic Studies

The Arabic studies into the field of offensive language detection is relatively emerging and still limited. Most of previous studies used supervised machine learning algorithms to develop classifiers to detect offensive and harmful content in Arabic social media platforms. Abozinadah et al. [15] trained three classifiers, Naive Bayes, SVM, and Decision Tree, with three sets of features obtained from users' accounts, including user profile features, textual features, and social graph features, to detect abusive Arabic accounts on Twitter. The results of the evaluation showed that the Naive Bayes classifier outperformed other classifiers with an F1-score of 90%. Alakrot et al. [16] employed a dataset that included 15K YouTube comments. They trained an SVM classifier to detect Arabic offensive language in YouTube comments using word N-gram features. The evaluation results showed that the classifier was able to achieve the best F1-score of 82% by integrating these N-gram features and data pre-processing techniques. Mubarak et al.[17] used hashtags and controversial user profiles to create a Twitter dataset. The dataset contains 1100 tweets that were manually classified as obscene, offensive, or clean by three annotators. They also created a list of seed words containing 228 Arabic swear words using a pattern-based search strategy. They classified Twitter users based on whether they were clean or profane using the seed word list. After that, they extracted unigram and bigram to generate a new list of potentially profane words used by aggressive users. They used both lists as features in both internal and extrinsic evaluations to identify tweets as obscene or clean. The results revealed that combining the seed word list with the extended list yielded the best F1-score of 60%.

3 DATA

3.1 Data Collection

The first step toward detecting offensive Arabic language is dataset construction. In this study, we selected Twitter as a source for building a new large multi-domain and multi-dialect Arabic dataset of offensive language. Tweets were extracted during two months from August 01, 2019, to October 01, 2019, using the TweetScraper tool. This tool enables the researcher to crawl tweets from the Twitter search engine [18]. To ensure that the dataset will not be biased to a specific type of offensive, dialect, topics, or targets, we used a blended approach combining two searching strategies: keyword-based and profile-based for extracting tweets and building up the dataset. In the keyword-based method, we explored three publicly available sources containing 404 offensive words and hashtags in Arabic to identify the prevalent offensive and obscene words and phrases that will be worked as a seeding list for our search. These sources are DataWorld dataset contained 37 dirty words in Arabic, HateBase lexicon, which included 79 hateful and offensive terms in the Arabic language, and a list of 288 abusive Arabic words and 127 Hashtags provided by the study Mubarak. et al. [17]. To ensure that the keywords selected from these sources are unbiasedly identifying the offensive language in Arabic tweets, we applied the following criteria: excluding hashtags; terms must be shared between the three sources and written in casual spoken language. Also, terms must not represent a specific Arabic dialect, e.g., the Egyptian or Iraqi dialect. After applying these criteria, we used 81 terms as searching seeds and performed 17 tweets crawling processes, in which we collected 4000 tweets for each crawling. We collected 68k tweets and 38943 users. In the profile-based approach, we randomly selected 1700 unique users who crawled using offensive keywords. One hundred users were selected from each crawling and collected their tweets within the same period that the offensive tweets were collected. We retrieved 525599 tweets after excluding the repeated tweets that were already collected using search seeds. In total, we collect 593599 tweets.

3.2 Data Cleansing

From the extracted data, we randomly sampled a dataset of 30K tweets for cleaning and labeling. In the cleaning stage, we cleaned the dataset by removing the following occurrences:

- Repeated tweets that have the same tweet ID.
- Short tweets that have less than three words, such as ‘private’.
- The less informative tweets that have no meaning, such as ‘Hhhhh.’

3.3 Data Labeling

After the data cleaning stage, now the Arabic offensive dataset contained 29901 tweets. We utilized Mango DB and Mongo DB Campus to label Tweets manually. Tweets were given one of two classes, offensive and non-offensive.

Offensive: Tweets include explicit or implicit insults, cursing, and obscene words intended to attack someone or a subgroup of people. The offensive label has Vulgar tweets that include explicit and rude references of a sexual nature; Hate tweets that include offensive comments or words that attack a group of people based on their race, religion, or nationality; Tweets that include insult and pejorative terms such as: call a person with an animal name, cursing; Tweets that mock the disabilities and shortcomings; Tweets that attack ethics and morals; and pornographic related tweets.

Non-Offensive: Tweets do not include any vulgar or offensive terms. We noticed that a keywords match method without considering tweet context would fail in some cases because some tweets may contain some harsh words. However, the whole tweet could not be regarded as offensive due to the tweet context and the users' intent. We considered the tweet's context and searched Twitter to find how actual users used terms to annotate the ambiguous tweets. Tab. 1 shows examples of vague tweets.

To assess the reliability of manual labeling, we selected a random sample of 10K tweets to be validated via a crowdsourcing platform. For this task, we used Appen (formerly known as Figure Eight), a well-known crowdsourcing platform, to collect, improve, annotate, label, and validate the data to make it practical for machine learning training [19]. In the crowdsourcing validation task, contributors will validate whether the tweet represents the assigned label or not. To make the label validation process more accessible, we developed a labeling guideline for offensive Arabic language. We ensured that our approach was compatible with those provided in [20, 21]. The inter agreement average was 83%.

4 THE PROPOSED MODEL

To investigate the detection of offensive Arabic language on Twitter, we proposed a novel model that utilizes a deep learning-based model with domain-specific word embeddings. Using domain-specific word embeddings helps to capture the most terms used in the offensive context and the intentionally spelling mistakes that are undetectable by general word embeddings since they trained on textual data without spelling errors such as Wikipedia. This section describes in detail our proposed model for detecting offensive Arabic language on Twitter. The primary purpose of this detection model is to process the collected tweets and extract a domain-specific word embedding, and then build a deep learning model (BiLSTM) to classify offensive and clean tweets.

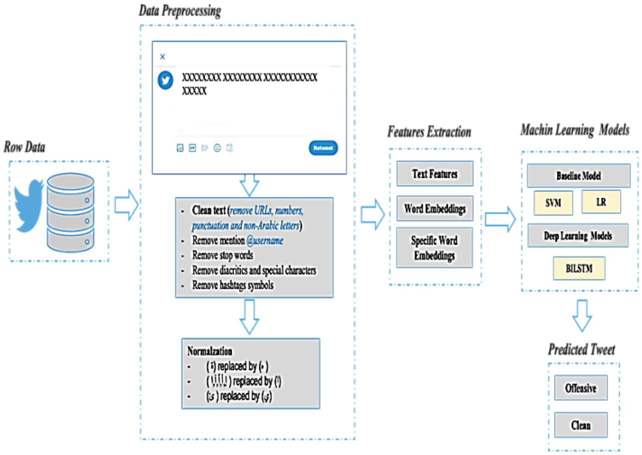


Figure 1 The proposed model for Arabic Offensive Language Detection.

As shown in Fig. 1, the model starts with a dataset labeled as offensive and clean tweets collected from Twitter. Then, we conduct several pre-processing techniques to prepare the dataset for the model. After that, the model extracts text and word embeddings features. Finally, it evaluates three machine-learning models to classify tweets into offensive and clean tweets.

4.1 Data Pre-processing

Text pre-processing is an important stage for text classification tasks. The texts crawled from social media platforms such as Twitter are unstructured and have conversational and noisy nature. So, applying some text pre-processing steps is vital before starting to feed data to the classification models. For Arabic tweets, text pre-processing becomes even more crucial because of the variety of Arabic dialectal used. Arabic tweets usually include tags, punctuations, URLs, symbols, and un-Arabic characters, which we want to remove from the dataset. We performed several pre-processing steps to our dataset. These steps include:

1) **Cleaning:** This step includes removing URLs, non-Arabic letters, numbers, punctuation. The cleaning action also comprises removing the word elongation (kashida), diacritics (tashkeel), and special characters.
2) **Remove Stop Words:** they are the most common words in the data which occur excessively and usually do not provide meaningful information for text classification, such as prepositions, articles, and conjunctions. We used the Arabic stop words list provided by Natural Language Toolkit (NLTK) in this study to remove stop words [22].
3) **Remove Hashtags:** In this step, only hashtags symbols were removed while keeping the keywords because they may represent contextual information.
4) **Normalization:** All Arabic characters that appeared invariants were rendered into a single stat in this step. For example, 'T marbotah' (ة) replaced by (ه), "hamza" on letters (ا ,آ ,ٱ ,أ ,إ) replaced by (ا), and (ئ) replaced by (ي).

4.2 Features Extraction

4.2.1 Text Features

For our baseline experiments, we adopted a variety of feature extraction techniques from the text, and we extracted several combinations of word n-gram and character n-gram features. We used the Term Frequency Inverse Document Frequency (TF-IDF) value to normalize all n-gram features. TF-IDF value reduces the effect of the less informative tokens that frequently appear in the dataset.

4.2.2 Domain-Specific Word Embeddings

A domain-specific word embedding is word representations extracted from a data corpus of a specific domain such as sports and politics domains. As mentioned previously, removing word embeddings from the Arabic offensive language domain helps capture the words usually used in offensive contexts. To extract a domain-specific word embedding, we used the corpus of more than 500K tweets, which was described, in section 3.1. After that, we performed some pre-processing techniques to clean the data and remove un-related words. However, we only applied standard techniques to remove the noise, and we did not handle the misspelling mistakes since some words are abbreviated or intentionally misspelled to avoid the detection models. After removing the noise and processing the data, we used the Arabic offensive tweets to train a continuous bag of word (CBOW) Word2Vec model to extract the embedding features. The CBOW model uses the neighboring words to detect the target word. We trained the model by using

Gensim library with a five-word window and a 300-vector size. We refer to our domain-specific embedding in this study as Arabic offensive Word2Vec (ArOffW2V).

The differences between our domain-specific word embeddings model (ArOffW2V) and the AraVec embeddings model [23]. The analysis results showed that the word's similarities that the AraVec model provides have more general meaning, while our domain-specific word embeddings model provides words that tend to be more related to offensive context. For example, the term ' كلب dog' in the AraVec embedding model referred in general to the name of an animal while in domain-specific embedding model, we noticed that words like "يا ابن الكـ*ـلب son of a dog " appeared which mostly tend to be used in an offensive context. Moreover, it can be shown that the intentionally misspelled words such as (عبـد- nig*er) appeared in the domain-specific embedding model while didn't appear in the AraVec model.

4.3 Supervised Machine Learning Models

In this study, we adapted two supervised machine learning classifiers as baseline models: Support Vector Machin (SVM) and Logistic Regression (LR).

4.3.1 Support Vector Machin (SVM)

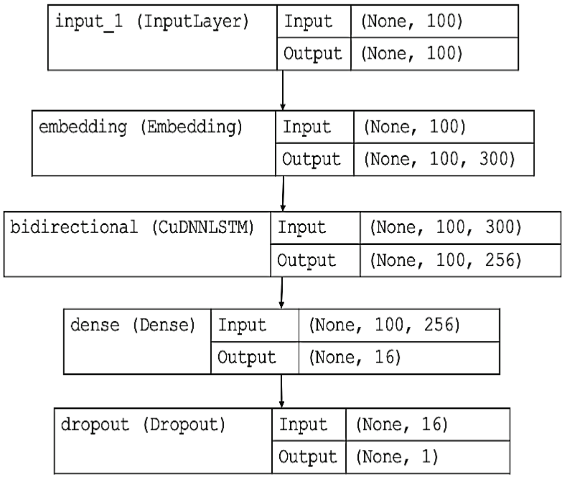
It is one of the most popular supervised machine learning classifiers that can be used for classification or regression. So, it can classify both linear and non-linear data. SVM algorithm outputs the best decision boundary, which is known as hyperplane, to separate n-dimensional space into classes using different kernel functions. For 2-dimensional space, the hyperplane is represented by a straight line dividing the plane into two parts, wherein each class place on either side [24]. The Linear kernel was applied in this study, which can be calculated as shown Eq. (1) where x_1, x_2 are the input of space vector.

4.3.2 Logistic Regression (LR)

It is another supervised machine learning algorithm used for binary classification tasks. It is a statistical model used to measure the correlation between the dependent and independent variables. It transforms the class (dependent variable) from categorical to numeric.

4.4 Deep Learning Model (BiLSTM)

We conducted a set of experiments to build the most suitable deep learning-based model for the Arabic offensive language detection task. From our experiments, we employed a Bidirectional LSTM (BiLSTM), a sequential processing model containing four layers [25]. The main reason that the BiLSTM is suitable for this task is that the model runs the input sequence in two ways, backward and forward, unlike LSTM, which only runs the input sequence backward. So, it's able to understand more information about the context. Fig. 2 shows the construction of our BiLSTM model; the model consists of four layers: embedding layer, bidirectional CuDNNLSTM, dense layer, and dropout layer.



input_1 (InputLayer)	Input	(None, 100)
	Output	(None, 100)

embedding (Embedding)	Input	(None, 100)
	Output	(None, 100, 300)

bidirectional (CuDNNLSTM)	Input	(None, 100, 300)
	Output	(None, 100, 256)

dense (Dense)	Input	(None, 100, 256)
	Output	(None, 16)

dropout (Dropout)	Input	(None, 16)
	Output	(None, 1)

Figure 2 The Structure of BiLSTM Model

5 EXPERIMENTS

Many experiments have been conducted on our created dataset to obtain strong classification results for offensive Arabic language on Twitter. We used a binary classification task in all our experiments to classify tweets into offensive and non-offensive classes (clean). The dataset, baseline models, and performance evaluation measures are all described in the following sections.

5.1 Dataset

In all experiments, we used a random sample from our created dataset. The data sample contains almost 30k tweets labeled manually, either offensive (offensive) or non-offensive (Clean). The dataset sample was described in section 3.2 and section 3.3.

$$K(x_1, x_2) = x_1 \cdot x_2 \qquad (1)$$

Since our data sample is relatively imbalanced, we used the Stratified sampling method to prepare our data for the binary classification task. When splinting the dataset into training and testing sets, this sampling method ensures an equal class distribution. As a result, we used this strategy to divide our data into 70% training and 30% testing sets. Tab. 1 shows the distribution of offensive and clean tweets in training and test datasets.

Table 1 Offensive and Clean Tweet in Training Set and Test Set

Class	Train	Test
Offensive	11,720	2939
Clean	12,199	3032

5.2 Baseline Models

As a baseline, we employed the Support Vector Machine (SVM) and the Naive Bayes (NB) classifiers from supervised machine learning. We tested the two classifiers with distinct *n*-gram features on a word and character level. Supervised machine learning models utilizing word *n*-gram and character n-gram features performed remarkably well in

Arabic offensive language detection tasks, according to various research investigations [7, 18]. The results of our experiments showed that the character n-gram (n = 2 – 5) achieved the highest F_1-score. Because this feature can detect alternative spelling which are common in online communication, it also helps detect a word's morphological makeup.

5.3 Evaluation Metrics

For performance evaluation, we computed true negatives (TN), true positives (TP), false negatives (FN), and false positives (FP) by comparing predicted and actual classes. Then we calculated average precision (P), recall (R), F-measure (F), and accuracy (A) as in the following equations:

$$Recall\,(R) = \frac{TP}{TP + FN} \tag{2}$$

$$F_1\ Score\,(F) = \frac{2 \times (P + R)}{P + R} \tag{3}$$

$$Precision\,(P) = \frac{TP}{TP + FP} \tag{4}$$

6 RESULTS

In this section, we discuss the experimental results of our models. It shows the performance results of the baseline models and the deep learning based-model (BiLSTM) with domain-specific word embeddings. The results are presented in terms of precision, recall, F_1-score. Moreover, we evaluated our BiLSTM model on two different datasets for the offensive Arabic language to generalize our results. We used Mubarak et al. Dataset [17], and abozinadah et al. Dataset [15].

6.1 Baseline Performance

As we mentioned previously, we evaluated two supervised machine learning models on word-n-gram and character n-gram features. From Tab. 2, we can observe that LR on the char n-grams (n = 2 – 5) performed the best overall performance of 92% and is also the best macro precision. On the other hand, SVM was trained on the same set of features and achieved an overall performance of 90%. In terms of features, we can see that the word n-gram (1-4) reached the second-best performance among all models on detecting Arabic offensive tweets.

Table 2 Evaluation Results of The Supervised Machine Learning Models

	SVM			*LR*		
	P	*R*	F_1	*P*	*R*	F_1
Char n-grams (2-5)	0.90	0.90	0.90	0.92	0.92	0.92
Word n-grams (1-4)	0.89	0.90	0.90	0.90	0.90	0.90

6.2 BiLSTM Model Performance

In Tab. 3, we present the performance of our proposed model from a feature perspective of three embedding models (AraVec, ArOffW2V, blending model of AraVec, and ArOffW2V) under the same classifier (BİLSTM).

Table 3 Evaluation results of BiLSTM Model

Dataset	Features	BiLSTM		
		P	*R*	F_1
OUR Data	ArOffW2V	0.91	0.91	0.91
	AraVec (CBOW)	0.90	0.90	0.90
	Blend Embeddings	0.93	0.93	0.93
Mubarak Dataset [17]	ArOffW2V	0.81	0.81	0.81
	AraVec (CBOW)	0.89	0.89	0.89
	Blend Embeddings	0.90	0.90	0.90
Abozinadah Dataset [15]	ArOffW2V	0.86	0.85	0.86
	AraVec (CBOW)	0.87	0.88	0.87
	Blend Embeddings	0.90	0.89	0.90

Among all embedding models, the blending model of AraVec and ArOffW2V achieved the best performance in detecting Arabic offensive tweets on our dataset. However, ArOffW2V performs better than AraVec on the same dataset.

$$Accuracy\,(A) = \frac{TP + TN}{TP + FP + TN + FN} \tag{5}$$

The performance of all embedding models on our datasets was very close to each other. Moreover, we can perceive that training the BiLSTM model with domain-specific embeddings on other datasets achieved high-performance results. These results were consistent with the results obtained when we trained the model in our dataset. To this end, we can conclude that using a specific domain word embeddings model could improve the performance of the Arabic offensive language detection model.

6.3 Error Analysis

This section presents an analysis of the top misclassification errors of the Arabic offensive language detection model. In this study, we were interested in achieving a high detection model accuracy for the Arabic offensive language detection task. Besides that, we were interested in detecting all the offensive Arabic tweets. Thus, we are trying to find misclassified tweets by our model and understand why the model failed at classifying these tweets.

We found that our model misclassified a total of 375 tweets. Most of the misclassified tweets were offensive tweets mis-predicted as clean tweets.

The tweets context and dialects are the reasons for the classification error in these tweets. These tweets have offensive terms in a natural context and dialectal terms that might confuse the classifiers and cause classification errors. However, we can conclude that detecting Arabic offensive language is still challenging and highly dependent on the context.

7 CONCLUSION

Offensive language has grown as a significant problem with the increase in online communication and the popularity of social media platforms. The main purpose of this study is to develop a deep learning-based model to detect Arabic offensive language. In this study, we built a multi-dialect and multi-domain Arabic dataset for detecting offensive language on Twitter. From this dataset, we extracted domain-specific word embeddings from the Arabic offensive language domain to capture the intentionally misspelled phrases that are usually used in offensive contexts. Combining the domain-agnostic word embeddings model with domain-specific word embeddings provides the best performance with the BİLSTM classifier.

For future work, we can explore the type of offensive by labeling the dataset with multi-classes. In addition, it would be interesting to utilize the domain-specific word embeddings with other neural network models for Arabic offensive language detection.

8 REFERENCES

[1] Moor, P. J., Heuvelman, A., & Verleur, R. (2010). Flaming on YouTube. Computers in Human Behavior, 26(6), 1536–1546. https://doi.org/10.1016/J.CHB.2010.05.023

[2] Statista. (2021) Internet: most common languages online 2020. Retrieved December 13, 2021, from https://www.statista.com/statistics/262946/share-of-the-most-common-languages-on-the-internet/

[3] Arabic Language Report. (2020). Retrieved November 18, 2021, from https://www.mcy.gov.ae/ar/arabic-language-report/

[4] 2020 Annual Social Media Report - New Media Academy. (2021.). Retrieved July 6, 2021, from https://newmediaacademy.ae/en/2020-annual-social-media-report/

[5] Man, Z., Ebadi, A. G., Mostafavi, S. M., & Surendar, A. (2019). Fuel oil characteristics and applications: economic and technological aspects. *Petroleum Science and Technology, 37*(9), 1041-1044. https://doi.org/10.1080/10916466.2019.1570256

[6] Radcliffe, D. & Abuhmaid, H. (2020). Social media in the Middle East 2019. *SSRN Electronic Journal*. https://doi.org/10.2139/ssrn.3517916

[7] Vandersmissen, B. (2012). Automated detection of offensive language behavior on social networking sites. *Master thesis*, Ghent University. https://libstore.ugent.be/fulltxt/RUG01/001/887/239/RUG01-001887239_2012_0001_AC.pdf

[8] Xu, Z. & Zhu, S. (2010). Filtering offensive language in online communities using grammatical relations. *The 7th Annual Collaboration, Electronic Messaging, Anti-Abuse and Spam Conference, CEAS 2010*.

[9] Jay, T. & Janschewitz, K. (2008). The pragmatics of swearing. *Journal of Politeness Research, 4*(2), 267-288. https://doi.org/10.1515/JPLR.2008.013

[10] Gaydhani, A., Doma, V., Kendre, S., & Bhagwat, L. (2018). Detecting hate speech and offensive language on twitter using machine learning: An N-gram and TFIDF based approach. ArXiv.

[11] Khazaal, J. A., Hasan, H. F., & Khalbas, H. N. (2021). A study of the market reaction to CEO change. *Economic Annals*, XXI, 187. https://doi.org/10.21003/ea.V187-20

[12] Pustokhina, I., Seraj, A., Hafsan, H., Mostafavi, S. M., & Alizadeh, S. M. (2021). Developing a Robust Model Based on the Gaussian Process Regression Approach to Predict Biodiesel Properties. *International Journal of Chemical Engineering, vol. 2021*, Article ID 5650499, 12 p. https://doi.org/10.1155/2021/5650499

[13] Davidson, T., Warmsley, D., Macy, M., & Weber, I. (2017). Automated hate speech detection and the problem of offensive language. Proceedings of *the 11th International Conference on Web and Social Media, ICWSM 2017*, 512-515.

[14] Chen, Y., Zhou, Y., Zhu, S., & Xu, H. (2012). Detecting offensive language in social media to protect adolescent online safety. Proceedings - *2012 ASE/IEEE International Conference on Privacy, Security, Risk and Trust and 2012 ASE/IEEE International Conference on Social Computing, SocialCom/PASSAT 2012*, 71-80. https://doi.org/10.1109/SocialCom-PASSAT.2012.55

[15] Abozinadah, E. A., Mbaziira, A. V., & Jones, J. H. J. (2015). Detection of Abusive Accounts with Arabic Tweets. *International Journal of Knowledge Engineering-IACSIT, 1*(2), 113-119. https://doi.org/10.7763/ijke.2015.v1.19

[16] Alakrot, A., Murray, L., & Nikolov, N. S. (2018). Towards Accurate Detection of Offensive Language in Online Communication in Arabic. In Procedia *Computer Science, 142*, 315-320. https://doi.org/10.1016/j.procs.2018.10.491

[17] Mubarak, H., Darwish, K., & Magdy, W. (2017). Abusive Language Detection on Arabic Social Media. Proceedings of the *First Workshop on Abusive Language Online*, 52-56. https://doi.org/10.18653/v1/w17-3008

[18] PyPI. (2018). Tweetscraper. Retrieved May 1, 2021, from https://pypi.org/project/tweetscraper/1.2.0/

[19] Appen. (2020). Confidence to Deploy AI with World-Class Training Data. Retrieved October 12, 2020, from https://appen.com/

[20] Mubarak, H., Rashed, A., Darwish, K., Samih, Y., & Abdelali, A. (2020). Arabic Offensive Language on Twitter: Analysis and Experiments. https://arxiv.org/abs/2004.02192v3

[21] Roozitalab, A. (2022), Employing strategic management to study the effect of brand awareness on customer's loyalty: Exploring the mediation effect of perceived brand quality and brand communication: A study of Samsung Electronics Company in Tehran branch. *SMART Journal of Business Management Studies, 18*(1), 38-46. https://doi.org/10.5958/2321-2012.2022.00005.7

[23] GitHub - bakrianoo/aravec: AraVec is a pre-trained distributed word representation (word embedding) open source project which aims to provide the Arabic NLP research community with free to use and powerful word embedding models. (n.d.). Retrieved October 28, 2021, from https://github.com/bakrianoo/aravec

[24] Hsu, C.-W. & Lin, C.-J. (2002). A Comparison of Methods for Multiclass Support Vector Machines. In *IEEE Transactions on Neural Networks, 13*(2). https://doi.org/10.1109/72.991427

[25] Hochreiter, S. & Schmidhuber, J. (1997). Long Short-Term Memory. Neural Computation, 9(8), 1735-1780. https://doi.org/10.1162/NECO.1997.9.8.1735

Authors' contacts:

Khulood O. Aljuhani
(Corresponding author)
Information Systems Department, Faculty of Computing and Information
Technology, King Abdulaziz University, Jeddah, Saudi Arabia
E-mail: kaljuhani0042@stu.kau.edu.sa

Khaled H. Alyoubi
Information Systems Department, Faculty of Computing and Information
Technology, King Abdulaziz University, Jeddah, Saudi Arabia
E-mail: Kalyoubi@kau.edu.sa

Fahd S. Alotaibi
Information Systems Department, Faculty of Computing and Information
Technology, King Abdulaziz University, Jeddah, Saudi Arabia
E-mail: fsalotaibi@kau.edu.sa