Discover Internet of Things

Discover

Research

arHateDetector: detection of hate speech from standard and dialectal Arabic Tweets

Ramzi Khezzar[1] · **Abdelrahman Moursi**[1] · **Zaher Al Aghbari**[1]

Received: 31 December 2022 / Accepted: 6 March 2023
Published online: 20 March 2023

Abstract
Hate speech has become a phenomenon on social media platforms, such as Twitter. These websites and apps that were initially designed to facilitate our expression of free speech, are sometimes being used to spread hate towards each other. In the Arab region, Twitter is a very popular social media platform and thus the number of tweets that contain hate speech is increasing rapidly. Many tweets are written either in standard, dialectal Arabic, or mix. Existing work on Arabic hate speech are targeted towards either standard or single dialectal text, but not both. To fight hate speech more efficiently, in this paper, we conducted extensive experiments to investigate Arabic hate speech in tweets. Therefore, we propose a framework, called arHateDetector, that detects hate speech in the Arabic text of tweets. The proposed arHateDetector supports both standard and several dialectal Arabic. A large Arabic hate speech dataset, called arHateDataset, was compiled from several Arabic standard and dialectal tweets. The tweets are preprocessed to remove the unwanted content. We investigated the use of recent machine learning and deep learning models such as AraBERT to detect hate speech. All classification models used in the investigation are trained with the compiled dataset. Our experiments shows that AraBERT outperformed the other models producing the best performance across seven different datasets including the compiled arHateDataset with an accuracy of 93%. CNN and LinearSVC produced 88% and 89% respectively.

Keywords Hate speech · Arabic · Twitter · Machine learning · Deep learning

1 Introduction

Recently, we noticed an astronomical growth in the use of social media. Many users tend to spend multiple hours a day on these platforms because of the value they get from it [1]. Probably the most important feature is the ability to freely express personal opinions without fear or intimidation. And with it, hate speech and abusive language have become a common phenomenon on social media of different languages including Arabic. It can be a reason of "cyber conflict", which may affect social life. Hate speech is defined in the Cambridge Dictionary as "public speech that expresses hate or encourages violence towards a person or group based on something such as race, religion, sex, or sexual orientation" [2].

Social media has been widely used in the Arab world. It has allowed users of these platforms to exercise their right of freedom of speech. However, because of this freedom of speech and several other reasons, it has also become much easier to spread hate speech and abusive comments. This attracted many researchers to build systems that can detect such hate speech or abusive comments in order to protect users of social media. However, the detection of hate speech on social media, such as Twitter, is a challenging task, particularly for non-English speaking users since there is no clear

✉ Zaher Al Aghbari, zaher@sharjah.ac.ae; Ramzi Khezzar, u18105670@sharjah.ac.ae; Abdelrahman Moursi, u18100789@sharjah.ac.ae | [1]Department of Computer Science, University of Sharjah, Street, Sharjah, UAE.

definition of what constitutes hate speech. Some words and statements appear to be obscene in one region, but might be totally acceptable in another.

Most of the hate speech detection research and works focus on the English language. Based on [3], Arabic language is ranked fourth among the top used languages on the web and sixth among the languages used on Twitter. However, Arabic is a rich morphological language that has complex grammar and structure that poses great challenges to automatic processing of Arabic text [4]. Moreover, Arabic has many dialects, where people of each Arab country speak a different dialect from people of other Arab countries. These dialects have morphological differences from each other [5].

Due to the above mentioned characteristics, Arabic language has relatively much less research in comparison with the English language. Moreover, the performance of the available trained models do not match the superiority of the English ones. The reason is that Arabic language has many dialects and most users of social media feel more comfortable expressing their feelings using their dialects. Existing hate detection works are either assume the social media posts are written in standard Arabic, or the use of one dialect. Therefore, it is essential for a hate detection system to support multi-dialect language in order to effectively identify hate phrases in postings. Another reason for the low performance of existing systems is the lack of large datasets that include hate postings, which is necessary for the effective training of the detection models, such as deep learning tools.

To this end, this paper proposes an Arabic hate speech detector from tweets, called arHateDetector. The proposed arHateDetector supports both standard and dialectal Arabic text, which makes arHateDetector very effective and highly accurate detector of Arabic hate speech. To help build highly accurate classifier model of hate speech, we compiled a large Arabic hate speech dataset, called arHateDataset, to train the classifier models. We intend to make the arHateDataset public for researchers. A comprehensive set of experiments are conducted to evaluate the machine learning models such as Linear SVC, Random Forest and logistic regression, in addition to deep learning models like convolutional neural networks (CNNs) and AraBERT. Moreover, a web application user interface is built to allow users test the proposed arHateDetector.

The main contributions of this work are:

- Proposing arHateDetector, which a framework for Arabic hate speech detection from tweets. arHateDetector supports standard and dialectal Arabic text.
- Compiling a large Arabic hate speech dataset, called arHateDataset, of 34,000 tweets, of which 32% are hate tweets and the remaining 68% are normal tweets.
- Assessing the performances of nine ML models and two DL models in the Arabic hate speech detection task.
- Developing a web app user interface for arHateDetector.

The remaining sections of this paper are as follows. Recent related works to Arabic hate speech detection are reviewed in Sect. 2. Collecting and compiling the arHateDataset are presented in Sect. 3. This section also includes the data preprocessing and model construction. Section 4 discusses the results of the conducted extensive experiments. Section 5 presents the web app. Finally, our work is concluded in Sect. 6.

2 Related works

This section presents the recent relevant research works to Arabic hate speech detection from social media posts.

Zampieri et al. [6] evaluated the performance of a proposed model that combines unigram and SVM to the performance of CNN and BILSTM. Based on the F1 score, it was reported that the CNN and BILSTM models outperformed SVM. Davidson et al. [7] trained several basic machine learning models, like Decision Trees, and Logistic Regression, to detect hate speech. The later produced the best F1-macro score based on the their used dataset. Mulki et al. [8] collected dataset of hate speech for the Levantine dialect with more than 5800 labeled tweets. N-gram features were investigated to identify hate tweets. Naïve Bayes model was found to produce the highest F1-score of 89.6%.

Mubarak et al. [9] evaluated different models to classify Arabic hate speech. For their used dataset, the highest F1 score result is 83.2%. This is was produced using AraBERT. On the other hand, Hatem et al. [10] collected social media comments to build a Tunisian dialect dataset of around 6K. SVM was found to produce the highest accuracy of 93% in classifying those comments into hate, abusive, and normal. Alternatively, the work in [11] tested traditional machine learning models to detect hate comments in YouTube.

Unlike the above mentioned works that employed traditional machine learning models, deep learning (DL) models have also used in classifying hate speech. Albadi et al. [12] collected an Arabic religious hate speech dataset of around 6600 labelled tweets, which were used to evaluate their model. As per their findings, the GRU model outperformed LSTM model in classifying religious hate speech with an accuracy of 79%. Social Network Graphs were used to represent hate speech data, which is collected by [12], along with word embeddings by Ghosh et al. [13]. In their experiments, an accuracy of 86% was achieved.

To detect Arabic hate phrases, [14] experimented with different machine learning and deep learning methods and achieved an F1-macro result of 87.03% using CNN and multilingualBERT. In [15], a deep recurrent neural networks model is proposed to classify Arabic hate speech into seven categories, which achieved a classification accuracy of about 84%. The work in [16] used a pre-trained deep learning model (marBERT) to classify hate speech. They showed that multi-task models outperforms single-task ones. The effect of preprocessing text on the detection of hate speech is investigated by [17]. Six preprocessing methods were tested. They showed that models like BERT can have a lower performance due to preprocessing Arabic text, unlike basic machine learning models.

A self-trained model was proposed by [18] to improve the detection accuracy of hate speech. The model utilizes the most confident hate text to iteratively learn from. In [19], the authors investigated the long tail problem of Arabic language data distribution by using loss functions. They evaluated different models and achieved a classification accuracy of around 87%. Religous radicalism text was detected by a model proposed by [20]. This model was trained on 3000 labeled Arabic tweets.

Multi-labeled dataset was used to detect Arabic hate speech by [21]. In this model, manual and semi-supervised annotation of short text was conducted. Detection of hate speech in Leventine Arabic was proposed in [22]. The proposed model was evaluated on several traditional and deep learning models. Deep learning classifiers were shown to outperform traditional machine learning classifiers. In [23], models based contextualized text representation were evaluated in Arabic hate speech detection. MarBERT showed to outperform other evaluated models on a dataset of 13K tweets. Table 1 shows summary of previous works.

However, all the above models either apply their Arabic hate speech models to standard Arabic and/or one, or two, dialectal Arabic. The proposed arHateDetector supports starndard Arabic as well as a range of dialectal Arabic.

Table 1 Comparison of previous research papers best performing classifiers

Paper	Dataset Size	Dialect	Best classifier	Performance (%)
[6]	14,100	English	CNN	80
[7]	24,802	English	Logistic Regression	90%
[8]	5846	Levantine	Naive Bayes	89.6
[9]	10,000	MSA, Gulf, Iraqi, Levantine, Egyptian, Maghreb	AraBERT	83.2
[10]	6039	Tunisian	SVM	93
[11]	11,268	MSA, Gulf, Iraqi, Egyptian	BERT	98
[12]	6600	MSA	GRU	79
[13]	6000	MSA	LSTM + CNN + NODE2VEC	86
[14]	5,361	MSA, Gulf	CNN + multilingualBERT	87.03
[15]	4203	MSA	DRNN	90.3
[16]	32,000	MSA, Tunisian, Levantine	MTL-M-L	92.3
[17]	13,000	MSA	SVM	95
[18]	13,140	MSA	CNN + SGClassifier	89
[19]	13,000	MSA	MARBERTV2 + MARBERT + QARiB	87
[20]	3000	MSA	SVM	92
[21]	44,000	MSA	LinearSVC	92.45
[22]	16,683	Syrian	GigaBERT	94.6
[23]	19,000	MSA, Levantine	CNN	94

Fig. 1 The phases of the proposed arHateDetector

Table 2 Comparison of existing datasets with the compiled dataset

Name of the dataset	Nickname	Balanced	Dialect	#IDs	#tweets
Arabic levantine hate speech detection [24]	Levantine	Yes	Levant	5846	5846
Hate speech detection in Arabic [25]	Raghad	Yes	Saudi	9316	4726
Religious aravic hate speech detection [26]	Nuha	Yes	Gulf	6136	3288
Hate and Offensive speech detection [27]	Sb	Yes	Gulf	5360	3075
Arabic COVID19 Multilabel Fake News and Hate Speech Detection [28]	AraCovid	No	Algerian	10,828	9198
Multi-lingual hate speech [29]	Multi	Yes	Gulf	4621	4621
Multilingual and multiAspect hate speech analysis [30]	Latest	No	Gulf	3353	3353
arHateDetector dataset	arHateDataset	Yes	Multi	45,460	34,107

3 arHateDetector framework

This section introduces the proposed arHateDetector framework. First, it starts by presenting an overview of the arHateDetector framework. Then, the collection of the Arabic hate speech dataset is discussed. Next, the preprocessing steps of the dataset are presented. Finally, ML models and DL models that are applied on the dataset are discussed.

3.1 arHateDetector overiew

The arHateDetector framework consists of five main phases as shown in Fig. 1 that are: Arabic hate speech collection, Arabic text preprocessing, model construction, model evaluation, and model deployment.

1. Dataset collection: Arabic hate speech datasets are gathered from multiple online public datasets of different dialects. These dataset are then integrated and compiled into a unified large dataset, called arHateDataset.
2. Arabic text preprocessing: Arabic tweets are cleaned and preprocessed to remove hashtag and stop words, filter out spam, replace emojis, normalize the text, and lemmatize the text.
3. Model construction: ML and DL models are trained and then used for classification.
4. Model evaluation: The constructed models are evaluated to assess their performance using some existing Arabic hate speech dataset and the compiled arHateDataset.
5. Model deployment: A web application interface for the proposed hate speech detection framework is built to support easy access to arHateDetector.

3.2 arHateDataset

In this section, the dataset collection, compilation and description are presented. Also, a comparison of existing public datasets and the compiled dataset of this paper is presented in Table 2.

3.2.1 arHateDataset collection

To compile a large dataset of Arabic hate speech, we searched for online available Arabic datasets. Multiple dataset exists, however, the size of these datasets is relatively small. That is each dataset does not exceed 10,000 tweets. This

motivated us to compile a significantly larger dataset to effectively train the proposed arHateDetector to achieve reliable experimental results. Therefore, we combined, cleaned, and integrated the existing public datasets into a unified large dataset, called arHateDataset.

Two of these datasets were found on the Kaggle platform, one dataset was acquired by contacting its owner and the remaining four were all acquired from GitHub. Four of these dataset contained only the tweets ids while the other three came with the tweets text. Therefore, the Twitter API is used to hydrate the tweets, which is a user-friendly tool with a GUI interface, called the Hydrator tool.

The first compilation of the datasets reached a total of 45,460 tweets. However, some of those tweets were inaccessible because they have been deleted, taken down by the Twitter company, or the owner account was deleted and no longer active. Thus, the available tweets that we were able to collect and compile are 34,107 and is called arHateDataset. Table 2 shows the size of datasets that were accessible and the size of the compiled arHateDataset.

3.2.2 arHateDataset description

It is crucial that the tweets get cleaned and integrated as one dataset meaning that things like emojis and irrelevant symbols get removed before we train our models on this data. So, before we preprocessed these datasets, we decided to change the labelling of the tweets due to some data being labeled with text while other was labeled with numbers. In order to unify the labeling across all different collected datasets, we used the number "1" to represent the presence of hate in the tweets while for the normal tweets were relabeled using the number "0". Figure 2 shows the percentage of hate tweets compared to normal text in the integrated arHateDataset. Note that 68% of the tweets are normal, while the remaining 32% are hate tweets. Table 3 presents some examples of tweets with their classification "1" for hate speech and "0" for normal speech.

3.3 Dataset cleaning and preprocessing

Cleaning and preprocessing the data are essential steps to arHateDetector. Simply, an effective model cannot be trained without first removing noisy data and irrelevant content that negatively affect the classification of tweets. This process consists of many steps to achieve a desirable outcome [31].

First, using the regular expression python module and the PyArabic library which has many predefined python functions dedicated to process Arabic text, we defined functions to perform the following cleaning steps. We conduct hashtags removal, spam filtering, and emoji replacement. Then, links and mentions are replace by "URL" and "mention", respectively. Next, the tweet text is cleansed by removing the punctuation, diacritics, additional white spaces and non-Arabic characters. After that, stops words are removed, where we can use a list of 356 Arabic stop words made available on GitHub [32].

Then, normalization of words is applied where, for example, the different variations of the same character are replaced. The steps are as follows:

- Different forms of "ا", which are "إ" ,"أ", and "آ", are replaced by "ا".
- "ى" is replaced by "ي".
- "ة" is replaced by "ه".

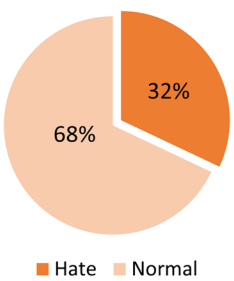

Fig. 2 Ratio of hate tweets vs normal tweets in arHateDataset

Table 3 Examples of Arabic tweets, their translation, and their classification

Arabic tweet	English translation	Class
وحاج حكي مثاليات وامه ومابعرف شو ،اخر هم عندكم	Hajj spoke of ideals and his mother, and I don't know what other worries you have	0
بلدك مرادف الارهاب	Your country is synonymous of terrorism	1
يلعن روحه من الجد لل ولد هات طناجر العرعور والحقني	Curses his soul from the grandfather to son. Bring the pots of Alaroor and follow me	1
يا حمار قمة يعني زبال	O donkey crest means rubbish	1
ما حدا غير يلي خلفتك غبية يا منحط	No one but the one who birthed you stupid, you degenerate	1
اخي الحاج اذا شعرت انك محرج من الانتقادات قوللي	Brother Hajj, if you feel embarrassed by the criticism, tell me	0
وزير الخارجية اللبناني جبران باسيل قال في سلسله	Lebanese Foreign Minister Gebran Bassil said in a series	0

The final step is lemmatization, where the different forms of the word are replaced by the root form. For this purpose, FARASA tool [33] is used. At the end, we obtain a clean text that is ready to be fed into the arHateDetector framework for training and classification.

Figure 3 shows a tweet before and after the preprocessing steps conducted on this tweet.

3.4 Model construction

This section presents the evaluated traditional ML models and the DL models.

3.4.1 Basic machine learning

In the proposed arHateDetector framework, several ML models were trained and tested on the compiled arHateDataset. These machine learning models are:

- Logistic Regression (LR): A type of supervised ML model that is used for classification tasks. It uses a logistic function to model the probability of a data point being in some class.
- Support Vector Classifier (SVC): It is based on the concept of support vectors, which are points in the training data that have the greatest impact on the decision boundary. The algorithm aims to find a hyperplane in the feature space that maximally separates the two classes. SVC can be used for both linear and nonlinear classification problems, depending on the kernel function used.
- Linear Support Vector Classifier (Linear SVC): A variant of SVC that is specifically designed for linear classification tasks. It seeks to find the hyperplane that maximally separates the two classes in the feature space by maximizing the margin between the hyperplane and the nearest support vectors.
- Stochastic Gradient Descent Classifier (SGD): A type of linear classifier that is trained using stochastic gradient descent. It is particularly well-suited for large-scale classification tasks where the data is too large to fit in memory.
- Bernoulli Naive Bayes (Bernoulli NB): A type of probabilistic classifier that is based on the assumption of independence between the features. It is often used for text classification tasks, where the features correspond to the presence or absence of certain words in a document.
- Multinomial Naive Bayes (Multinomial NB): A variant of Naive Bayes that is specifically designed for classification tasks where the features are counts or frequencies. It predicts the class of a new data point based on the frequency of the features in the training data and the likelihood of the features in the new data point given each class.
- Decision Tree (DT): It works by creating a tree-like model of decisions based on the input features. At each node of the tree, the algorithm splits the data based on the feature that maximally separates the two classes. The resulting tree can be used to make predictions about the class of new data points.
- Random Forest (RF): A type of ensemble learning model that is trained by large number of decision trees. Each of these decision trees is trained on subsets of the training data. The aggregation of the predictions of the individual trees constitutes the final prediction.
- K-Nearest Neighbors Algorithm (KNN): It classifies a new data point based on the majority class of the K nearest points in the training set, where K is a user-specified hyperparameter. KNN is simple and effective, but it can be computationally expensive and may not scale well to large datasets. It is often used with other techniques to improve performance.

A pipeline is created that first computes the TF-IDF vectorizer then it uses one of the previous machine learning algorithms mentioned above. We set the min_df to 0.0001 and the max_df to 0.95, and we kept the rest of the parameters as default.

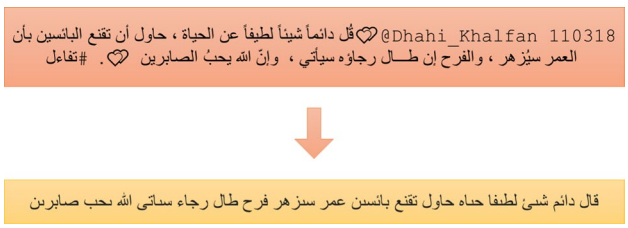

Fig. 3 Example tweet before and after preprocessing

For the first six machine learning algorithms are used with their default settings. However, for decision tree classifier, the max_depth is specified to 20. For random forest classifier, the parameters are set as follows: max_depth to 3, n_estimators to 10, and max_features to 1. Finally, for K nearest neighbors classifier, we set the number of neighbors to 15. These parameters were chosen because they produced the best results after manually trying different combinations.

3.4.2 Convolutional neural network

It is commonly known that convolutional neural networks (CNNs) were intended initially to be used for image classification until a research paper that came out in 2014 [34], which proved that CNN can be used for text classification as well. After that many research papers used it as method on their text datasets. Therefore, CNN is selected to be evaluated by arHateDetector to detect hate tweets. The CNN architecture is shown in Fig. 4, which is used by the proposed arHateDetector framework.

As shown in Fig. 4, the CNN model is composed of five layers. These layers start with an embedding layer. The second layer is a 1-D Convolutional layer that uses the ReLU activation function. The third layer consist of a global max pooling. Finally, the fourth and fifth layer are dense layers. The fourth layer uses a ReLU function, while the fifth layer uses a Sigmoid function. The weights of the CNN model are update using the Adam optimizer.

3.4.3 AraBERT

AraBERT is the Arabic version of Google's BERT model, developed to support the pecularities of Arabic [35]. AraBERT is designed with 110 million parameters that can learn from the input dataset. It is also equipped with 12 self-attention layers [36]. We used the newest version of the model "AraBERTv0.2-Twitter-base", which was pretrained by a huge dataset of around 77 gigabytes of Arabic text. This dataset consists of about 200,000,000 sentences that generated 8.6 billion tokens. Moreover, we chose this model because it is was purposefully tuned using multi dialect data. It was trained on 60,000,000 tweets of multi dialect Arabic, collected and extracted from social media platforms from all regions of the Arab world; from Morocco in the west all the way to the Gulf countries. This is suitable for our arHateDataset as it contains four different dialects. Additionally, emojis and common words were added to the vocabulary that weren't present at first. Figure 5 is an abstract diagram of the implementation process of arHateDetector, which includes AraBERT, CNN, and traditional machine learning algorithms.

4 Performance and results

For arHateDetector, NumPy [37], NLTK [38], Scikit-learn [39] were used to develop the models. Experiments were conducted with Tensorflow [40] and Keras [41].

The models were trained with the arHateDataset after being preprocessed. The preprocessed dataset is split into two parts 75% and 25%, where the 75% part is used for training whereas the 25% part is used for testing. This was performed

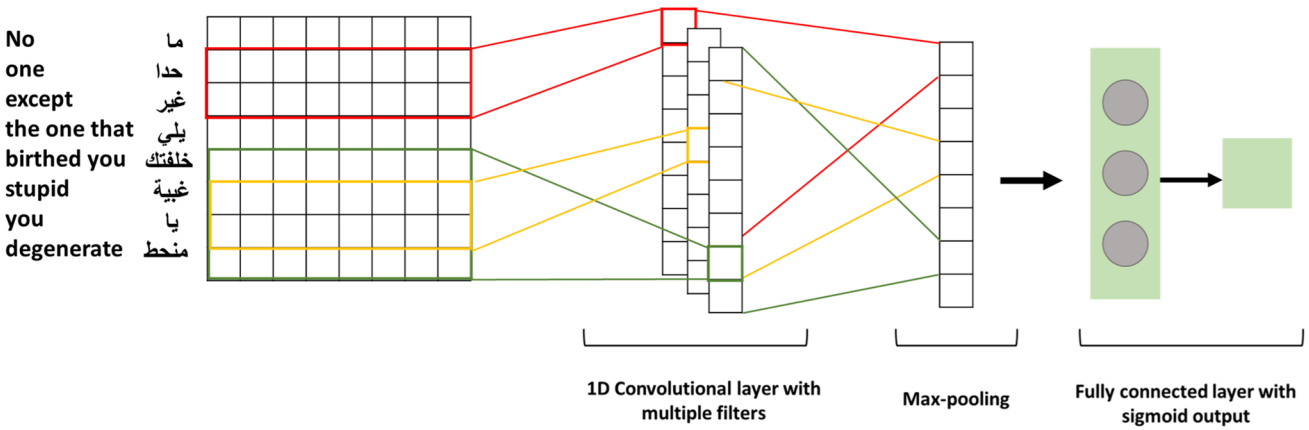

Fig. 4 Convolutional neural network architecture

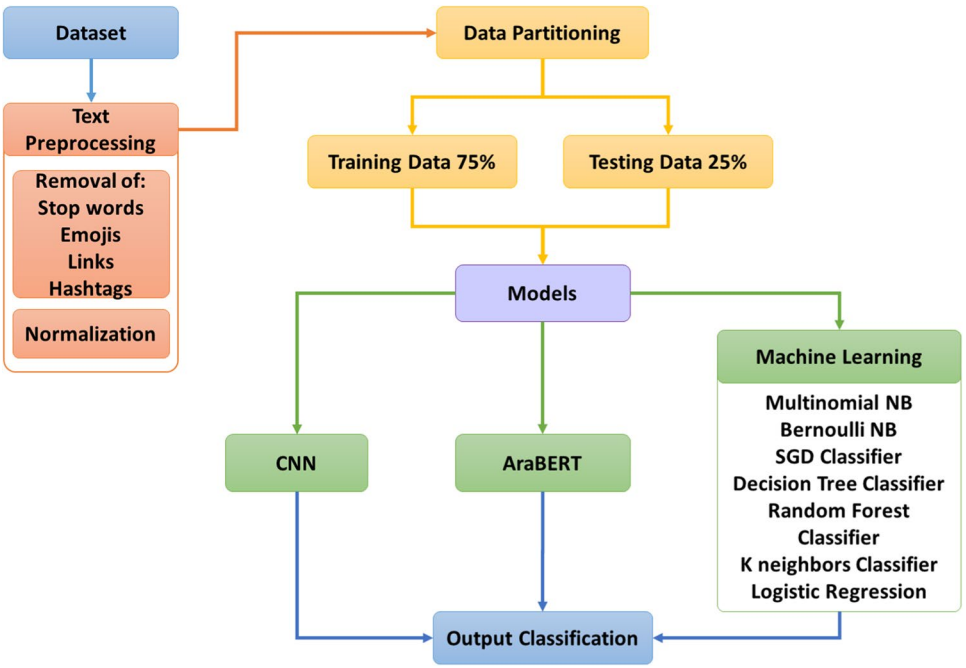

Fig. 5 Abstract diagram of the implementation process of arHateDetector framework

while maintaining the balanced feature in both partitions in order to train and test the model accurately. This can be seen in the arHateDataset files online.

The AraBERT model was already pretrained on MSA and Dialectical Arabic non-hate text, which is available on GitHub [42]. We achieved the best results using the following parameters: learning rate to 2e-5 and Adam epsilon to 1e-8. batch size for training = 16 and epochs = 2.

Four metrics were used to evaluate the performance of the different models. These metrics are: Precision (Eq. 1), Recall (Eq. 2), Accuracy (Eq. 4), and F1-score (Eq. 4). In these equations, True positive (TP) refer to the number of accurately classified hate tweets. True negative (TN) is the number of accurately classified non-hate tweets. False positive (FP) is the number of non-hate tweets misclassified as hate tweets. And, false negative (FN) is the number of hate tweets misclassified as non-hate tweets.

$$Precision = \frac{TP}{TP + FP} \tag{1}$$

$$Recall = \frac{TP}{TP + TN} \tag{2}$$

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{3}$$

$$F1score = 2 * \frac{Precision * Recall}{Precision + Recall} \tag{4}$$

4.1 Basic machine learning performance

As we mentioned above we successfully collected seven datasets. Five of them were already balanced. The aracovid dataset was unbalanced so we downsampled the normal category. And the latest dataset we found contained only hate tweets, therefore it was added to the compiled large dataset to contribute to the balancing of the arHateDataset. To sum up, we tested all models on six datasets in addition to the compiled arHateDataset.

Table 4 Comparison of machine learning models in term of accuracy using all datasets

Algorithm	Levantine	Multi	Nuha	AraCovid	Raghad	Sb	arHateDataset
LinearSVC	**0.83**	**0.83**	**0.83**	**0.81**	**0.81**	**0.83**	**0.89**
SVC	0.80	0.80	0.80	0.75	0.75	0.80	**0.89**
MultinominalNB	0.80	0.80	0.80	0.74	0.74	0.80	0.87
BernoulliNB	0.77	0.77	0.77	0.74	0.74	0.77	0.89
SGDClassifier	0.82	0.82	0.82	**0.81**	**0.81**	0.82	0.87
DecisionTree	0.74	0.74	0.75	0.77	0.77	0.75	0.80
RandomForest	0.60	0.60	0.60	0.73	0.73	0.60	0.67
KNearestNeighbor	0.80	0.80	0.80	0.74	0.74	0.80	0.81
LogisticRegression	0.78	0.78	0.78	0.75	0.75	0.78	0.87

The bold values represent the highest accuracies

Table 4 presents a comparison between all machine learning models using all datasets in terms of accuracy. Notice that Linear SVC is the best performing algorithm followed by SGD classifier. while SVC and Multinomial NB came third. The compiled arHateDataset gave the best result on all models. This is due to the large size of the arHateDataset and the multiple dialects with rich content. As a consequence, arHateDetector was well trained to detect a hate tweets written in various dialects and various writing styles.

4.2 CNN and AraBERT performance

The same datasets, which are used to evaluate the machine learning algorithms, are also used to evaluate the CNN and the AraBERT models. Table 5 show the numerical accuracies of the evaluated CNN and AraBERT models. Clearly, AraBERT outperforms CNN across all datasets. That is due to the fact that AraBERT has more parameter to fine-tune during training, which in turn leads to better performance. Figure 6 illustrates the comparison between the best performing machine learning model, which is LinearSVC, with the two deep learning models, which are CNN and AraBERT.

When comparing AraBERT and CNN with the best performing machine learning algorithm (LinearSVC) in Fig. 6, we noted that the AraBERT model is the best overall. The arHateDataset produced a highest accuracy result of 93%. This is

Table 5 Comparison of deep learning models in term of accuracy using all datasets

Algorithm	Levantine	Multi	Nuha	AraCovid	Raghad	Sb	arHateDet
CNN	0.82	0.79	0.76	0.75	0.78	0.84	0.88
AraBERT	**0.91**	**0.90**	**0.82**	**0.98**	**0.87**	**0.93**	**0.93**

The bold values represent the highest accuracies

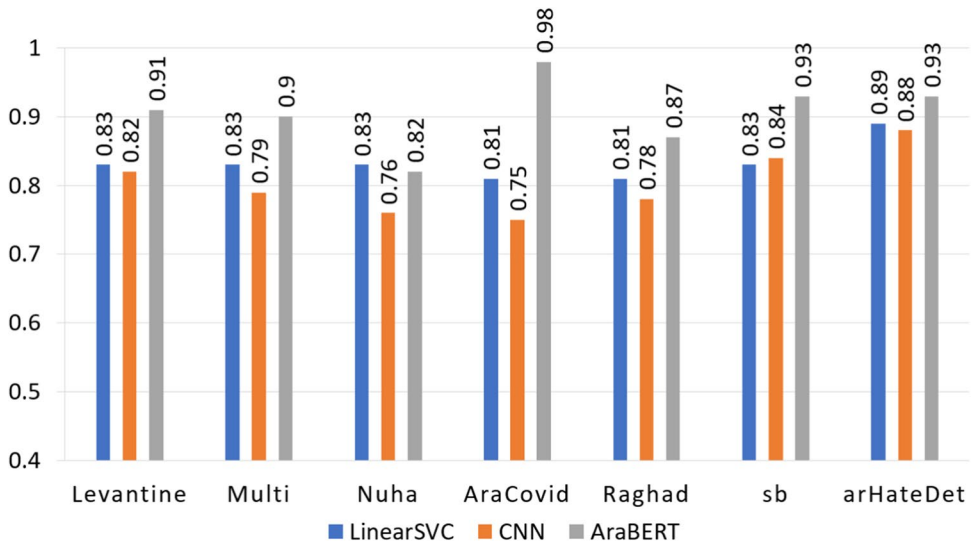

Fig. 6 Comparison between deep learning models with the best performing machine learning algorithm

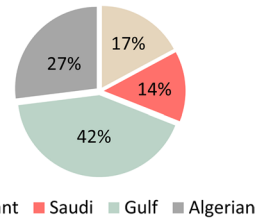

Fig. 7 Percentage of each dialect in the arHateDataset

due to the fact it was trained on a larger multi-dialect tweets corpus. Figure 7 shows the percentage of each dialect in our compiled corpus. The only result that was an outlier is for the araCovid dataset, this is mainly because of the imbalance found in the dataset, to counter that we did down sampling of the normal tweets but since the ratio was 91% normal text to 9% hate text it still couldn't regulate the result.

Upon analysing the performance of our different models across all datasets, we notice that the largest one arHateDataset produced the highest results. When looking at the perfomaces of each dataset and compring it to arHateDatset we see an increase in accuracy of 3–9% for AraBERT, for CNN it is 4–17% while for LinearSVC it was 7–9%. This proves that using a larger multi dialects dataset results in a better performance.

5 arHateDetector App

To help users classify tweets, or any sentence, as hate or normal, a web application was developed. The front end consists of the User Interface, which was developed using HTML5, CSS3, and JavaScript. The User Interface is very simple and is made up of one page where the user can enter text to predict whether it contains hate or not, see Fig. 8.

The back-end server was developed using Flask, which is a web application framework written in Python suitable for small to medium-scale projects. When the user inputs a text sentence, the back-end server passes it through the same pre-processing steps used in training arHateDetector. Afterwards, the model is loaded to the server and the preprocessed text is passed through the model to detect whether it is hate or normal text. Finally, the prediction is passed to the front-end to be presented to the user.

6 Conclusion

In this work, we proposed an Arabic hate speech detection framework, called arHateDetector. Additionally, a large dataset of Arabic hate tweets was compiled for arHateDetector. This dataset contains multiple dialects as well as standard Arabic tweets. A comparison between several traditional machine learning and two deep learning algorithms in classifying hate tweets was conducted in terms of accuracy. Among the traditional machine learning algorithms, it was concluded that

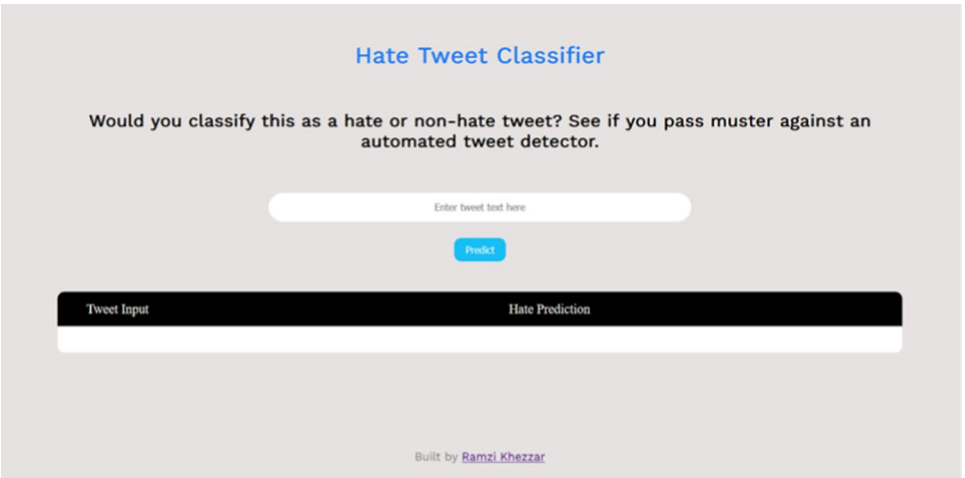

Fig. 8 Web application user interface

linear SVC is the best performer followed closely by SVC, SDG classifier and logistic regression. On ther hand, AraBERT outperformed CNN and LinearSVC in classifying Arabic hate speech across all tested datasets.

It was clear from the results we presented earlier that AraBERT is the best performing model with 93% accuracy for our arHateDataset. A comparison between the used datasets, the compile arHateDataset results in the highest accuracy across all models we evaluated, which is due the richness of its content and the several dialects that it has.

Author contributions implementation, testing, and first draft are conducted by Ramzi and Abdelrahman, The, project ideas, design, and final manuscript is by Zaher. All authors read and approved the final manuscript.

Funding This project has no funding.

Data availability The datasets analysed during the current study are available from the corresponding author on reasonable request. After acceptance of the paper, the author intend to make the compiled arHateDataset public.

Code availability Code available upon request from researchers.

Declarations

Competing interests The author declare that there is no conflict of interest.

References

1. Saeed MM, Al Aghbari Z. Artc: feature selection using association rules for text classification. Neural Comput Appl. 2022;34(24):22519–29.
2. Cambridge-Dictionary https://dictionary.cambridge.org/us/dictionary/english/hate-speech.
3. Statista-Inc: The Most Common Languages on the Internet, https://www.statista.com/statistics/262946/share-of-the-most-common-languages-on-the-internet. 2019.
4. Elzobi M, Al-Hamadi A, Al Aghbari Z, Dings L, Saeed A. Gabor wavelet recognition approach for off-line handwritten arabic using explicit segmentation. In: S. Choras, R. (ed.) Image Processing and Communications Challenges. Springer, Heidelberg 2014; pp. 245–254.
5. Dinges L, Al-Hamadi A, Elzobi M, Al Aghbari Z, Mustafa H. Offline automatic segmentation based recognition of handwritten arabic words. Int J Sign Process Image Processing Pattern Recogn. 2011;4(4):131–43.
6. Zampieri M, Malmasi S, Nakov P, Rosenthal S, Farra N, Kumar R. Predicting the type and target of offensive posts in social media. Proceedings of Human Language Technologies: The Annual Conference of the North American Chapter of the Association for Computational Linguistics (NAACL-HLT), p. 1415-1420. 2019.
7. Davidson T, Warmsley D, Macy M, Weber I. Automated hate speech detection and the problem of offensive language. In: Proceedings of the International AAAI Conference on Web and Social Media, vol. 11; 2017. p. 512–5.
8. Mulki H, Haddad H, Ali CB, Alshabani H. L-hsab: A levantine twitter dataset for hate speech and abusive language. In: Proceedings of the Third Workshop on Abusive Language Online. 2019. p. 111–8.
9. Mubarak H, Rashed A, Darwish K, Samih Y, Abdelali A. Arabic offensive language on Twitter: Analysis and experiments. In: Proceedings of the Sixth Arabic Natural Language Processing Workshop, pp. 126–135. Association for Computational Linguistics, Kyiv, Ukraine (Virtual). 2021.
10. Haddad H, Mulki H, Oueslati A. T-hsab: A tunisian hate speech and abusive dataset. In: International Conference on Arabic Language Processing, Springer. 2019; p. 251–63.
11. Boulouard Z, Ouaissa M, Ouaissa M. Machine learning for hate speech detection in arabic social media. In: Computational Intelligence in Recent Communication Networks. Springer, New York. 2022. p. 147–62.
12. Albadi N, Kurdi M, Mishra S. 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). 2018; p. 69–76.
13. Chowdhury AG, Didolkar A, Sawhney R, Shah R. Arhnet-leveraging community interaction for detection of religious hate speech in arabic. In: Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: Student Research Workshop, 2019. p. 273–80.
14. Alsafari S, Sadaoui S, Mouhoub M. Hate and offensive speech detection on arabic social media. Online Soc Netw Media. 2020;19: 100096.
15. Anezi FYA. Arabic hate speech detection using deep recurrent neural networks. Appl Sci. 2022;12(12):6010.
16. Aldjanabi W, Dahou A, Al-qaness MA, Elaziz MA, Helmi AM, Damaševičius R. Arabic offensive and hate speech detection using a cross-corpora multi-task learning model. Informatics. 2021;8:69.

17. Husain F, Uzuner O. Investigating the effect of preprocessing arabic text on offensive language and hate speech detection. Trans Asian Low-Resource Language Inform Process. 2022;21(4):1–20.
18. Alsafari S, Sadaoui S. Semi-supervised self-learning for arabic hate speech detection. In: 2021 IEEE International Conference on Systems, Man, and Cybernetics (SMC), 2021. p. 863–8.
19. Mostafa A, Mohamed O, Ashraf A. Gof at arabic hate speech 2022: breaking the loss function convention for data-imbalanced arabic offensive text detection. In: Proceedinsg of the 5th Workshop on Open-Source Arabic Corpora and Processing Tools with Shared Tasks on Qur'an QA and Fine-Grained Hate Speech Detection, 2022. p. 167–75.
20. Mursi KT, Alahmadi MD, Alsubaei FS, Alghamdi AS. Detecting islamic radicalism arabic tweets using natural language processing. IEEE Access. 2022;10:72526–34.
21. Omar A, Mahmoud TM, Abd-El-Hafeez T, Mahfouz A. Multi-label arabic text classification in online social networks. Inform Syst. 2021;100:101785.
22. AbdelHamid M, Jafar A, Rahal Y. Levantine hate speech detection in twitter. Soc Netw Anal Mining. 2022;12(1):1–13.
23. Bennessir MA, Rhouma M, Haddad H, Fourati C. icompass at arabic hate speech 2022: Detect hate speech using qrnn and transformers. In: Proceedinsg of the 5th Workshop on Open-Source Arabic Corpora and Processing Tools with Shared Tasks on Qur'an QA and Fine-Grained Hate Speech Detection, pp. 176–180; 2022.
24. Dataset: Arabic Levantine Hate Speech. https://dictionary.cambridge.org/us/dictionary/english/hate-speech.
25. Dataset: Hate Speech Detection in Arabic Twittersphere. https://github.com/raghadsh/Arabic-Hate-speech
26. Dataset: Religious Hate Speech Detection for Arabic Tweets. https://github.com/nuhaalbadi/Arabic_hatespeech
27. Dataset: Hate and Offensive Speech Detection on Arabic Social Media. https://github.com/sbalsefri/ArabicHateSpeechDataset.
28. Dataset: AraCOVID19-MFH: Arabic COVID-19 Multi-label Fake News Hate Speech Detection. https://github.com/MohamedHadjAmeur/AraCOVID19MFH.
29. Dataset: Multi-lingual Hate Speech. https://www.kaggle.com/datasets/wajidhassanmoosa/multilingual-hatespeech-dataset?resource=download.
30. Ousidhoum N, Lin Z, Zhang H, Song Y, Yeung D-Y. Multilingual and multi-aspect hate speech analysis. arXiv preprint arXiv:1908.11049. 2019.
31. Alshalan R, Al-Khalifa H. A deep learning approach for automatic hate speech detection in the saudi twittersphere. Appl Sci. 2020;10(23):8614.
32. Stop-Words: List of Arabic Stop Words on Github. https://github.com/nuhaalbadi/Arabic_hatespeech/blob/master/stop_words.csv.
33. El Mahdaouy A, El Alaoui SO, Gaussier E. Word-embedding-based pseudo-relevance feedback for arabic information retrieval. J inform Sci. 2019;45(4):429–42.
34. Kim Y. Convolutional neural networks for sentence classification. CoRR arXiv:abs/1408.5882. 2014.
35. Alkouz B, Al Aghbari Z, Al-Garadi MA, Sarker A. Deepluenza: Deep learning for influenza detection from twitter. Expert Syst Appl. 2022;198:116845.
36. Antoun W, Baly F, Hajj H. Arabert: Transformer-based model for arabic language understanding. arXiv preprint arXiv:2003.00104. 2020.
37. NumPy: The Fundamental Package for Scientific Computing with Python. https://numpy.org/
38. NLTK: Natural Language Toolkit. https://www.nltk.org/.
39. scikit-learn: Tools for Predictive Data Analysis. https://scikit-learn.org/stable/.
40. TensorFlow: Open Source Platform for Machine Learning. https://www.tensorflow.org/overview.
41. Keras: Deep Learning API Written in Python. https://keras.io/api/.
42. AraBERT: Arabic Pretrained Language Model Based on Google's BERT. https://github.com/aub-mind/arabert#AraBERT.