Multimedia Systems (2022) 28:1963–1974
https://doi.org/10.1007/s00530-020-00742-w

SPECIAL ISSUE PAPER

Detection of hate speech in Arabic tweets using deep learning

Areej Al-Hassan[1] · **Hmood Al-Dossari**[1]

Received: 11 October 2020 / Accepted: 20 December 2020 / Published online: 21 January 2021

Abstract
Nowadays, people are communicating through social networks everywhere. However, for whatever reason it is noticeable that verbal misbehaviors, such as hate speech is now propagated through the social networks. One of the most popular social networks is Twitter which has gained widespread in the Arabic region. This research aims to identify and classify Arabic tweets into 5 distinct classes: none, religious, racial, sexism or general hate. A dataset of 11 K tweets was collected and labelled and SVM model was used as a baseline to be compared against 4 deep learning models: LTSM, CNN + LTSM, GRU and CNN + GRU. The results show that all the 4 deep learning models outperform the SVM model in detecting hateful tweets. Although the SVM achieves an overall recall of 74%, the deep learning models have an average recall of 75%. However, adding a layer of CNN to LTSM enhances the overall performance of detection with 72% precision, 75% recall and 73% F1 score.

Keywords Hate speech · Arabic tweets · Arabic NLP · Deep learning · Multiclassification · Social networks · Text mining

1 Introduction

Social media platforms are getting more popular in the Arab region. According to The Arab Social Media Report [1], "the penetration of social media in some countries of the Arab region reached 90% of the population". It shows also that 58% of the Arabs are expressing both of their positive and negative thoughts through social networks. Twitter has shown a rapid growth in the recent years. Arab users generate 27.4 million tweets per day. From that big number, we can assume that hate speech can spread easily and quickly through these platforms. Blaya [2] argued that there is a consensus that cyber-hate is an international issue that needs to be dealt with. The noteworthiness of moderating the cyber hate is because of the relation between hate speech and actual hate crimes and violence. Social networks provide a chance for radical groups to aggregate people with similar thoughts to create a solidarity for some ideology to follow together [3]. Hate speech is a controversial issue that cannot be prevented unilaterally due to the massive scale of social networks. In this research, we are taking the role in countering the Arabic hate speech in twitter in response to national and international needs. This will be achieved by harnessing the power of supervised deep learning techniques to automate the identification of Arabic hate speech in Twitter.

✉ Areej Al-Hassan
aralhassan@ksu.edu.sa

Hmood Al-Dossari
hzaldossari@ksu.edu.sa

[1] Information Systems Department, College of Computer Science and Information Systems, King Saud University, Riyadh, Saudi Arabia

2 Background

2.1 Hate speech

It is not easy to comprehend hate speech. However, each culture has different characteristics that can be distinguished and recognized. These characteristics are debatable. Gelashvili and Nowak [3] say that it is difficult to regulate hate speech since many questions will be raised, such as: which kind of hate need to be dealt with?

For studying hate speech, some common terminologies have been agreed on by a number of researchers, for example, some researchers [4] have surveyed general rules for hate speech recognition. In brief, it can be recognized when stereotyping group of people together or individuals by using racial and sexist slurs with intent to harm. In addition, indecently speaking about religion or specific country.

For social media, Waseem and Hovy [5] contributed with a number of characteristics to identify hate speech specifically in Twitter platform which include: "using sexiest and racial terms, attacking and criticizing minority, promoting violence, distorting the truth with lies and supporting suspicious hashtags" [5]. Also, Anis [6] discussed the hate speech in the Arabic newspapers and analyzed the common theme of it and concluded that the religion and sectarian themes are the most common hate-speech theme in the Arab region. The given parameters will make it easy to identify and recognize hate speech in any text.

2.2 Arabic text in twitter

The Arabic tweets are known by their complexity, where it is difficult to understand the intent of the user [1]. Add to that, Arabic tweets are very noisy and usually they are not correctly spelled. Also, some tricky variances can be found such as: writing from right to left and the usage or the neglection of diacritics [7]. As well as the different local informal dialect used. Because of this controversy, Researchers usually work with specific Arab region by choosing data and algorithms that fit this specific region. Tweets text is generally considered as unstructured text that includes the regular natural language used in daily life, it is hard to extract insights from such text since they are context-dependent sentences. However, text mining methods are capable of interpreting the variability of unstructured data [8].

2.3 Text representation methods

When performing NLP task, such as hate-speech detection, a prior step needs to be taken by transforming the unstructured text into a form that enables the classification algorithms to perform the designated task. Using lexicons can be a way, they can be used either to learn from the existing corpuses or to learn manually from domain independent lexicon of terms labelled with their correspondent weight or frequency. This type of feature usually used in unsupervised machine learning settings [9].

Another approach is called Bag-of-Words, this approach can be implemented by tokenizing each tweet and make a list of words and then a vectorization process is performed for each word in the tweet by giving it a weight based on the word frequency in the tweet and in between different tweets. Vectorization can be performed using several models, one of them is called TF-IDF scheme. After words weighting, a vector of weights is created and it contains a list of words or Bag Of Words (BOW) [10].

In addition, a generative probabilistic model called the Latent Dirichlet Allocation [11], which can catch the semantic relations between words in the same topic. Also, in this model the topics will be estimated for the entire corpus without taking account to which class these topics belong, for that reason, LDA works well with unlabelled datasets.

Finally, Word2Vec [12], is one of the word-embedding methods that uses neural networks. It is considered as a predictive model that can predict the context of given word. In addition, AraVec, is an Arabic project [13] which is a pre-trained word embedding for Arabic words representation that has a total of 1,169,075,128 tokens of Arabic words.

2.4 Machine learning approaches

The mentioned features are the foundation to prepare the text to make it eligible for further processing via machine learning models. Machine learning algorithms can be categorized into: supervised, semi-supervised and unsupervised algorithms.

Supervised approach which relies on a manual labelling of the data. This approach is useful for domain-dependent events but it is effort and time consuming. Examples of this approach can be: Support Vector Machine (SVM), k Nearest Neighbours (k-NN) and Maximum Entropy. All of them were used by Bouazizi and Otsuki [14] to detect sarcasm in twitter. They concluded that their argument raises a limitation on the supervised approaches that deals with unstructured text in terms of the need to incorporate a large volume of training data to improve the accuracy.

Semi-supervised approaches deal with both of labelled and unlabelled data that can be an alternative to the time-consuming supervised approaches. Xiang et al. [15] have taken an advantage of linguistic regularities in offensive language. They replaced the data labelling with an automatically generated feature (topical and lexicon features).

Finally, unsupervised approaches which dynamically extract domain-related key terms instead of the costly data labelling. Gitari et al. [16] proposed a classifier that takes an advantage of sentiment analysis techniques to detect hate speech. They enhanced the model by applying semantic features related to hate speech.

2.5 Deep learning approaches

Deep learning is the part of machine learning which depends entirely on deep artificial neural networks that learn and identify patterns by mimicking the event in layers of neurons. According to Goodfellow et al. [17] there are two perspectives in deep learning. First, the right representation of data. Secondly, the depth of the neural networks is an important factor since the greater depth will give more effective power, because the complicated task will be broken into a series of layers.

3 Related work

Starting with the World Wide Web, Warner and Hirschberg [18], are the first to investigate how to identify hate speech in the world wide web. Their work is targeted to specific type of hate which is anti-Semitic. For Twitter, Watanabe et al. [19] proposed a supervised approach for hate-speech detection Their approach proved that supervised classifier performs better in the binary classification when compared with ternary classification. Another binary classifier is developed by Burnap and Williams [20] that detects hateful and non-hateful tweets from labelled dataset.

Many recent researchers have tended to use deep learning for the hate-speech detection task. For instance, Gambäck and Sikdar [21] have experimented Waseem and Hovy's dataset [5]. They have trained four convolutional neural networks which resulted in a high precision. In addition, Badjatiya et al. [22] also used Waseem and Hovy's corpus but for different deep learning scenarios. They compared different combinations of deep learning models and state-of-art classifiers. They concluded that combining deep neural network models with GBDT classifier will result in the best accuracy. Also, Zhang et al. [23] have conducted a comparative evaluation and examined combining both of convolutional neural networks and gated recurrent networks. Their work was performed on several public datasets.

For the Arabic language, there is probably no work that is directly related to the detection and classification of general hate speech. However, other antisocial behaviors were investigated by some Arab researchers, such as Abozinadah et al. [24] who investigated the abusive language detection in twitter using a statistical learning approach. Also cyberbullying detection has received attention from researchers, such as Haidar et al. [25] who investigated the detection of cyberbullying in Arabic tweets.

For Arabic hate-speech detection, one contribution is found which is specifically performs a binary classification of religious hate speech. For instance, Albadi et al.[26] have worked with the classification of religious hatred in Arabic tweets. They used both of supervised and unsupervised approaches.

4 Proposed solution

This research aims to develop a model for detecting Arabic hate speech in Twitter platform, then classifying the Arabic tweets based on the type of hate used in each tweet.

4.1 Hate classes

We choose to assign five distinct classes of hate (Religious, Racism, Sexism, General hate speech, Not hate speech) [27]. Prior work in Arabic hate-speech detection has targeted mainly binary classification of hate.

In Table1, we summarized what constitute each one of the particular classes of hate by mixing the previous hate-speech properties with local Arab culture.

4.2 Model architecture

A high-level view of the system is drawn (Fig. 1) to visualize and summarize all the phases related to our Arabic hate-speech detection model.

Table 1 Interpretation of the 5 classes of hate

Class	Interpretation
Religious	Any Religious discrimination, such as: Islamic sects "Sunni, Sheie, Alrafidhah, …etc." Also, anti-Judaism or anti-Hinduisand, anti-Christian and their respective denominations, calling for atheism or other religions. Also attaching relations of following or not following a particular religious group, these groups include but not limited to: ISIS and Al-qaedah, Muslim Brotherhood. Al-Houthi and many others
Racism	Any Racial offense or tribalism, regionalism, prejudice against particular tribe or region, xenophobia (especially for migrant workers) and nativism (hostility against immigrants and refugees). Also, offending the appearance and color of individual or offending particular country leader or country politics
Sexism	Any post that offense particular gender using any form of hostility or devaluation based on person's gender. In addition, any form of misogyny tendency
General hate speech	Any general type of hate which is not mentioned in the previous classes. Whether it contains: general hatred, obscene, offensive and abusive words that are not related to religion, race or sex
No hate	If the tweet does not contain any form of hatred

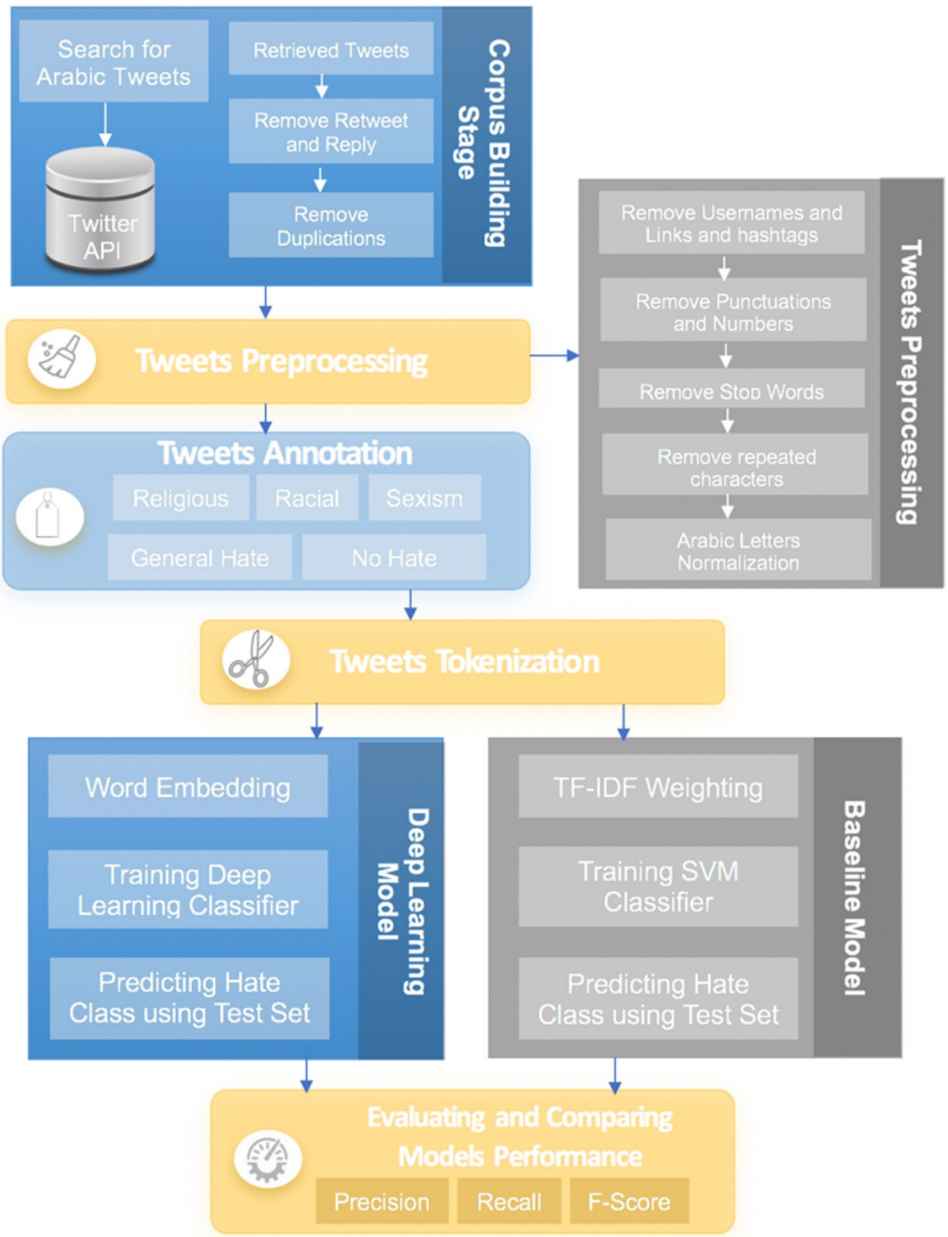

Fig. 1 Model architecture for Arabic hate-speech detection

5 Arabic hate-speech corpus

Twitter platform is used as our data source for the experiment. Since we are employing deep learning approach, this model usually requires relatively big corpus in order to train the model and to get some reasonable and realistic results. In this section, we will go through how we build and prepare the corpus.

5.1 Twitter API

Twitter API platform enables any researcher to experiment and build solutions using public tweets. It contains endpoints that serves many purposes, such as Twitter Streaming API which is used to fitch data happening in a real-time or near real-time, where Twitter Search API is used for older tweets. To use this platform, researcher needs to sign-up for a developer account in order to set up OAuth tokens for the access. OAuth is a standard protocol for authorization used by Twitter.

5.2 Tweets collection

For collecting the tweets, Tweepy Python library was used for the authentication using our Twitter credentials and to search for tweets using a cursor. A list of hashtags that attract and trigger the hateful content has been created.

Table 2 Examples of the hashtags

Targeted class	Hashtag	Hashtag in English
Religious	#الحوثي، #شيعي	#Houthi, #Sheie (Islamic groups).
Racism	#خضيري	#Khadiri (Person with unrecognized tribe).
Sexism	#النسويات	#Feminists.
General hate speech	A number of known hated Arab names including politicians, Social media influencers, TV Actors	

Table 3 Tweets pre-processing output

Tweet	Clean Tweet
@AsinHadi كلاااام جميييل جدا اتفق معك الحكومة خارج البلاد والعدو الحوثي داخل البلاد يعبث بهاااااا ولو كانت الشرعية هي الأقوى… https://t.co/ukiDJrADsO	كلام جميل جدا اتفق معك الحكومه خارج البلاد والعدو الحوثي داخل البلاد يعبث بها ولو كانت الشرعيه هي الاقوي

Table 2 contains a sample of the hashtags that have been used in the search query parameter.

Balancing the number of hate tweets with non-hate tweet is not realistic and does not reflect the real situation in social media. Hence, we just identified the list of hashtags that surely contains both of hateful and non-hateful content to preserve a natural and realistic scenario.

5.3 Tweets pre-processing

In order to put a little bit of structure and standard for the tweets, a number of pre-processing tasks has performed:

1. Removing the punctuations—for that, we declared a list for the Arabic punctuations and used a ready-made list of English punctuations found in NLTK Python library.
2. Normalizing Arabic Text: this process is mainly to produce more consistent tweets. In Arabic language, we have different variations for representing some letters which are:

    a. Letter (Alef) (أ) which has the forms (أ-إ-آ-ا) we normalized all these four letters into one letter which is (ا).
    b. Letter (Alef Maqsora) (ى) which can be mistaken and written as (ي). It will be normalized to (ي).
    c. Letter (Taa Marbouta) (ة) has been normalized to (ه).

    Finally, we included the removal of Arabic dash that is used to expand the word (e.g. اهـــلا) to (اهلا); which means "hi". As well as removing new lines.

3. Removing repeated characters: such as (اهليييييييين) which means "Hiiiii", to be (اهلين), which means "Hi".
4. Processing the general structure of the tweet. For instance, the removal of (@username), URLs, hashtags.

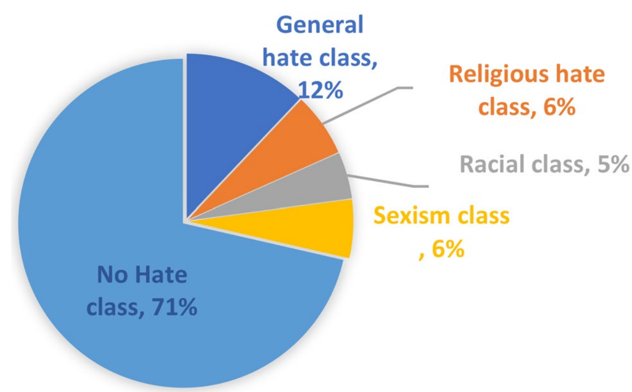

Fig. 2 The percentage of tweets in each class

Table 3 contains a sample of the output from the text pre-processing stage.

5.4 Tweets labelling

We performed a manual annotation for our data set. We had a help from 2 volunteers to review the labelled data in order to get rid of any annotator bias. The volunteers were provided with a guide to follow to distinguish the hate classes. However, labelling and distinguishing the right class of hate is not trivial due to the differences in context and what is intended by the user. The output of this stage is a new column containing the classes of tweets each class was abbreviated to one character for simplicity.

5.5 Hate-speech corpus

In this section, we will present some statistics (Fig. 2) for our hate-speech corpus.

We appended all the tweets resulted from different hashtags together in one corpus and shuffled the resulted corpus. We ended up having a corpus of 11 k labelled tweets

out of 37 k retrieved tweets. In Fig. 2, we can notice that the major subset of tweets goes for the non-hate class, were the minor subset goes for the racial class.

6 Baseline model

A baseline gives us a point of comparison for the more advanced models. By using the baseline, we can comment on how much better the performance of our intended deep learning method compared to that baseline algorithm. Since we are working with a newly collected corpus for a relatively new problem (Arabic Hate speech and multiclass classification) which has never been investigated before, we will implement our own baseline.

SVM is a supervised approach and advanced way for classifying high-dimensional data that has lots of features. The basic idea in SVM that it finds high-dimensional support vectors across which to divide the data, these support vectors define hyperplane in multidimensional space to separate different classes. SVM has proven its powerful capability in classifying Arabic text with a satisfying performance. For instance, Alabbas et al. [28] results showed that SVM worked very well in Arabic binary text classification problems.

Our SVM baseline was built using "SciKit-learn" which is a machine learning library in Python. For the text representation we are going to use TF-IDF weighting method It basically a vectorizer that converts the text into numerical form based on the weight of the text. This scheme assigns the weight according to the word occurrence in the document. Hence, when the term occurs more in a document, it can be said that this term is more representative of the content of the document and vice versa. In order to train our SVM classifier, we preserved 70% of the dataset for training and 30% for testing.

7 Deep learning models

Deep learning techniques can learn automatically from the data following a supervised strategy. Labelled training data need to be provided as an input. These models are competent to understand and analyze text using deep artificial neural networks with multiple stacked layers. However, two of the most popular examples of deep artificial neural networks are:

- CNN: Convolutional Neural Networks.
- RNN: Recurrent Neural Networks.

CNN is considered as an effective network for extracting features from the data. On the other hand, RNN is more suitable for modelling orderly sequence tasks [23]. In this research, we are going to experiment the effectiveness of using different deep learning settings using RNN alone and using a combination of RNN and CNN. We assume that combining the two architectures will show a better performance as they will be able capture more hate-speech patterns. However, RNN is a family of different architectures with different gating mechanisms, which includes the following:

- LTSM (Long Short-Term Memory network)—which is capable of learning long-term dependencies between words by remembering words for long period of time.
- GRU (Gated Recurrent Unit)—which is a variant of LTSM but GRU is simpler and faster in the training process, where the LTSM is more powerful and complex than GRU.

We can have different settings of our deep neural networks architecture by adjusting the layers of neural networks and fine tuning the parameters until they satisfy our Arabic hate-speech problem. Three main settings of deep neural networks will be experimented:

- LTSM model.
- Ensemble model of LTSM and layer of CNN.
- GRU model.
- Ensemble model of GRU and a layer of CNN.

In order to build and experiment these 4 models, we are going to use Keras, which is a deep learning library in Python that works on top of TensorFlow.

7.1 Word embeddings

In order to represent the text in our corpus, we are going to use word embedding representation which is more improved technique over the traditional vectorization methods. We will utilize Keras embedding layer to do the job. Keras Embedding requires the input corpus to be tokenized and encoded to integers. To do that, we use the tokenizer provided by Keras library. We also need to transform our variable length list of tokenized words into a sequence of the same length as expected by the any deep learning model. Finally, we need to transform each character label in our label column into an integer value to make it compatible with the embedding layer.

Now our corpus is ready to be fed to the embedding layer of our model. So, we define our deep learning model as sequential that works sequentially (layer by layer).

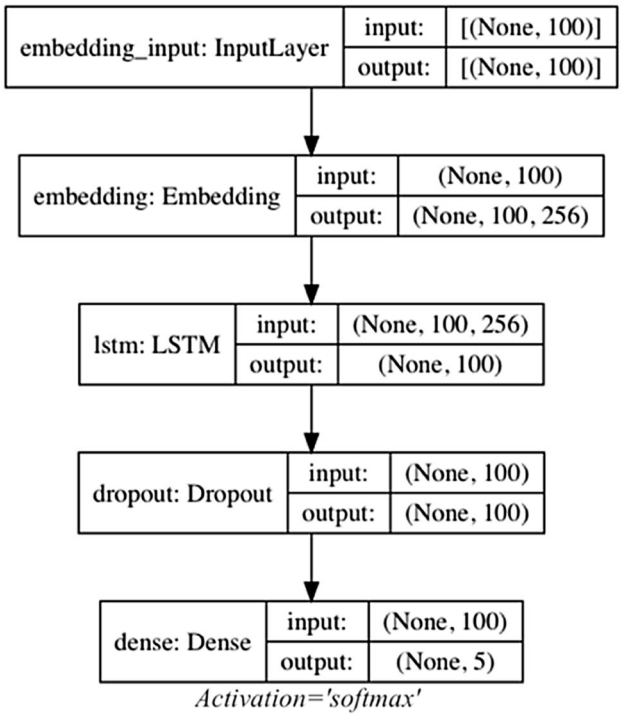

Fig. 3 LTSM Model Architecture

7.2 Deep learning models' architecture

In this section, we are going to draw the architecture of our 4 deep learning models. We have fine-tuned the parameters of each model to reach their optimum performance.

1. LTSM Model Architecture:

In this model, we set one layer for the LTSM itself (Fig. 3). Moreover, since that RNNs are prone to overfitting, we add a dropout of 0.2 to avoid overfitting. Finally, we need to add a dense layer to narrow the number of neurons down to 5 neurons with SoftMax activation function to fit the number of classes of our hate-speech detection problem.

1. Ensemble Model of LTSM + CNN:

This is our second model, where we add a convolutional layer to speed up the training time of the neural network (Fig. 4). CNN provides a filter to give a higher-level representation of data. For the activation of the CNN layer, Relu activation function is used. Then, the output will be pooled to smaller dimension then it will be fed to the LTSM. Finally, the same a dense layer is added to narrow the number of neurons down to 5 neurons with Softmax activation function.

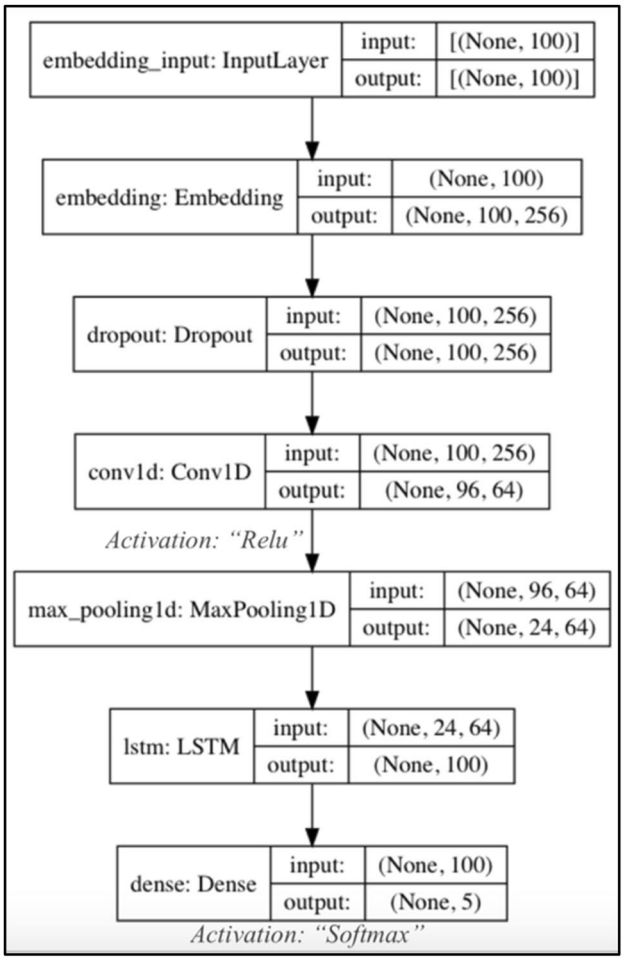

Fig. 4 Ensemble LTSM and CNN Model Architecture

1. GRU Model:

By referring to the comparative study by Yin et al. (Yin, Kann, & Yu, 2017), that systematically compares the performance of LTSM, CNN and GRU. Their results showed that GRU outperformed the others in text analysis tasks. For that reason, we decided to build a third model using GRU for our experiment and compare its results with the previous models. For our GRU that is shown in Fig. 5, the activation function "Tanh" has been used, after that a layer of dropout and finally, SoftMax layer for the classification.

1. Ensemble Model of GRU + CNN:

Our CNN-GRU model showed in Fig. 6 consists of one initial CNN layer with Relu activation function. The output of it will be pooled to smaller dimension then it will

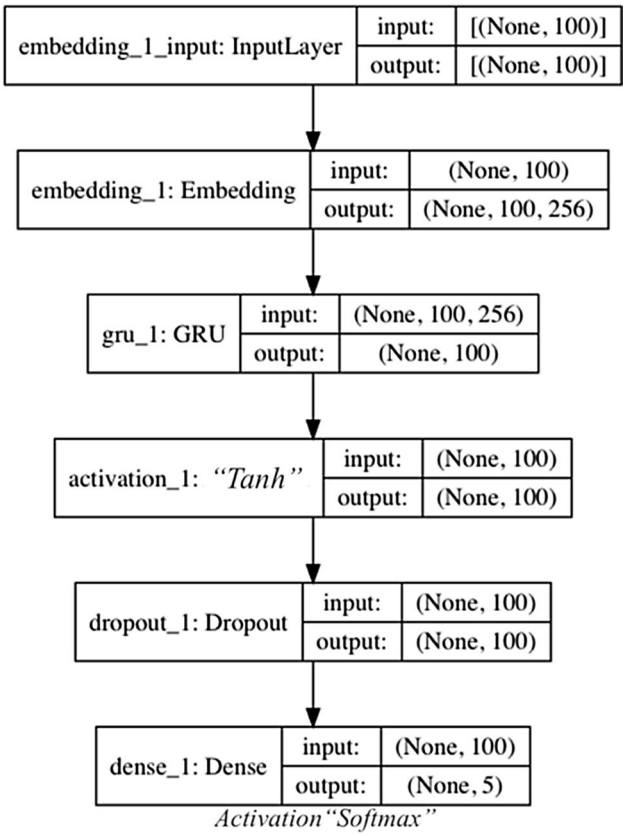

Fig. 5 GRU Model Architecture

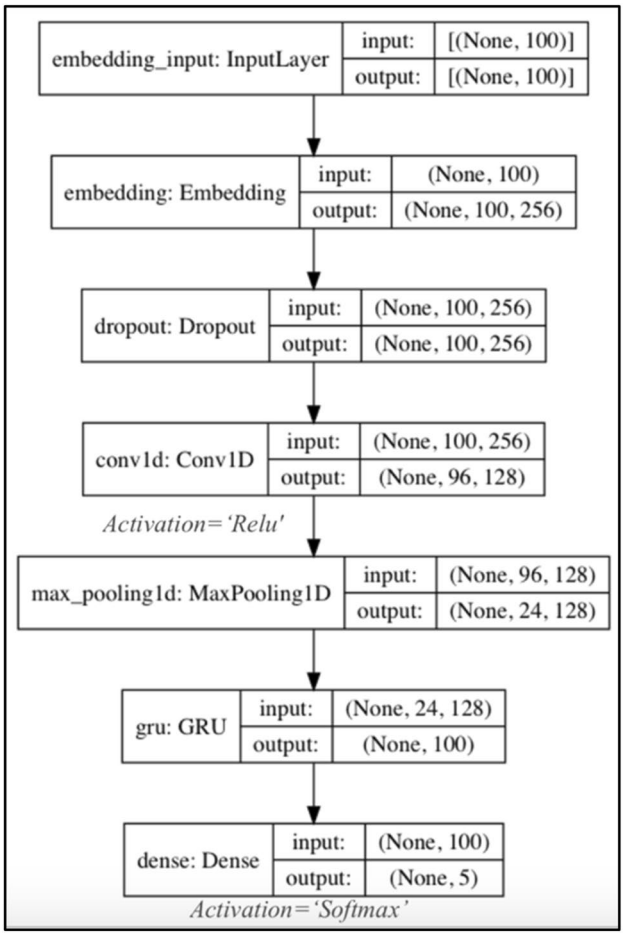

Fig. 6 CNN and GRU Model Architecture

be fed to the GRU which will be fed to the dropout layer and finally, SoftMax layer for the classification. This will allow a faster computational time and hopefully a better performance.

8 Expermintal setup

For all of the four models, we used the same training and compilation settings. we used the same train_test_split() from SciKit-learn. Before training the models, we need to configure the training process, for that we can use the compile method in Keras, this method is used to specify the required parameters to reach better learning process. The following parameters need to be specified in this method:

- Loss Function: from the available loss functions in Keras, we choose "sparse_categorical_crossentropy" because our target is a multiclass categorical format (5 classes). And since our labels are converted to integers, this loss function is compatible with our labels.
- Optimizer: there are many algorithms that can be used to optimize the learning process, we choose to use the popular Adam algorithm is an adaptive learning rate optimization algorithm.

Now we can fit the models that we have built to our training set, we experimented 5, 10 and 15 epochs and we were satisfied with the results of 10 epochs since that specifying too many epochs may lead to overfitting.

9 Evaluation and results

In this section, we are going to evaluate the 4 models and present the results achieved by them.

9.1 Evaluation metrics

In order to measure the performance of our deep learning models, we are going to use different types of evaluation metrics that can fairly judge the models.

1. Classification Accuracy:

It is the ratio of number of correctly classified tweets (regardless the class) to the total number of tweets in the test set. The following equation is used to calculate accuracy:

$$\text{Accuracy} = \frac{\text{Number of correctly classified tweets}}{\text{Total number of classified tweets}} \quad (1)$$

Accuracy works well for balanced datasets, but since that hate-speech detection task is an imbalanced classification problem (due to that the majority of real-life tweets are not hate speech), we will not pay big attention to that metric.

2. Recall:

We should pay more attention to that metric, since it focuses on the positive cases. For each class label, recall means the ratio of the correctly classified tweets for that particular class, to the number of tweets that were belonging to that class but they were incorrectly classified by the model. For instance, the recall equation:

$$\text{Recall} = \frac{\text{Racial tweets correctly identified}}{\text{Total number of Racial tweets}} \quad (2)$$

3. Precision:

This measure is also relevant to our case, it represents the fraction of retrieved tweets from particular class that are relevant. For instance, when we take the (Sexism).

$$\text{Precision} = \frac{\text{Sexism tweets correctly classified}}{\text{Total Tweets classified as "Sexism"}} \quad (3)$$

4. F1-Score:

Which is the weighted average of precision and recall, and it is more useful than accuracy for our case which is uneven class distribution.

$$\text{F1 - Score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \quad (4)$$

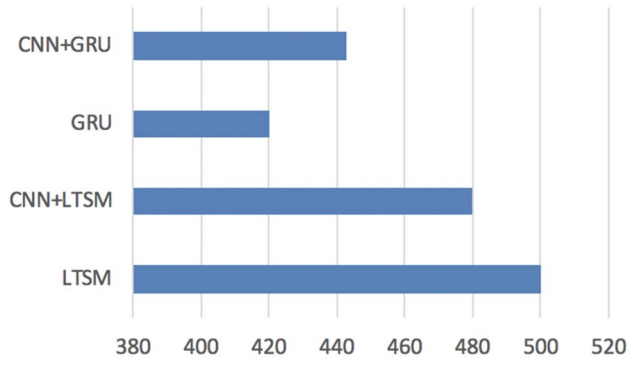

Fig. 7 Models training time in seconds

9.2 Model evaluation

For our baseline, the evaluation method used is the tenfold cross-validation which is performed using the test set. This method will split the test data into 10 randomly assigned segments and will reserve one segment as the test data, then it will train on each of the remaining 9 segments and measure their performance against the test set. For that, we used the cross-validation function in SciKit-Learn.

For the deep learning models, we used to "Evaluate" function in Keras, which will return the loss value and metrics values for the model using the test set.

The training time in seconds for each model is presented in Fig. 7.

9.3 Experimental results

SVM resulted in average accuracy of 75% of the classifier. But as we mentioned before, since we are working with imbalanced dataset problem, we will rely on Recall and precision (shown in Table 4) for evaluating the models.

From the results, we can see that SVM is able to distinguish non-hate-speech tweets, this can be seen from the high recall resulted. On the other hand, SVM is not able to distinguish the hate-speech classes (Very poor recall in the other classes), which means that in our case, SVM works poorly for multiclassification and imbalanced datasets, but it works fine in case of binary classification.

We can justify the high precision of hate-speech classes by referring to its equation, the denominator of the precision equation describes the total number of tweets classified to a specific class by the model itself, so precision quantifies only the correctly classified tweets with respect to the total classified tweets by the model. Hence, we can imagine that the model was able to detect a very low number of hate-speech tweets.

Now, we are going to interpret the results of each deep learning model:

Table 4 SVM classification results

Class	Precision	Recall	F1-score
Non-hate speech	0.73	1	0.85
General hate speech	0.71	0.07	0.13
Religious hate speech	0.85	0.19	0.31
Racial hate speech	0.91	0.04	0.07
Sexism	0.75	0.05	0.1
Average	0.75	0.74	0.65

Table 5 LTSM classification results

Class	Precision	Recall	F1-score
Non-hate speech	0.81	0.9	0.86
General hate speech	0.34	0.3	0.32
Religious hate speech	0.61	0.45	0.52
Racial hate speech	0.34	0.16	0.22
Sexism	0.51	0.29	0.37
Average	0.71	0.74	0.72

Table 6 CNN + LTSM classification results

Class	Precision	Recall	F1-score
Non-hate speech	0.82	0.9	0.86
General hate speech	0.36	0.28	0.31
Religious hate speech	0.64	0.55	0.59
Racial hate speech	0.33	0.21	0.25
Sexism	0.54	0.36	0.43
Average	0.72	0.75	0.73

Table 7 GRU classification results

Class	Precision	Recall	F1-score
Non-hate speech	0.79	0.93	0.86
General hate speech	0.34	0.19	0.24
Religious hate speech	0.66	0.47	0.55
Racial hate speech	0.22	0.12	0.15
Sexism	0.47	0.26	0.33
Average	0.69	0.74	0.70

1. LTSM Model Result:

The result of our first deep learning model shows more consistency in precision, recall and f1-score. Also, the recall here is more satisfying, which means that LTSM was powerful enough to identify a tweet that contains Arabic hate speech. Again, religious tweets were obviously easier to identify by the model, this can be seen from the good recall (45%) (Table 5).

1. Ensemble Model (CNN + LTSM) Result:

Overall, this ensemble model of CNN and LTSM offers better results, there is a good enhancement in the recall of all the hate-speech classes, which is approximately 9% additional enhancement in the recall (Table 6).

Table 8 CNN + GRU classification results

Class	Precision	Recall	F1-score
Non-hate speech	0.8	0.94	0.86
General hate speech	0.37	0.19	0.25
Religious hate speech	0.68	0.45	0.54
Racial hate speech	0.36	0.15	0.21
Sexism	0.48	0.35	0.41
Average	0.71	0.75	0.72

1. GRU Model Result:

GRU resulted in a slightly better recall in the (non-hate class) when compared with LTSM. However, LTSM provided better recall in hate-speech classes. We can say that the overall performance of GRU is similar and comparable with LTSM (Table 7).

1. Ensemble Model (CNN + GRU) Result:

This model shows a bit of change in the performance. There is a noticeable increase in the recall of the sexism class (10% increase) when compared with the standalone GRU model. This slight increase in recall raised the average recall for this model (75%) which is the best overall recall among all of our models (Table 8).

9.4 Results discussion

For the resulted training time of deep learning models in Fig. 7, we can notice that LTSM model is the slowest in terms of training time and GRU is the fastest, this is very reasonable because GRU structure is simpler than LTSM since it has only 2 gates, but on the other hand, LTSM is known that it is computationally expensive and has sophisticated structure.

Based on the presented models results, it is obviously clear that deep learning approaches outperforms the SVM approach in the Arabic hate-speech multiclassification task. The results also show that SVM has a better ability in the binary classification since it was able to distinguish non-hate classes, but it was not able to specifically identify hate-speech classes.

From Table 9, we marked the highest recall results in bold, it can be seen that our proposed deep learning models have relatively similar recall. However, we can notice

Table 9 Overall recall for each class in each model

Model\Class	Non-hate	General hate	Religious hate	Racial hate	Sexism
LTSM	0.9	**0.3**	0.45	0.16	0.29
CNN + LTSM	0.9	0.28	**0.55**	**0.21**	**0.36**
GRU	0.93	0.19	0.47	0.12	0.26
CNN + GRU	**0.94**	0.19	0.45	0.15	0.35

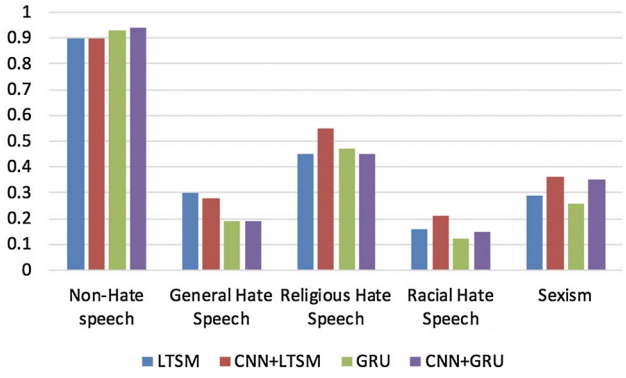

Fig. 8 Overall recall of the 4 models

that the 2 LTSM models outperform the 2 GRU models in the recall of the four hate classes, this can be explained due to the ability of LTSM in handling multiclass classification over GRU. On the other hand, GRU outperformed in the non-hate-speech tweets identification. This proves the robustness of LTSM in capturing and identifying multiple classes. In fact, we can assume that GRU will give better results in the case of binary classification.

In addition, we can notice that the performance of the LTSM can get better when adding a layer of CNN. We can conclude that both GRU and LTSM works good for classifying Arabic hate speech into 5 classes. Also, adding a layer of CNN to LTSM enhances the overall performance of the model, resulting in better recall for the four hate-speech classes (Fig. 8).

10 Conclusion and future work

In this research, we supported the national need developing a model that automatically detect Arabic hate speech in twitter. After acquiring a sufficient knowledge in what constitutes Arabic hate speech, we used this knowledge to label a dataset of 11 K tweets. Then, we built our SVM baseline using TF-IDF words representation and proposed four deep learning architectures that are capable of identifying and classifying Arabic hate speech in twitter into 5 classes. We compared the proposed models with the SVM baseline. Comparison results show that our deep learning approaches outperformed the baseline in the multiclassification of hate classes. However, the ensemble model of CNN + LTSM produced the best results.

As future work we will consider expanding our data set and intensifying the training of our neural networks by including data from another platform "Facebook" as it is the most used platform in the Arab region. Also, we aim to handle and classify hate in real-time stream of tweets. In addition, for the representation of our text, in the future we have a chance to try different word representation methods such as utilizing the AraVec project. Finally, since that our standard memory and CPU did not allow us to investigate and implement more deep learning layers, another direction of future work can be used in better hardware and experimental settings that are more suitable for deep learning models, this can be done by utilizing powerful GPU.

Acknowledgements The authors would like to thank Deanship of scientific research in King Saud University, for funding and supporting this research through the initiative of DSR Graduate Students Research Support (GSR).

References

1. Salem, F.: Arab Social Media Report : Social Media and the Internet of Things: Towards Data-Driven Policymaking in the Arab World - Potential, Limits and Concerns, MBR School of Goverment **7**, (2017). https://www.mbrsg.ae/home/publications/research-report-research-paper-white-paper/arab-social-media-report-2017.aspx
2. Blaya, C.: Cyberhate: A review and content analysis of intervention strategies. Aggress. Violent Behav. **45**, 0–1 (2018)
3. Gelashvili, T., Nowak, K.A.: Hate Speech on Social Media. Lund University (2018)
4. Fortuna, P., Nunes, S.: A survey on automatic detection of hate speech in text. ACM Comput. Surv. **51**(4), 1–30 (2018)
5. Waseem Z., Hovy, D.: Hateful symbols or hateful people? Predictive features for hate speech detection on twitter. In: Proc. NAACL Student Res. Work., pp. 88–93 (2016). https://www.aclweb.org/anthology/N16-2013/
6. Anis M.Y., Maret, U.S.: Hatespeech in Arabic Language. In: International Conference on Media Studies, September 2017
7. Alshutayri A., Atwell, E.: Creating an Arabic Dialect Text Corpus by Exploring Twitter, Facebook, and Online Newspapers, May 2018
8. Irfan, R., et al.: A survey on text mining in social networks. Knowl. Eng. Rev. **30**(2), 157–170 (2015)

9. Assiri, A., Emam, A., Al-Dossari, H.: Towards enhancement of a lexicon-based approach for Saudi dialect sentiment analysis. J. Inf. Sci. **44**(2), 184–202 (2018)
10. Soumya George, K., Joseph, S.: Text classification by augmenting bag of words (BOW) representation with co-occurrence feature. IOSR J. Comput. Eng. **16**(1), 34–38 (2014)
11. Blei, D.M., Ng, A.Y., Jordan, M.I.: Latent Dirichlet Allocation. J. Mach. Learn. Res. **3**, 993–1022 (2003)
12. Mikolov, T., Chen, K., Corrado, G., Dean, J.: Efficient estimation of word representations in vector space. arXiv preprint arXiv :1301.3781 (2013)
13. Soliman, A.B., Eissa, K., El-Beltagy, S.R.: AraVec: a set of Arabic word embedding models for use in Arabic NLP. Procedia Comput. Sci. **117**, 256–265 (2017)
14. Bouazizi, M., Otsuki, T.: A pattern-based approach for sarcasm detection on twitter. IEEE Access **4**, 5477–5488 (2016)
15. Xiang, G., Fan, B., Wang, L., Hong, J., Rose, C.: Detecting offensive tweets via topical feature discovery over a large scale twitter corpus. In: Proc. 21st ACM Int. Conf. Inf. Knowl. Manag.—CIKM'12, pp 1980 (2012)
16. Gitari, N.D., Zuping, Z., Damien, H., Long, J.: A lexicon-based approach for hate speech detection. Int. J. Multimed. Ubiquitous Eng. **10**(4), 215–230 (2015)
17. Goodfellow, I., Bengio, Y., Courville, A.: Deep Learning. MIT Press, Cambridge (2016)
18. Warner W., Hirschberg, J.: Detecting Hate Speech on the World Wide Web. In: Proceedings of the Second Workshop on Language in Social Media, pp. 19–26 (2012)
19. Watanabe, H., Bouazizi, M., Ohtsuki, T.: Hate speech on twitter: a pragmatic approach to collect hateful and offensive expressions and perform hate speech detection. IEEE Access **6**, 13825–13835 (2018)
20. Burnap, P., Williams, M.L.: Us and them: identifying cyber hate on Twitter across multiple protected characteristics. EPJ Data Sci (2016). https://doi.org/10.1140/epjds/s13688-016-0072-6
21. Gambäck, B., Sikdar, U.K.: Using convolutional neural networks to classify hate-speech. Assoc. Comput. Linguist. **7491**, 85–90 (2017)
22. Badjatiya P., Gupta S., Gupta, M., Varma, V.: Deep learning for hate speech detection in tweets. In: Proceedings of the 26th International Conference on World Wide Web Companion, pp. 759–760 (2017)
23. Zhang, Z., Robinson, D., Tepper, J.: Detecting hate speech on twitter using a convolution-GRU based deep neural network. In: ESWC 2018: The Semantic Web, pp. 745–760 (2018)
24. Abozinadah E.A., Jones J.H.: A statistical learning approach to detect abusive twitter accounts. In: Proc. Int. Conf. Comput. Data Anal.—ICCDA '17, pp. 6–13 (2017)
25. Haidar, B., Chamoun, M., Serhrouchni, A.: A multilingual system for cyberbullying detection: arabic content detection using machine learning. Adv. Sci. Technol. Eng. Syst. J. **2**(6), 275–284 (2017)
26. 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 Int. Conf. Adv. Soc. Networks Anal. Min., pp. 69–76 (2018)
27. Al-Hassan, A., Al-Dossari, H.: Detection of hate speech in social networks: a survey on multilingual corpus. Comput. Sci. Inf. Technol. (CS IT) **9**(2), 83 (2019)
28. Alabbas W., Haider, M., Mansour, A., Epiphaniou, G., Frommholz, I.: Classification of Colloquial Arabic Tweets in real-time to detect high-risk floods. In: 2017 International Conference On Social Media, Wearable And Web Analytics (Social Media), pp. 1–8 (2017)