Hate Speech Detection in Saudi Twittersphere: A Deep Learning Approach

Raghad Alshalan
King Saud University & Imam
Abdulrahman bin Faisal University,
Saudi Arabia
Rsalshaalan@iau.edu.sa

Hend Al-Khalifa
King Saud University, College of
Computer and Information Sciences,
Saudi Arabia
hendk@ksu.edu.sa

Abstract

With the rise of hate speech phenomena in Twittersphere, significant research efforts have been undertaken to provide automatic solutions for detecting hate speech, varying from simple machine learning models to more complex deep neural network models. Despite that, research works investigating hate speech problem in Arabic are still limited. This paper, therefore, aims to investigate several neural network models based on Convolutional Neural Network (CNN) and Recurrent Neural Networks (RNN) to detect hate speech in Arabic tweets. It also evaluates the recent language representation model BERT on the task of Arabic hate speech detection. To conduct our experiments, we firstly built a new hate speech dataset that contains 9,316 annotated tweets. Then, we conducted a set of experiments on two datasets to evaluate four models: CNN, GRU, CNN+GRU and BERT. Our experimental results on our dataset and an out-domain dataset show that CNN model gives the best performance with an F1-score of 0.79 and AUROC of 0.89.

1 Introduction

In the Arab region, Twitter is considered as one of the most popular platforms used by Arabic speaking users and it has indeed revolutionized the way people communicate and share opinions, ideas and information. However, due to the dynamic, democratic and unrestricted nature of Twitter platform, it has been increasingly exploited for the dissemination of aggressive and hateful content. While there is no formal definition of hate speech, there is a general agreement among scholars and service providers to define it as any language that attack a person or a group based on some characteristic such as race, color, ethnicity, gender, religion, or other characteristic (Schmidt & Wiegand, 2017).

The pressing need for effective automatic solutions for hate speech detection has attracted significant research efforts recently. Several studies relied on traditional machine learning approaches and leveraged surface features such as bag of words, word and character n-grams which have been shown to be very effective for hate speech detection (Davidson et al., 2017; Fortuna & Nunes, 2018; Jaki & De Smedt, 2018; Waseem & Hovy, 2016). Recently, there has been a clear trend towards the adoption of deep learning methods, which indeed showed better performance over classic methods in hate speech detection task (Badjatiya et al., 2017; Gambäck & Sikdar, 2017; Park & Fung, 2017; Pitsilis et al., 2018; Zhang et al., 2018). However, there is a very limited work that employed deep learning approaches to address the problem in Arabic language (Al-Hassan & Al-Dossari, 2019). To the best of our knowledge, Albadi et al. (2018) work is the only Arabic study published in the area, and it examined GRU architecture, focusing mainly on religious hate speech.

While different deep learning models have been investigated in this area, we are more interested to explore these methods for Arabic language. We believe that the special nature of Arabic languages and its richness and complexity at morphological, lexical and orthographical levels (Darwish et al., 2012) pose some unique challenges that could complicate the task of detecting hate speech. Moreover, the high variety of dialectal Arabic used by Arabic users on social media makes the problem even more complex. In fact, dialects of Arabic are manifold, varying not only from country to country but also within the same country, resulting in many similarly spelled words to have different meanings across different dialects and regions.

Recently, Devlin et al. (2019) introduced BERT (Bidirectional Encoder Representations from Transformers.), a new language representation model that has been successfully applied to numerous NLP tasks, achieving the state-of-the-art results for eleven NLP tasks including sentiment analysis, question-answering and textual entailment. However, works examining BERT effectiveness for hate speech detection, specifically in Arabic are very limited.

To that end, this paper presents a set of experiments to investigate the merit of using different neural networks including Recurrent Neural Networks (RNNs) and Convolutional Neural Networks (CNNs) and their variants for hate speech detection in Arabic. We also evaluated BERT on our downstream classification task by using the pre-trained BERT model, adding a simple task-specific layer and then fine-tuning the entire model on the hate speech detection task. Given the fact that Arabic is considered as a low-resourced language, this paper also described our approach for creating a hate speech dataset for Arabic, covering racist, religious and ideological hate speech. Following this approach, we created a new dataset containing a total of 9,316 tweets labeled as normal, abusive or hateful as well as a new annotation guideline.

The main contributions of this work are three-folds:

1) Constructing a public dataset of 9,316 tweets labelled as hateful, abusive and normal.
2) Comparing the performances of three neural network models CNN, GRU, CNN+GRU for Arabic hate speech detection task.
3) Evaluating the recent language representation model BERT for Arabic hate speech detection task.

The remaining of the paper is organized as follows. Section 2 reviews work related to hate speech detection. Section 3 describes the data construction process. Section 4 describes our methods, including a detailed description of the investigated models' architectures and the preprocessing steps. Section 0 and 6 discuss the experiments and the results, respectively. Finally, Section 7 concludes our work and discusses future directions.

2 Related work

Abusive language in social media is a complex phenomenon with a wide spectrum of overlapping forms and targets. Hate speech, offensive language, and cyberbullying are examples of abusiveness and several works in the literatures have been conducted to detect and locate such types of languages.

Recently, researchers have shown an increased interest in automatic hate speech detection in social media. Most of the existing studies in the literature have primarily modeled the problem as a supervised classification task whether using classical machine learning approaches or deep neural network approaches (Fortuna & Nunes, 2018; Schmidt & Wiegand, 2017).

As for classical machine learning approaches, there were several studies that leveraged simple surface and linguistic features such as bag of words (BOW), n-grams, and POS as fundamental features. Schmidt and Wiegand (2017) presented a review of the various features that were used for the task in the literature. One of the earliest studies that addressed the problem of hate speech detection was presented by Warner & Hirschberg (2012), which focused on detecting anti-Semitic language as the specific type of hate speech. Kwok & Wang (2013) proposed a supervised machine learning model to detect racist hate speech against black people in Twittersphere and they suggested to improve the model by considering other features such as sentiment features and bigrams. Waseem and Hovey (2016) presented a study to detect hate speech in Twitter platform and specifically targeted two types of hate: racism and sexism. They also built and published a dataset of 16K annotated tweets which has become a well-known benchmarking dataset for following studies. Word embeddings as features were also investigated by Djuric et al. (Djuric et al., 2015) and the experimental results on Yahoo! Finance user comments showed that paragraph2vec representation performed better in term of Area Under the Curve (AUC). Following this study, Nobata et al. (2016) incorporated various features such as n-grams, linguistic and syntactic features and distributional semantics to detect abusive language in online user comments. The results showed that combining all features led to outperform the prior art in term of AUC. Davidson et al. (2017) trained a multi-class classifier to distinguish between offensive and hate language in tweets. They also built and published a dataset of tweets labelled for three categories: hate speech, offensive language, or neither, resulted into 24,802 labelled tweets.

In recent years, attention has shifted toward deep learning approaches to tackle the problem of hate speech detection. Most of the works have employed various architectures of Convolutional Neural Network (CNN) and Recurrent Neural Networks (RNN). Park & Fung (2017) proposed a two-step classification approach by combining two classifiers: one to classify the text to abusive or not, and another to classify the text into a specific type of abusive language (sexist or racist) given that the text is abusive. Gambäck and Sikdar (2017) proposed the use of CNN-classifier with word2vec word vectors and their results outperformed the prior art. Pitsilis et al. (2018) built an ensemble of LSTM-based classifiers to detect hate speech content in Twitter. The results showed that the proposed approach achieved better results compared to the state-of-the-art approaches. Zhang et al. (2018) showed that a CNN+GRU neural network model has improved the performance of hate speech detection. Badjatiya et al. (2017) investigated three different neural models for hate speech detection and the results showed that embeddings learned from LSTM model when combined with GBDT led to the best accuracy result. Recently, some works showed that incorporating BERT-based models to learn the representations for noisy text (Dai et al., 2020) and capturing hateful context (Mozafari et al., 2020) can enhance the offensive and hate speech detection systems.

There are several works that tackle the problem of hate speech and offensive language in non-English languages such as German (Jaki & De Smedt, 2018), Greek (Pitenis et al., n.d.), Danish (Sigurbergsson & Derczynski, n.d.), and Turkish (Çöltekin, 2020) corpora. Arabic studies that address the problem of hate speech detection are still limited, however, there are several studies that address some related problems such as detecting ISIS support messages (Magdy et al., 2015) and identifying vulgar, obscene, offensive or flaming language (Alakrot et al., 2018; Mubarak et al., 2020, 2017). A recent dataset for Levantine hate speech and abusive detection has been published by Mulki et al. (2019). This dataset contains 5,846 tweets labeled as Hate, Abusive or Normal, collected from Twitter using terms of potential entities that are usually targeted by abusive/hate speech. Albadi et al. (2018) proposed an RNN architecture with Gated Recurrent Units (GRU) and pre-trained word embeddings to detect religious hate speech in Arabic tweets. For evaluation, they created a dataset containing 6,000 Arabic tweets, 1,000 for each of the six religious' groups (Muslims, Jews, Christians, Atheists, Sunnis and Shia). The tweets were annotated as hate or not-hate tweet. The results showed that the GRU-based RNN model outperformed other traditional classifiers with respect to all evaluation. They also published their dataset as the first and the only available hate speech dataset for Arabic language.

3 Dataset construction

In this work, we built a new dataset that covers different types of hate speech, e.g., racist, religious and ideological hate speech, focusing mainly on Saudi Twittersphere. We used the standard Twitter search/streaming API with tweepy[2] python library to collect our data. The data was collected in span of 6 months, from March 2018 to August 2018 using a keyword-based and thread-based search. In keyword-based approach, we included in the search query a total of 164 impartial unique terms (excluding the various morphological variations of the same term) that refer to groups, or to people belonging to groups that are likely to be targeted by hate speech. For example, we used terms that refer to people who belong to different tribes (such as "Otaibi"," عتيبي ") and different regions ("Hijaz"," حجازي ") to collect tweets related to tribal and regional hate speech, respectively. For the thread-based search, we included in the search query different hashtag that discusses controversial topics which are considered to be strong indicators of hate speech. We monitored twitter trends during the period of the data collection, and we ended up with a total of 10 hashtags used for data retrieving. Examples of the used terms and hashtags are illustrated in Table 1.

Hate speech type	Example of Terms/hashtags
Racist	عرب الشمال, هوية_الحجاز
Regional	نجدي، قصيمي ،حجازي
Tribal	عتيبي، زهراني، قحطاني
Inter-religious	الوهابية، الشيعة، الأشاعرة
Ideological	ليبرالية، يسارية، نسوية

Table 1. Examples of terms used for data retrieving

Since hateful tweets are generally less common than natural tweets, we further boost our dataset and improve the representation of the hate class by using the top-scored terms in the lexicon of religious hate terms released by Albadi et al. (2018). For all queries, we collected only original and Arabic tweets, excluding retweets and non-Arabic tweets from the search. In total, we retrieved 54 million tweets, from

[2] https://www.tweepy.org/

which we sampled 10,000 tweets for annotation. To obtain the annotations for the dataset, Figure Eight[3] crowdsourcing platform was utilized to recruit both crowed and internal workers (volunteers and freelancers). Before submitting the task to the annotators, we generated an annotation guideline for hate speech annotation with the help of experts in interreligious dialogue, Islamic jurisprudence and media studies. The guideline is available online on GitHub[4]. The entire annotation process was completed in a course of three weeks, starting from 1 to 23 September 2019. The dataset was submitted to the annotators in different batches. The first batch (1,000 tweets) was annotated by crowed workers. The second batch (4,000 tweets) was annotated by 15 different Saudi annotators. The third and final batch (5,000 tweets) was annotated by three freelancers who are familiar with Saudi dialect. The annotated dataset is available online on GitHub[5]

4 Methods

In this section, we present the main components of our proposed approach to detect Arabic hate speech. As a first step, the tweets pass through various preprocessing steps to clean and prepare the data for the training phase. Then, several classification models for detecting hate speech in Arabic texts are investigated. In this work, we evaluated three neural network architectures: CNN-based classifier, GRU-based classifier, and finally, a classifier that combines both CNN and GRU. Moreover, we evaluated BERT (Devlin et al., 2019), a recent language representation models, on the task of hate speech detection. In the next subsections we discuss more these steps.

4.1 Data preprocessing

Before feeding the tweets as an input to the classification models, we applied several preprocessing steps:

- **Hashtags removal**: to avoid any bias toward certain classes, we dropped any hashtag used as a keyword search while collecting the data from the tweets.
- **Spam filtering:** in this step, we implemented the rule-based algorithm and lexicon proposed in (Al-Humoud et al., 2016) to filter spams with some additional terms and hashtags that we found to be highly associated with spam tweets.
- **Stop words removal:** In this step we used a list of 356 stop words that were made publicly available by (Albadi et al., 2018) on GitHub[6].
- **Emojis description:** Motivated by (Singh et al., 2019) work, we used demoji[7], a python library that provide utilities to replace emojis with their description. This library utilized readily available lists of emojis and their textual descriptions in English[8]. For our task, we translated the list into Arabic by Google translator API[9]. Then, the produced translations were manually revised and updated. The translated list was used with demoji to replace emojis in tweets with their Arabic textual description.
- **Cleaning:** In this step, punctuations, additional whitespaces, diacritics, non-Arabic characters are removed.
- **Normalization:** The goal of this step is to reduce orthographical variations observed in tweets as well as normalizing Twitter specific tokens. The normalization steps are as follows:
  - Different forms of "ا " (“ إ“ ,”أ ” and “آ”) were replaced by “ا”, “ى” is replaced by “ي” and “ة” is replaced by ‘ه”
  - Links and mentions are replaced by “URL” and “mention”, respectively.
- **Lemmatization**. We applied this step using the Farasa stemmer (Abdelali et al., 2016). We selected Farasa based on a study by El Mahdaouy et al. (2018) which showed that Farasa outperformed other tools such as MADAMIRA.

4.2 Neural network models

In this section, we first briefly describe the feature representation for the evaluated neural networks, then we describe the architectures of each model (CNN, GRU, and CNN+GRU).

[3] https://www.figure-eight.com/
[4] https://github.com/raghadsh/Arabic-Hate-speech/blob/master/Annotation%20Guidlines.pdf
[5] https://github.com/raghadsh/Arabic-Hate-speech
[6] https://github.com/nuhaalbadi/Arabic_hatespeech
[7] https://pypi.org/project/demoji/
[8] http://unicode.org/Public/emoji/12.0/emoji-test.txt
[9] https://cloud.google.com/translate/docs/

Features representation: the first layer of all proposed architectures is an embedding layer that maps each tweet, represented as a sequence of integer indexes, to a 300-dimension vector space using a pretrained word2vec model (Mikolov et al., 2013). We trained our own word2vec model using the Continuous Bag of Words (CBOW) training algorithm. We used a subset of our collected data as training collection. The collection has a total of 17.6 million tweets and 536K tokens. Before training, all tweets were preprocessed following the same preprocessing steps described earlier. As for the model's hyperparameters, we selected a window of size 3, since the length of tweets is generally small. We set the vector dimensions to 300 and the other hyper-parameters to their defaults.

CNN architecture: The first model we evaluated is a CNN model inspired by Kim's work (Kim, 2014) for text classification. Our CNN architecture, as illustrated in Figure 1, contains five layers: input layer (embedding layer), a convolution layer, which basically consists of 3 parallel convolution layers with different kernel sizes (2,3 and 4), pooling layer, hidden dense layer, and finally the output layer.

GRU architecture: Currently, GRU is the state-of-the-art model for Arabic hate speech detection. In this experiment, we employed a GRU-based network similar to the one applied in (Albadi et al., 2018). Our network architecture, as illustrated in Figure 2 contains 4 layers: input layer (embedding layer), GRU layer, hidden dense layer, and finally the output layer.

CNN+GRU architecture: The third experimented architecture is combination of both CNN and GRU, as illustrated in Figure 3. In this architecture, CNN is used as a feature extractor that passes the 'extracted feature' to a GRU layer which treats the generated feature dimension as timesteps. CNN+GRU architecture contains 6 layers: input layer (embedding layer), convolution layer with 100 filters and a kernel size of 4, max pooling layer, GRU layer, another max pooling layer, and finally the output layer.

BERT model: BERT (Devlin et al., 2019) was released in multilingual versions, pre-trained on monolingual corpora (Wikipedia) in 104 languages including Arabic. In this experiment, we fine-tuned the cased version of the multilingual pre-trained model by adding a simple classification layer that performs a binary classification to classify the tweets into hate or not hate. To prepare our data for BERT, we examined two different preprocessing procedures. The first one followed the same steps described in Section 4.1. The second one is a lite version of the first one in which stop words, punctuations and non-Arabic words are kept. The lite preprocessing showed a better result in term of F1 score on 5-fold cross validation on training set.

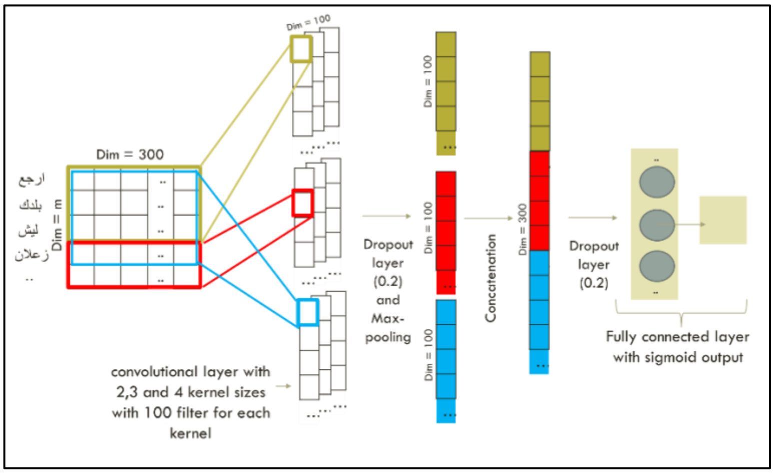
Dim = 100
Dim = 300
ارجع
بلدك
ليش
زعلان
Dim = m
Dim = 100
Dim = 100
Dim = 100
Dropout layer (0.2) and Max-pooling
Concatenation
Dim = 300
Dropout layer (0.2)
Fully connected layer with sigmoid output
convolutional layer with 2,3 and 4 kernel sizes with 100 filter for each kernel

Figure 1. Neural network models architecture

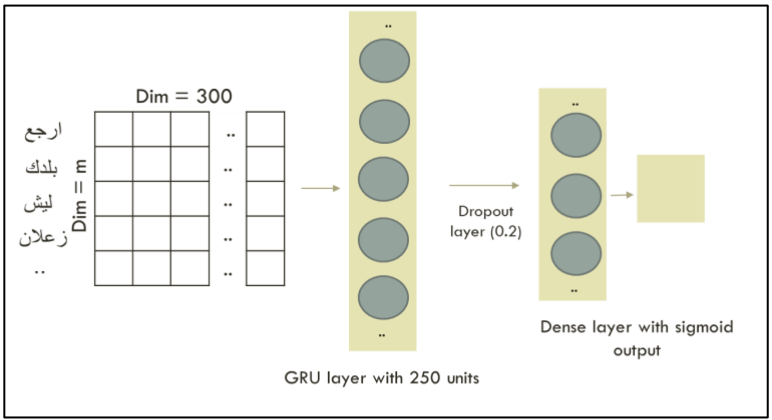

Figure 2. GRU architecture

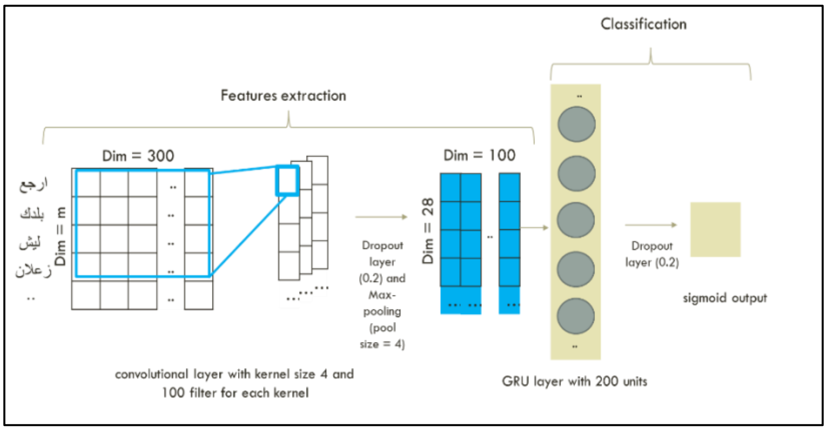

Figure 3. CNN +GRU architecture

5 Experiments

We conducted a set of experiments to evaluate the proposed models for Arabic hate speech detection tasks (CNN, GRU, CNN+GRU and BERT). In all experiments, we performed a binary classification task in which tweets are classified to hate or non-hate classes. In this section we describe the datasets used in our experiments and the experiment setup including the models' implementation, hyperparameters tuning, baselines and evaluation metrics.

5.1 Datasets

In all experiments, we used our created dataset (as described earlier in this paper) to train the proposed models for hate speech detection. The dataset contains a total of 9,316 tweets classified as hateful, abusive or normal (not hateful or abusive). Since our task is to perform a binary classification to classify the tweet into hate or non-hate speech, we only kept tweets annotated as hateful or normal. This left us with 8,964 tweets, split into 75%–25% for training (6425 instances) and testing (2539 instances). We used the 75% for training and hyperparameter-tuning through cross-validation and then we tested the optimized models on the 25% held-out data. Hereinafter, we refer to this dataset as general hate speech dataset (GHSD).

	GHSD	**RHSD (testing)**
# Tweets	8,964	600
# Hate instances	2539	250
# Non-hate instances	6425	350

Table 2. Statistics of datasets used in the experiments

To investigate to what extent the models trained on our dataset generalize to other datasets that focused on different/specific types of hate speech, we used Albadi et al. (2018) religious hate speech dataset (referred as Religious hate speech dataset (RHSD) hereinafter) to test our models. Particularly, we trained our proposed models using the whole GHSD dataset and use the testing set of RHSD, which contains 600 tweets to test the models. Statistics of the used datasets are shown in Table 2.

5.2 Experimental settings

Implementation. We implemented all neural network models using Keras library with a TensorFlow as a backend[10] while we implemented BERT using huggingface Pytorch library[11]. We run the experiments on Google Collaboratory[12], which provides a free Jupyter notebook environment with GRU accelerator.

Hyperparameters tuning. In all experimented models, we employed a grid search using 5-fold cross-validation on the training set to find the optimal training-specific and model-specific hyperparameters. All proposed network architectures were experimented with binary cross entropy as a loss function and "Adam" as an optimizer. We modified the loss function to incorporate class weights during evaluation, since our dataset is relatively imbalanced, and we are particularly interested in increasing the recall of the positive class (hate class). Then, we evaluated all network architectures with and without incorporating class weights. When fine-tuning BERT model, all hyperparameters were kept the same as in pretraining except batch size, learning rate, and number of training epochs. For batch size and number of epochs, we searched according to the suggested ranges in (Devlin et al., 2019). For the learning rate, we found that the suggested range produce poor results, so we extended the searched range to be {0.2e-5, 0.3e-5 ... 3e-5, 5e-5}.

Baselines and evaluation metrics. For our baselines, we used SVM (support vector machine) and LR (logistic regression) classifiers because they have shown to be effective in previous studies (Chen et al., 2018; Davidson et al., 2017; Waseem & Hovy, 2016). For features, we found that character n-gram features (n = 1-4) yielded the highest F1 score. We used Python scikit-learn library[13] to implement both models. We used macro-averaged Precision (P), Recall (R), accuracy, F1-measures and Area Under the Receiver Operating Characteristic curve (AUROC) for evaluation. Moreover, since our hate speech data collection is relatively imbalanced and given the serious consequences of failing to detect a hateful content, we also reported our results using positive class recall.

6 Experiments results

In this section we discuss the results of two set of experiments. The first one is the in-domain experiments, where all models were trained on the training set of our created dataset (GHSD) and tested on its testing set. We also performed a set of out-domain experiments to assess how well our models, when trained on one dataset generalize to work on different datasets. For that purpose, we trained all the models on the whole GHSD dataset, then we used Albadi et al. (2018) test set (RHSD) for testing.

	GHSD (in-domain)						**RHSD (out-domain)**					
Model	**Precision**	**Recall**	**F1**	**Accuracy**	**Hate class recall**	**AUROC**	**Precision**	**Recall**	**F1**	**Accuracy**	**Hate class recall**	**AUROC**
SVM (char n-grams)	0.74	0.74	0.74	0.78	0.63	0.85	0.70	0.66	0.66	0.68	0.46	0.73
LR (char n-grams)	0.75	0.74	0.75	0.79	0.63	0.84	0.68	0.66	0.65	0.68	0.47	0.72
CNN	**0.81**	**0.78**	**0.79**	**0.83**	**0.67**	**0.89**	0.72	**0.69**	**0.69**	**0.70**	**0.56**	**0.79**
GRU	0.80	0.77	0.78	0.82	0.63	0.87	0.70	0.65	0.64	0.68	0.43	0.76
CNN+GRU	0.80	0.76	0.77	0.82	0.62	0.88	**0.74**	0.67	0.67	**0.70**	0.44	0.77
BERT	0.76	0.76	0.76	0.80	0.65	0.76	0.63	0.60	0.59	0.63	0.38	0.6

Table 3. Evaluation results of the experimented models. Best result for each metric is boldfaced.

[10] https://keras.io/
[11] https://github.com/huggingface/transformers
[12] https://colab.research.google.com
[13] https://scikit-learn.org/

6.1 Results and discussion

Table 3 summarizes the best results achieved by our evaluated models: CNN, GRU, CNN+GRU and BERT in terms of precision, recall, F1-score, accuracy, hate class recall and AUROC. The left part of the table shows the results of the in-domain experiments. The right part of the table illustrates the out-domain experiments results.

Since neural networks are stochastic in nature, we expect some variations in models' performance at every different run. Therefore, we performed 10 runs for each experiment and reported the average of all metrics (we limited the number of runs to 10 given the time constraint on the project).

Baselines performance. Results in all metrics show that SVM and LR classifiers achieved almost the same performance on both datasets (GHDS and RHSD). We can also notice that both classifiers achieved a considerably high AUROC scores, suggesting that they were highly capable of distinguishing between the two classes.

Neural models performance on GHDS. As Table 3 shows, we experimented with three neural models, namely CNN, GRU and CNN+GRU and compared their performance against our baselines. Generally, the results show that all models outperformed the baselines by approximately 3-5% in all metrics. The only exception was the hate class recall for both GRU and CNN+GRU, in which they achieved similar and 1% lower results than the baselines, respectively. It can be noticed that CNN model achieved the best results among the three neural network models. It showed a consistent improvement in all metrics with a high AUROC score. It also achieved the highest hate class recall compared to other models. The performance improvement achieved by CNN can be attributed to its ability at extracting local and position-invariant features such as surrounding words and word orders. These features can work very well on tasks like hate speech, since hate speech, much similar to sentiment can be determined by some terms and multi-word phrases. GRU also showed very similar results to those achieved by CNN, with a minor decrease (only 1%) in the performance with respect to all metrics except for the hate class recall. While not expected, stacking GRU layer on the top of the CNN architecture led to a slight drop in CNN performance. We suspect that the reason of the performance decline is the complexity of the CNN+GRU architecture, which may not be able to work very well on relatively small datasets. As a final note, neural network models showed to be more effective in the task of hate speech detection, and it can be attributed to their abilities to capture more contextual information and long-term dependencies and hence, a deeper understanding of the texts. However, it is worthwhile to mention that our baselines were also able to achieve a very competitive results, specifically in terms of AUROC and hate class recall. This confirms previous studies that show that character n-grams can be highly predictive feature in hate speech detection.

BERT performance on GHDS. We fine-tuned BERT model for the binary text classification task by adding a simple classification layer. From Table 3, we can notice that BERT offers a slight performance improvement over the baselines with respect to almost all metrics. However, it showed a large drop (approximately 10%) in terms of AUROC compared to the baselines. Compared against our neural network models, BERT failed to provide any improvement in the classification performance. This was not expected, since BERT model has been proved to be very powerful, achieving the state-of-the-art results in many NLP tasks such as sentiment analysis, question-answering, textual entailment [94]. The poor performance of BERT can be attributed to the fact, as stated by Al-Twairesh & Al-Negheimish (2019), that BERT was trained on different dataset genre (Wikipedia).

Out-Domain experiments. The right part of Table 3 shows the results of the out-domain experiments. Not surprisingly, all models' performance, including the baselines, is always lower on the out-domain dataset. While being always lower, results of the out-domain experiments are very consistent with the results obtained in the in-domain experiments. Specifically, CNN in both set of experiments achieved the best results compared to the baselines and other models in almost all metrics. Moreover, GRU and CNN+GRU showed comparative results. However, in out-domain experiments, BERT failed to beat the baselines with respect to all metrics. There are many reasons that could cause the consistent drop in the classification performance of all experimented methods on the RHSD dataset. Firstly, RHSD focuses on different type of hate speech, namely a religious hate speech. This type of hate speech was not directly considered while collecting the GHSD dataset (used for training), which focus on wider range of hate speech types including racism, ideological, and inter-religious. In fact, this could cause the two datasets to have different types of swear/abusive/offensive terms and also different targets names. This can greatly impact the lexical distribution of the two datasets, and hence, will impact how/what the models learn as predictive features. To investigate this, we firstly generated two lexicons from our dataset (GHSD) following two corpus statistical-based approach: chi-square (x^2) and Pointwise Mutual

Information (PMI), which leverage the labeled corpus in order to learn a domain-specific lexicon and measures the association strength between a term and a class (hate or non-hate classes, in our case). More details on how x^2 and PMI values are calculated can be found in (Kaji & Kitsuregawa, 2007). Then, we examined the lexicons generated from RHSD by Albadi et al. (2018) and from GHSD and analyzed the terms with high positive scores (strongly associated with hate class). We found that some of the highly scored terms in RHSD have a very low score (nearly zero) in GHSD. Examples of these words are the religious affiliations such as *"Christian/نصراني"* and *‘’polytheist /مشرك’’*. Beside the lexicon differences, we notice that tweets in RHSD testing set were mostly written in MSA, while tweets in GHSD dataset were mostly written in Saudi dialect. The differences of the linguistic characteristics between MSA and dialects could also affect the ability of our in-domain models to generalize well. In the next section, we attempt to analyze the error produced by our evaluated models to gain deeper understanding of the limitation and the challenges of the hate speech detection task.

6.2 Error analysis

To understand the challenges of this task, we carry out further analysis of the errors made by our models, including the baselines. We identified the hateful tweets that all evaluated models on both datasets predicted incorrectly (misclassified as non-hate). We ended up with a total of 151 tweets for GHSD and 51 tweets for RHSD. We further qualitatively analyzed these tweets and we found that nearly 90% of the misclassified tweets do not have offensive/abusive terms, this could confuse the classifiers and cause its performance to be degraded.

We also found that many tweets would be even difficult for human to classify as hate without being provided with the full context of the tweet. For example, some of the tweets contain comments on images or videos, which will be very tricky to interpret without seeing the associated media. The lack of the context problem also appeared with tweets that refer to external links or quoted tweet. For example, this tweet quoted another tweet and with this comment *"This is serious, this means that those people will be back to work in secret so that no one will pay attention to them"*. This tweet was written by an extremely patriotic user who uses "those" to refer to foreigners working in Saudi Arabia, indicating they have some agenda to call for separation of Hijaz region from the rest of the country. This tweet is difficult to be interpreted correctly without knowing what it quoted exactly. Mentions also have the same problem, some mentions contain only few natural words, but it should be classified as hate when we take into the account the original tweet that it replied to. For example, this tweet *"@user Qahtani freshman"* replied to an offensive tweet that negatively stereotypes a particular tribe's members, but the classifiers incorrectly classified it as non-hate. Finally, we found that some tweets were indirect, implicit or sarcastic, which make it difficult of automatic solutions to detect correctly. For example, this sarcastic tweet *"Be hold for now we present a play called 'Freemen with #Mohammad_Bin_Salman in #Saudi_The_Great'"* was misclassed as non-hate, but it was written by non-Saudi user against a Saudi user as a part of hateful conversation between them.

We also performed further analysis to understand the cases in which the classifiers misclassified non-hate tweets as hateful. We ended up with 33 misclassified non-hate tweets in GHSD and 20 tweets in RHSD. Mostly all of these tweets include one or more religious, national or ideological affiliations terms, which are highly associated with hate class (according to our generated lexicons). Examples: *"Hahaha... Fear Allah brother, first of all Khurasan scholars are not related to the Persians Magians! Second Arab was and still the head of knowledge"* and *"Hahaha... you blended with our Hadarms*[14] *ladies so that's why they consider you one of them; that's the way our parents raised us even if you're Saudi"*. These tweets may cause some confusion that leads the classifiers to classify them as hateful. We also found that some of misclassified tweets are debatable and could be hard to decide whether to consider it hate or non-hate even for human annotators, such as this tweet: "*Most of them moved from one radicalism to another; and what we got here in this world are people called the new Arab liberals and you can see what's on their shoulders (different Ideas that contradict the simplest principles of liberalism)*".

[14] From/of Hadramaut; Hadramauti.

All these factors led us to conclude that hate speech is still a difficult phenomenon. It is highly dependent on the context and, in fact can be conveyed in a very polite and genuine form (without using any offensive words or being aggressive).

7 Conclusion and future work

Hate speech in Arabic Twittersphere has become a notable problem, resulted in a pressing need for effective automatic solution for hate speech detection. In this work, we constructed a public dataset of 9,316 tweets labeled as hateful, abusive, and normal. We evaluated and compared four different models: CNN, GRU, CNN+GRU and BERT. The obtained results from our experiments were promising, showing the effectiveness of the proposed models in the detection task. The results showed that CNN successfully outperformed other models, with an F1-score of 0.79 and AUROC of 0.89. Our results also showed that BERT failed to improve over the baselines and the other evaluated models. This could be attributed to the fact that BERT was trained on different dataset genre (Wikipedia).

We believe that there are several ways to extend and improve this study in the future. The dataset can be extended to capture more writing styles, patterns and topics. The dataset can also be annotated with multi-labels to enhance the results of the detection task beyond the binary classification. For example, annotating the target of the hate speech and the aggressive level (week/strong) could introduce new direction for future works and applications.

We also think that the conducted experiments could be enhanced in different ways. First, we aim to extend our out-domain experiments and evaluate the proposed models on other abusive/offensive language datasets such as those found in (Alakrot et al., 2018; Mubarak et al., 2017). This could help us to understand how to distinguish between hate speech and abusive/offensive language. Moreover, different embeddings methods could be investigated beside Word2Vec such as ELMo (Peters et al., 2018), BERT, FastText (Joulin et al., 2016), and the Universal Sentence Encoder (USE) (Cer et al., 2018). Other features can be also incorporated with word embeddings such as the user's gender, age and location. Moreover, to alleviate the lack of the context problem we found when we analyzed the errors made by our models, we aim to incorporate tweets' context (e.g. the original tweet of the replies, quoted tweets, text from the external links) into the feature space and investigate if this could boost the performance of the classifiers.

References

Abdelali, A., Darwish, K., Durrani, N., & Mubarak, H. (2016). Farasa: A Fast and Furious Segmenter for Arabic. *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Demonstrations*, 11–16.

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

Albadi, N., Kurdi, M., & Mishra, S. (2018). Are they Our Brothers? Analysis and Detection of Religious Hate Speech in the Arabic Twittersphere. *2018 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM)*, 69–76. https://doi.org/10.1109/ASONAM.2018.8508247

Al-Hassan, A., & Al-Dossari, H. (2019). DETECTION OF HATE SPEECH IN SOCIAL NETWORKS: A SURVEY ON MULTILINGUAL CORPUS. *Computer Science & Information Technology(CS & IT)*, 83–100. https://doi.org/10.5121/csit.2019.90208

Al-Humoud, S., Al-Twairesh, N., Altuwaijri, M., & Almoammar, A. (2016). *Arabic Spam Detection in Twitter*.

Al-Twairesh, N., & Al-Negheimish, H. (2019). Surface and Deep Features Ensemble for Sentiment Analysis of Arabic Tweets. *IEEE Access*, *7*, 84122–84131. https://doi.org/10.1109/ACCESS.2019.2924314

Badjatiya, P., Gupta, S., Gupta, M., & Varma, V. (2017). Deep Learning for Hate Speech Detection in Tweets. *Proceedings of the 26th International Conference on World Wide Web Companion*, 759–760. https://doi.org/10.1145/3041021.3054223

Cer, D., Yang, Y., Kong, S., Hua, N., Limtiaco, N., John, R. S., Constant, N., Guajardo-Cespedes, M., Yuan, S., Tar, C., Sung, Y.-H., Strope, B., & Kurzweil, R. (2018). Universal Sentence Encoder. *ArXiv:1803.11175 [Cs]*. http://arxiv.org/abs/1803.11175

Chen, H., McKeever, S., & Delany, S. J. (2018). A Comparison of Classical Versus Deep Learning Techniques for Abusive Content Detection on Social Media Sites. In S. Staab, O. Koltsova, & D. I. Ignatov (Eds.), *Social Informatics* (pp. 117–133). Springer International Publishing.

Çöltekin, Ç. (2020). A Corpus of Turkish Offensive Language on Social Media. *Proceedings of The 12th Language Resources and Evaluation Conference. 2020.*, 6174–6184. https://www.aclweb.org/anthology/2020.lrec-1.758

Dai, W., Yu, T., Liu, Z., & Fung, P. (2020). Kungfupanda at SemEval-2020 Task 12: BERT-Based Multi-Task Learning for Offensive Language Detection. *ArXiv:2004.13432 [Cs]*. http://arxiv.org/abs/2004.13432

Darwish, K., Magdy, W., & Mourad, A. (2012). Language processing for arabic microblog retrieval. *Proceedings of the 21st ACM International Conference on Information and Knowledge Management*, 2427–2430.

Davidson, T., Warmsley, D., Macy, M., & Weber, I. (2017). Automated Hate Speech Detection and the Problem of Offensive Language. *ArXiv:1703.04009 [Cs]*. http://arxiv.org/abs/1703.04009

Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2019). BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *ArXiv:1810.04805 [Cs]*. http://arxiv.org/abs/1810.04805

Djuric, N., Zhou, J., Morris, R., Grbovic, M., Radosavljevic, V., & Bhamidipati, N. (2015). Hate Speech Detection with Comment Embeddings. *Proceedings of the 24th International Conference on World Wide Web - WWW '15 Companion*, 29–30. https://doi.org/10.1145/2740908.2742760

El Mahdaouy, A., El Alaoui, S. O., & Gaussier, E. (2018). Word-embedding-based pseudo-relevance feedback for Arabic information retrieval. *Journal of Information Science*, 016555151879221. https://doi.org/10.1177/0165551518792210

Fortuna, P., & Nunes, S. (2018). A Survey on Automatic Detection of Hate Speech in Text. *ACM Comput. Surv.*, *51*(4), 85:1–85:30. https://doi.org/10.1145/3232676

Gambäck, B., & Sikdar, U. K. (2017). Using Convolutional Neural Networks to Classify Hate-Speech. *Proceedings of the First Workshop on Abusive Language Online*, 85–90. https://doi.org/10.18653/v1/W17-3013

Jaki, S., & De Smedt, T. (2018). Right-wing German hate speech on Twitter: Analysis and automatic detection. *Manuscript Submitted*.

Joulin, A., Grave, E., Bojanowski, P., & Mikolov, T. (2016). Bag of Tricks for Efficient Text Classification. *ArXiv:1607.01759 [Cs]*. http://arxiv.org/abs/1607.01759

Kaji, N., & Kitsuregawa, M. (2007). Building Lexicon for Sentiment Analysis from Massive Collection of HTML Documents. *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL*.

Kim, Y. (2014). Convolutional Neural Networks for Sentence Classification. *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 1746–1751. http://www.aclweb.org/anthology/D14-1181

Kwok, I., & Wang, Y. (2013). Locate the hate: Detecting tweets against blacks. *Twenty-Seventh AAAI Conference on Artificial Intelligence*.

Magdy, W., Darwish, K., & Weber, I. (2015). #FailedRevolutions: Using Twitter to Study the Antecedents of ISIS Support. *ArXiv:1503.02401 [Physics]*. http://arxiv.org/abs/1503.02401

Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013). Efficient Estimation of Word Representations in Vector Space. *ArXiv:1301.3781 [Cs]*. http://arxiv.org/abs/1301.3781

Mozafari, M., Farahbakhsh, R., & Crespi, N. (2020). A BERT-Based Transfer Learning Approach for Hate Speech Detection in Online Social Media. In H. Cherifi, S. Gaito, J. F. Mendes, E. Moro, & L. M. Rocha (Eds.), *Complex Networks and Their Applications VIII* (pp. 928–940). Springer International Publishing. https://doi.org/10.1007/978-3-030-36687-2_77

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. http://www.aclweb.org/anthology/W17-3008

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

Mulki, H., Haddad, H., Bechikh Ali, C., & Alshabani, H. (2019). L-HSAB: A Levantine Twitter Dataset for Hate Speech and Abusive Language. *Proceedings of the Third Workshop on Abusive Language Online*, 111–118. https://doi.org/10.18653/v1/W19-3512

Nobata, C., Tetreault, J., Thomas, A., Mehdad, Y., & Chang, Y. (2016). Abusive Language Detection in Online User Content. *Proceedings of the 25th International Conference on World Wide Web - WWW '16*, 145–153. https://doi.org/10.1145/2872427.2883062

Park, J. H., & Fung, P. (2017). One-step and Two-step Classification for Abusive Language Detection on Twitter. *Proceedings of the First Workshop on Abusive Language Online*, 41–45. https://doi.org/10.18653/v1/W17-3006

Peters, M. E., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K., & Zettlemoyer, L. (2018). Deep contextualized word representations. *ArXiv Preprint ArXiv:1802.05365*.

Pitenis, Z., Zampieri, M., & Ranasinghe, T. (n.d.). *Offensive Language Identification in Greek*. 7.

Pitsilis, G. K., Ramampiaro, H., & Langseth, H. (2018). Effective hate-speech detection in Twitter data using recurrent neural networks. *Applied Intelligence*, *48*(12), 4730–4742. https://doi.org/10.1007/s10489-018-1242-y

Schmidt, A., & Wiegand, M. (2017). A Survey on Hate Speech Detection using Natural Language Processing. *Proceedings of the Fifth International Workshop on Natural Language Processing for Social Media*, 1–10. http://www.aclweb.org/anthology/W17-1101

Sigurbergsson, G. I., & Derczynski, L. (n.d.). Offensive Language and Hate Speech Detection for Danish. *Proceedings of The 12th Language Resources and Evaluation Conference*, 3498--3508.

Singh, A., Blanco, E., & Jin, W. (2019). Incorporating Emoji Descriptions Improves Tweet Classification. *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, 2096–2101. https://doi.org/10.18653/v1/N19-1214

Warner, W., & Hirschberg, J. (2012). Detecting Hate Speech on the World Wide Web. *Proceedings of the Second Workshop on Language in Social Media*, 19–26. http://dl.acm.org/citation.cfm?id=2390374.2390377

Waseem, Z., & Hovy, D. (2016). Hateful Symbols or Hateful People? Predictive Features for Hate Speech Detection on Twitter. *Proceedings of the NAACL Student Research Workshop*, 88–93. https://doi.org/10.18653/v1/N16-2013

Zhang, Z., Robinson, D., & Tepper, J. (2018). Detecting Hate Speech on Twitter Using a Convolution-GRU Based Deep Neural Network. In A. Gangemi, R. Navigli, M.-E. Vidal, P. Hitzler, R. Troncy, L. Hollink, A. Tordai, & M. Alam (Eds.), *The Semantic Web* (pp. 745–760). Springer International Publishing.