Hate Speech Detection using Word Embedding and Deep Learning in the Arabic Language Context

Hossam Faris[1], Ibrahim Aljarah[1] Maria Habib[1] and Pedro A. Castillo[2]
[1]*Department of Information Technology, King Abdullah II School for Information Technology, The University of Jordan, Amman, Jordan*
[2]*Department of Computer Architecture and Technology, ETSIIT - CITIC, University of Granada, Spain*

Keywords: Hate Speech, Classification, Word Embedding, Machine Learning.

Abstract: Hate speech over online social networks is a worldwide problem that leads for diminishing the cohesion of civil societies. The rapid spread of social media websites is accompanied with an increasing number of social media users which showed a higher rate of hate speech, as well. The objective of this paper is to propose a smart deep learning approach for the automatic detection of cyber hate speech. Particularly, the detection of hate speech on Twitter on the Arabic region. Hence, a dataset is collected from Twitter that captures the hate expressions in different topics at the Arabic region. A set of features extracted from the dataset based on a word embedding mechanism. The word embeddings fed into a deep learning framework. The implemented deep learning approach is a hybrid of convolutional neural network (CNN) and long short-term memory (LSTM) network. The proposed approach achieved good results in classifying tweets as Hate or Normal regarding accuracy, precision, recall, and F1 measure.

1 INTRODUCTION

Hate speech is the use of offensive, abusive, or insulting language towards an individual or a minority of people. The objective of hate speech is disseminating hatred and discrimination based on the grounds of race, sex, religion, or disability. The European Court of Human Rights (ECHR) identifies the concept of hate speech as any use of the language expressions that spread, encourage, or incite hatred based on race or xenophobia, and any form of intolerance towards immigrants or minorities (Court,).

Twitter is a social networking site and a microblogging service. Social networking sites facilitate the ability of freely communicating ideas and opinions among end users. The immense use of social networking sites results in a large amount of data that can be analyzed by smart machine learning algorithms. The use of machine learning and data mining for interpreting such big data provides the ability to capture the hidden pattern of information. Hence, a higher potential for detecting the hatred patterns of data.

Natural Language Processing (NLP) is concerned with applying different statistical preprocessing techniques. The purpose of NLP techniques is transforming the textual datasets into datasets that are feasible by machine learning algorithms. Such NLP processes are data normalization, stemming, tokenization, and features extractions. However, NLP processes face several obstacles when handling complex languages.

The Arabic language is the fourth used language on the Internet and ranked as the sixth used language on Twitter (Inc.,). Arabic is rich and complex morphological language that exposes the Arabic text classification problem to many challenges. In essence, different factors made the Arabic NLP techniques relatively hard to implement. For instance, Arabic language has different forms such as the dialectical Arabic. The dialectical colloquial Arabic is the most used form of Arabic on social media networks. Yet, each Arabic country has different dialect, which makes the text preprocessing more difficult than the processing of simple languages. Further, the colloquial Arabic has many misspellings that differs morphologically and phonologically. Also, Arabic has complex orthography and morphosyntactic rules. All these factors made the Arabic preprocessing techniques much complex (Badaro et al., 2019; Alrefai et al., 2018).

In this paper, a dataset is created that is targeting the problem of hate speech on Twitter at the Arabic countries. The data was collected using several keywords such as racism, sport, and Islam. Mainly,

Faris, H., Aljarah, I., Habib, M. and Castillo, P.
Hate Speech Detection using Word Embedding and Deep Learning in the Arabic Language Context.
DOI: 10.5220/0008954004530460
In *Proceedings of the 9th International Conference on Pattern Recognition Applications and Methods (ICPRAM 2020)*, pages 453-460
ISBN: 978-989-758-397-1; ISSN: 2184-4313

the dataset contains two classes; the Hate class and the Normal class. The word embedding technique is used for extracting a set of words features that can capture the hidden relations of words of the dataset. The utilized word embeddings are the Word2Vec and the AraVec implementations. Keras (Gulli and Pal, 2017) is a deep learning framework used for the implementation of the deep learning model. The proposed deep learning model is a recurrent convolutional network, which is a combination of convolutional network layer and LSTM network. The proposed methodology is evaluated based on different performance evaluation measures including the accuracy, precision, recall, and F1 measure. The results of detecting hate tweets based on deep learning framework were very good. Yet promising for further research as the problem of cyber hate speech detection over Arabic countries is poorly investigated.

The rest of the paper is organized as follows. Section 2 gives a review of related works of cyber hate speech detection. Section 3 is the proposed methodology including a description of the dataset, the word embedding and deep learning frameworks, in addition to the evaluation performance metrics. Section 4 is a discussion of the conducted experiment and obtained results. Finally, Section 5 is the concluding remarks and potential future works.

2 RELATED WORKS

Recently, Arabic natural language processing has been sparsely studied. Yet, cyber hate speech detection in Arabic context is poorly investigated. However, this section reviews previous research studies for cyber hate speech detection in Arabic context.

Authors in (Al-Hassan and Al-Dossari, 2019) presented the main challenges for articulating hate speech over Arabic online social networks. Where they stated that the colloquial Arabic has many grammatical and spelling mistakes. Also, in some Arabic countries there are words considered hate, while in other Arabic countries they are normal. Further, authors claimed that all conducted studies in this area suffer of low recall and precision values. One of the early attempts for detecting hate speech of Arabic tweets can be found in (Abozinadah et al., 2015), in which, three types of features were extracted that are profile-based features, tweet-based features represented by Term Frequency (TF) and Term Frequency-Inverse Document Frequency (TF-IDF) models, and social graph features. Their proposed approach implemented traditional machine learning algorithms including Naive Bayes (NB), Support Vector Machine (SVM), and decision tree (DT), where it achieved very good performance in terms of recall, precision, and f-measure.

In (Mubarak et al., 2017), authors created a corpus of cyberbullying words, which used for abusive language detection over Arabic social media. Whereas, (Alakrot et al., 2018) constructed a dataset for offensive speech detection on YouTube, covering several Arabic dialects. The constructed dataset encompasses three classes; offensive, inoffensive, and neutral. Additionally, (Albadi et al., 2018) constructed a dataset for religious hate speech detection over Arabic Twitter environment. In the proposed approach, authors developed an Arabic lexicon that contains commonly used religious terms with their polarities. Also, the constructed dataset is applied into different classification models involving a lexicon-based, N-gram-based, and a deep learning-based approach. Where the implementation of a Recurrent Neural Network with Gated Recurrent Unit and a pre-trained word embedding model achieved (84%) in terms of Area Under Curve measure (AUC).

Nonetheless, (Mulki et al., 2019) collected a Twitter dataset for hate speech and abusive language detection in Arabic context. The created dataset is a benchmark dataset known as (L-HSAB). The proposed dataset is classified into three classes; normal, abusive, and hate. However, authors applied the N-gram and TF word representation models into SVM and NB classifiers, where the results were very good in terms of accuracy, recall, precision, and f-measure. Similarly, (Haddad et al., 2019) designed a dataset for hate speech detection for Tunisian dialect, which aims for an automatic prevention of any toxic language. While (Bleiweiss,), presented an LSTM approach based transfer learning for abusive speech detection on Twitter, which accomplished very good results in regard to F-measure.

To the best of our knowledge, very few studies investigated cyber hate speech detection on Arabic online context. Even that, cyber hate speech detection is also presented in other different languages such as English, Italian, and Indonesian. For instance, (Watanabe et al., 2018) proposed an approach for hate speech detection on Twitter, in which, unigram and sentiment features were extracted and fed into SVM, DT, and Random forest (RF), achieving good performance regarding accuracy, recall, precision, and f-measure. While (Pitsilis et al., 2018) presented an ensemble approach of recurrent neural networks, for the detection of racism and sexism on Twitter. In (Del Vigna12 et al., 2017), authors introduced an approach for Italian hate speech detection on Facebook, where several syntactical, sentimental, and word embedding

features were extracted and adopted into SVM and recurrent neural networks. As a result, the proposed approach showed promising performance. In addition, (Polignano and Basile, 2018) presented an ensemble of deep neural networks for Italian hate speech detection.

The authors in (Fauzi and Yuniarti, 2018) presented the application of hate speech detection in Indonesia. Where the collected dataset is applied into various machine learning algorithms such as SVM, NB, and k-nearest neighbors. Nonetheless, there are other efforts in other languages including the Spanish (Ortiz et al., 2019), Turkish (Şahi et al., 2018), German (Jaki and De Smedt, 2018), and Sri Lanka (Wijeratne, 2018).

3 METHODOLOGY

In order to develop an intelligent approach for cyber hate speech detection, a methodology is designed. First step is collecting and preparing the dataset. Second is transforming the texts (tweets) into features that are comprehensible by the machine learning algorithms. Third is building the smart prediction model that is a deep learning model. Finally, is the evaluation of the model in detecting Hate tweets. Fig. 1 shows a summary of the designed methodology.

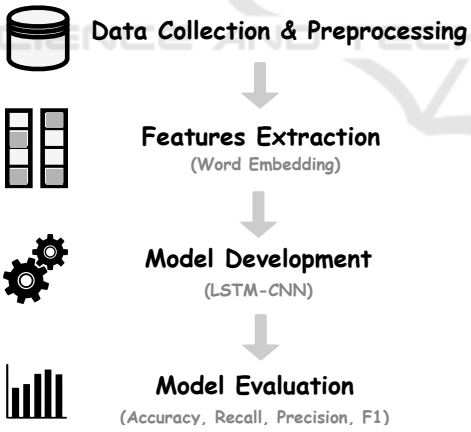

Figure 1: An overview flowchart of the methodology.

3.1 Dataset

The dataset is collected from Twitter using Twitter streaming Application Programming Interface (API) and "rtweet" library [1]. The collection of data covers different critical and debatable areas including sport, religion, racism, and journalism. Table 1 shows the used keywords for collecting the tweets. The total number of collected tweets after removing duplicates and irrelevant tweets is 3696.

The collected tweets were annotated based on the overall perceived meaning of the tweet into Hate or Normal. The total number of Hate tweets is 843, while the number of Normal tweets is 791. The rest of the tweets accounts for 2062 and were annotated as Neutral, which does not exhibit neither Hate nor Normal orientations. The dataset is a combination of merely Hate and Normal classes, where the positive class is the Hate and the negative class is the Normal class. Table 1 shows the count of the tweets and the distribution of the classes per each used keyword. From Table 1, it is obvious that the rate of Hate tweets is the largest in case of religion and terrorist, as well as in case of immigrants and journalism even they are slightly less.

Primarily, the process of preparing the data is described in Fig. 2. It starts by collecting the tweets using R language and Twitter APIs, annotating the data by two volunteers, cleaning the data from any irrelevant or redundant tweets, normalizing the tweets, tokenizing tweets, and text vectorization for the objective of features representation.

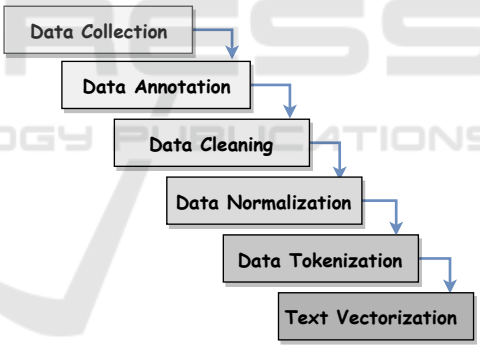

Figure 2: The steps of data collection and preprocessing.

In data cleaning, all non-Arabic characters, numbers, symbols, punctuation, hashtags, web addresses, diacritics, and the Arabic stop words were filtered out. Whereas, tweets normalization is the process of converting the Arabic characters from the standard-writing way into the colloquial-writing way. For instance, the Alef Arabic character is transformed from أ, إ, آ, ٱ into ا. Another example is transforming ي into ى.

Tweets tokenization is concerned with dividing the tweets into set of words based on the white-space delimiter. Herein, the tokenization process is implemented using the Natural Language Toolkit (NLTK) library (Loper and Bird, 2002) within Python development framework. Text vectorization is the pro-

[1] rtweet: https://rtweet.info/

Table 1: Description of used keywords for collecting tweets and the percentage of hate and normal classes.

Keyword	Translated Keyword	Tweets#	Positive (%)	Negative(%)
الوحدات	Alwahadat sport club	14	28.6%	71.4%
الدوري الاردني	The Jordanian league	44	13.6%	86.4%
الفيصلي الاردن	Faisaly Jordan	24	41.7%	58.3%
الاسلام والارهاب، تدمير الاسلام	Islam and terrorism, damage Islam	100	82.0%	18.0%
العنصرية	Racism	1193	46.8%	53.2%
لاجئ، لاجئون	Refugees	240	70.0%	30.0%
الحرية، الاعلام، الوطن، ناهض حتر، يساري متطرف	Freedom, media, homeland, Nahed Hattar, extreme	19	78.9%	21.1%

cess of transforming the raw textual data into a collection of numerical features. Mainly, the numerical features express statistical measurements of the words. The collections of features are represented like vectors; thus, it is called text vectorization. In literature, there are diverse models for text vectorization. The most popular models are the Bag-of-Words, Term Frequency, and Term Frequency-Inverse Document Frequency. However, these models cannot capture the underlying semantic of the words. Therefore, a more resilient approach for understanding the implicit relationships between words is required which is the word embedding. In this article, the Word2Vec word embedding model is utilized.

3.2 Word Representation

Recently, several representations have been proposed for representing texts in a way that is understandable by machine or deep learning algorithms. Taking an example is the one-hot encoding approach. In the one-hot encoding, all unique words are extracted and represented in a vector. Then each word is encoded into a numerical vector of a length equals to the number of unique words. Where the numerical vector is a vector of zeros except at the index of the respective word is set to one. This model is inefficient as it cannot grasp the implicit meaning of words, while the words are mainly represented as vectors of zeros.

A more robust approach is the word embedding mechanism. In word embedding approach, all words of similar meaning have an analogous encoding. Embedding means a dense vector, where the length of the vector is a parameter that is set previously. Also, the components of the dense vector are parameters that are learned during the training process. A higher dimension of the embedding corresponds to a better ability for learning the semantic meaning of words, but also the need for more large training data.

One of the models that is developed by the word embedding approach is the Word2Vec (Goldberg and Levy, 2014). Word2Vec is a neural network of three-layers, the input layer, the hidden layer, and the output layer. However, this architecture of neural networks is merely used to learn the weights of the hidden layer that are the words embeddings. Mainly, Word2Vec has two structures of implementations; the Continuous Bag-of-Words (CBoW) and the Skip-Gram (SG) models. The CBoW model depends on several neighboring words to predict a certain middle word. Whereas, SG model is the opposite, that is used to predict the neighboring words for a single determined word. Fig. 3 depicts the main principle of CBoW and SG models. In which, the CBoW model takes a set of input words, where the number of words is determined by a window size. When the window size is two; four words are considered to predict the corresponding word (w(t)). The four words are two preceding words w(t-1), w(t-2) and another two following words w(t+1), w(t+2). While in the SG model, a single word is considered to predict the neighboring words. Predicting the neighboring words is accomplished by finding the highest probability words that are the most similar to the corresponding word.

In this paper a pre-trained word embedding model is utilized. AraVec is an open source implementation of word embedding for the Arabic NLP processes. AraVec created 12 different models based on Arabic content from Twitter, Wikipedia, and general Arabic websites, in which, the total number of used vocabularies is 3,300,000,000, all integrated within SG and CBoW models (Soliman et al., 2017). The word embedding inside the deep learning framework is represented by an embedding layer. Each embedding layer is also known as a lookup table that has an input length and a dimension length. The input length is the size of the unique vocabulary of the respective dataset. While, the dimension length is a parameter specifies the length of the embedding. The dimension length should be fit to suite the problem and the size of the dataset.

3.3 Recurrent Convolutional Network Architecture

This subsection presents the proposed approach for detecting the hate orientation of tweets. A general architecture of the proposed deep learning approach is

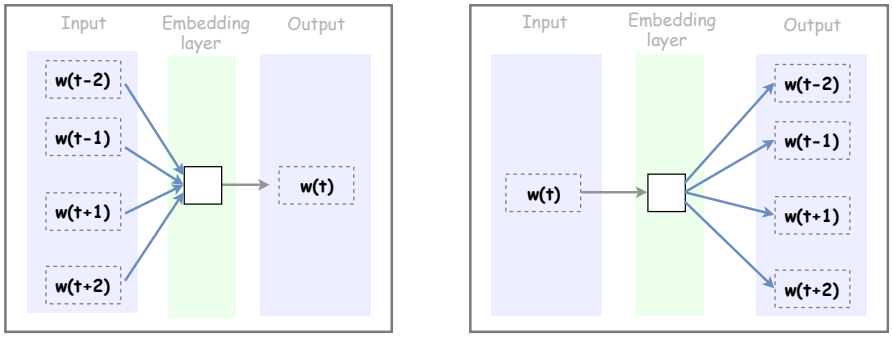

Figure 3: An illustration of continuous bag-of-words and skip gram word embedding models.

presented in Fig. 4, which is a sequential layered steps start by the embedding layer, the dropout layer, the convolutional layer, the max pooling layer, the LSTM layer, the dense layer, and the output layer. The embedding layer is constructed using a word embedding approach which is the Word2Vec, as discussed in the previous sub-section. The need of the dropout layer is for avoiding the tendency of overfitting.

Even that the embedding layer encoded each word into a unique vector. But now the objective is how to propose a model that understands a textual sentence to decide if it carries Hate or Normal expressions. Given that the sentence is a sequence of words. One of the approaches to represent a variable length sentences (sequences) is the convolutional approach. The convolutional layer takes a sequence of embedding vectors as an input and produce a compressed tensor as an output. The convolutional layer has a collection of filters that take a group of words of variable window-sized in order to predict new potential features (words). Fig.5 represents a convolutional layer. The aim of the max-pooling layer is to reduce the dimension of the produced tensor by applying a max filter.

LSTM is a kind of artificial Recurrent Neural Networks (RNN). An LSTM unit acts like a memory cell. The structure of the cell includes input and output activation layers in addition to three gates. The input gate, the output gate, and the forget gate. The objective of LSTM network is to solve the problem of the vanishing gradient of neural networks (Gers et al., 1999). In addition, LSTM is very efficient for handling sequences of data, or sequences of words in case of textual data (Collobert et al., 2011).

The fully connected layer (the dense layer) takes the output of the LSTM and converts it into class labels or probabilities. The output of the dense layer goes through an output layer with Sigmoid activation, since the problem is a binary classification.

3.4 Evaluation Measures

The used performance evaluation measures are accuracy, precision, recall, and F1 measure. The accuracy is the ratio of correctly classified Hate and Normal tweets over all the correct and the incorrect number of classified tweets. Where the accuracy is formulated in Eq. 1 (Han et al., 2011).

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \quad (1)$$

The precision metric is identified as the ratio of tweets that correctly identified as Hate over the total number of Hate tweets, which is represented in Eq. 2.

$$Precision = \frac{TP}{FP+TP} \quad (2)$$

Whereas, the recall (known as sensitivity) quantifies how much the classifier can recognize Hate tweets that is given by Eq. 3.

$$Recall = \frac{TP}{FN+TP} \quad (3)$$

The F1 measure is a metric for indicating the balance between precision and recall measures, represented by Eq. 4.

$$F1 = 2\,x\,\frac{Precision\ x\ Recall}{Precision+Recall} \quad (4)$$

4 EXPERIMENTS AND RESULTS

This section presents the experimental settings for applying the collected hate speech dataset into a deep learning framework. Keras is used as a deep learning framework, and utilized using Spyder platform (Raybaut, 2009) and Python version 3.7.

The dataset is divided into (80%) for training and (20%) for testing. The dataset is approximately balanced at the class level, since the positive class accounts for 52%, whereas the negative class is 48%.

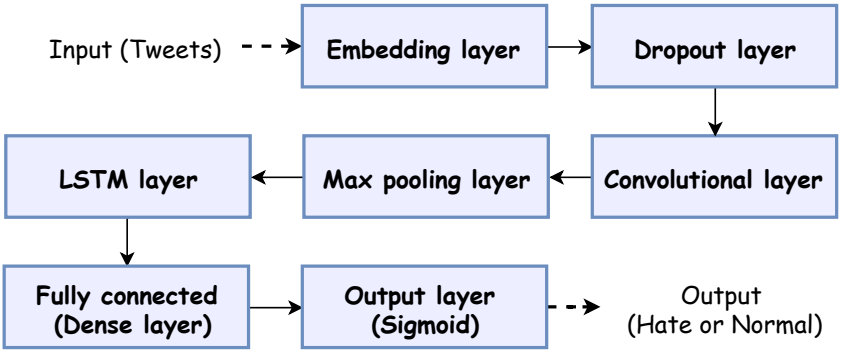

Figure 4: Proposed deep learning architecture.

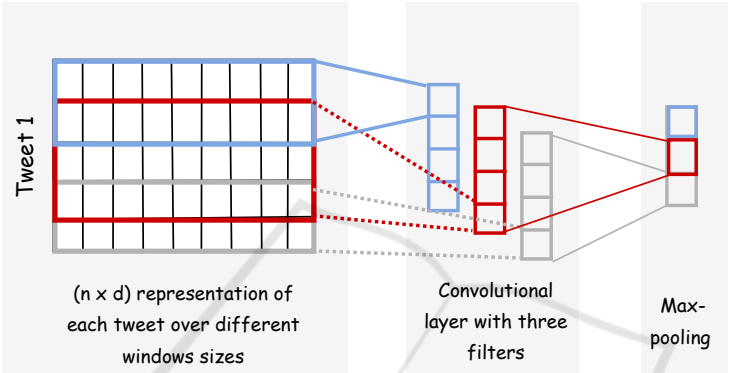

Figure 5: A mapping between the embedding layer, the convolutional layer, and the max-pooling layer.

The problem of detecting hate speech is formulated as binary classification problem; where the "1" labeled class is the Hate and "0" is the Normal.

The unique tokens of the dataset were stemmed using ISRI Arabic stemmer from NLTK library (Loper and Bird, 2002). ISRI stemmer is a light stemmer for filtering out the affixes of words. Thus, the stemmed dataset is adopted into two different implementations of pre-trained word embedding approaches. The used embedding approaches are the Word2Vec and the AraVec. The Word2Vec is an implementation from Gensim library (Rehurek and Sojka, 2011). In which, the window size is set to 5, the number of workers is 4, and the minimum count of word occurrences is 1. For the AraVec approach, two models were used. The full N-gram with SG model and the full N-gram with CBoW model. The two models are pre-trained using Twitter with embedding dimension (100). In consequence, using either the Word2Vec or the AraVec the embedding layer is constructed.

A sequential model of deep learning is designed. The sequential deep learning model is a layered approach, which means that the model is created of stacked layers. Where each layer has different function. All tweets sequences before entering the deep learning model were padded to the maximum length of the tweets. Hence, the first layer is the previously constructed embedding layer. Next is the dropout layer with a rate of 25%. Then is the one-dimensional convolutional layer (conv1D). The conv1D layer convolves with the input in one dimensional space in order to create the output tensor. The number of filters of the conv1D layer is set to 64, the kernel size is 5, the used activation is "relu", and the number of strides is 1. For the max-pooling layer, the pool size is set to 4, while the LSTM output size is set to 70. The output layer is designed of one unit with Sigmoid activation function.

The model is compiled with the "Adam" optimizer which is an adaptive learning optimization algorithm. The loss parameter of the optimizer is set to "binary-crossentropy" that is recommended for binary classification problems. Table 2 shows the performance evaluation results of the proposed deep learning model.

The recurrent convolutional model is tested over different number of epochs as shown in Table 2. Obviously, by comparing the performance of the Word2Vec and the AraVec, AraVec accomplishes superior results in terms of all metrics. Hence, achieved best accuracy of (66.564%), best recall of (79.768%),

Table 2: The performance results of the recurrent convolutional network based on the accuracy, precision, recall, and F1 measure.

Epochs	Word embedding structure	Embedding dimension	Accuracy	Recall	Precision	F1-score
25	word2vec(cbow)	100	52.760	68.072	52.803	59.473
50	word2vec(cbow)	100	58.282	70.700	55.223	62.011
25	Aravec(N_grams & cbow)	100	64.110	65.142	67.058	66.086
50	Aravec(N_grams & cbow)	100	60.429	64.935	57.124	60.790
25	Aravec(N_grams & SG)	100	65.337	59.523	**68.965**	63.897
50	Aravec(N_grams & SG)	100	**66.564**	**79.768**	65.094	**71.688**

best precision of (68.965%), and best F1 measure of (71.688%). Even that Word2Vec achieved relatively close performance regarding the recall and F1 measure, but AraVec still superior.

The result of comparing the two AraVec models shows that the SG model obtained better results than the CBoW model. For instance, when the number of epochs was 25, the SG model achieved better accuracy and precision by holding percentages of (65.337%) and (68.965%), respectively. Even that they were performing slightly close. Whereas, comparing the SG and CBoW models at 50 epochs shows that the SG model outperformed the CBoW regarding all metrics. Where it achieved 66.564%, 79.768%, 68.965%, and 71.688% for the accuracy, recall, precision, and F1 measure, respectively.

5 CONCLUSION AND FUTURE WORK

Cyber hate speech is a critical serious problem not only over the Arabic region, but also worldwide. Very few studies have investigated the problem of hate speech detection over online networks. Nonetheless, much fewer targeted the Arabic language since it is highly rich complex language. This paper interpreted the hate speech detection on Twitter, where a dataset is collected and processed using NLTK library. The detection approach is a deep learning approach that takes the word embeddings features as an input. While the deep learning model is a hybrid of convolutional and LSTM networks. The results of the AraVec word embedding approach with the recurrent convolutional networks were very good and competent. This research disclosed such uncharted challenges for further exploration. For example, the need for more standard and large benchmark datasets for hate speech, the need of more comprehensive lexical resource of abusive offensive Arabic expressions, as well as, to more deeply dive into deep learning approaches, yet investigate the evolutionary optimization within the deep learning.

ACKNOWLEDGEMENTS

This research is funded by the Deanship of Scientific Research in the University of Jordan, Amman, Jordan.

REFERENCES

Abozinadah, E. A., Mbaziira, A. V., and Jones, J. (2015). Detection of abusive accounts with arabic tweets. *Int. J. Knowl. Eng.-IACSIT*, 1(2):113–119.

Al-Hassan, A. and Al-Dossari, H. (2019). Detection of hate speech in social networks: asurvey on multilingual corpus. *Computer Science & Information Technology (CS & IT)*, 9(2):83.

Alakrot, A., Murray, L., and Nikolov, N. S. (2018). Dataset construction for the detection of anti-social behaviour in online communication in arabic. *Procedia Computer Science*, 142:174–181.

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

Alrefai, M., Faris, H., and Aljarah, I. (2018). Sentiment analysis for arabic language: A brief survey of approaches and techniques. *International Journal of Advanced Science and Technology*, 119(1):13–24.

Badaro, G., Baly, R., Hajj, H., El-Hajj, W., Shaban, K. B., Habash, N., Al-Sallab, A., and Hamdi, A. (2019). A survey of opinion mining in arabic: A comprehensive system perspective covering challenges and advances in tools, resources, models, applications, and visualizations. *ACM Transactions on Asian and Low-Resource Language Information Processing (TALLIP)*, 18(3):27.

Bleiweiss, A. Lstm neural networks for transfer learning in online moderation of abuse context.

Collobert, R., Weston, J., Bottou, L., Karlen, M., Kavukcuoglu, K., and Kuksa, P. (2011). Natural language processing (almost) from scratch. *Journal of machine learning research*, 12(Aug):2493–2537.

Court, E. European court of human rights. https://www.echr.coe.int. Accessed: July, 2019.

Del Vigna12, F., Cimino23, A., Dell'Orletta, F., Petrocchi, M., and Tesconi, M. (2017). Hate me, hate me not:

Hate speech detection on facebook. *t Italian Conference on Cybersecurity (ITASEC17), Venice, Italy*.

Fauzi, M. A. and Yuniarti, A. (2018). Ensemble method for indonesian twitter hate speech detection. *Indonesian Journal of Electrical Engineering and Computer Science*, 11(1):294–299.

Gers, F. A., Schmidhuber, J., and Cummins, F. (1999). Learning to forget: Continual prediction with lstm. *IET digital library*.

Goldberg, Y. and Levy, O. (2014). word2vec explained: deriving mikolov et al.'s negative-sampling word-embedding method. *arXiv preprint arXiv:1402.3722*.

Gulli, A. and Pal, S. (2017). *Deep Learning with Keras*. Packt Publishing Ltd.

Haddad, H., Mulki, H., and Oueslati, A. (2019). T-hsab: A tunisian hate speech and abusive dataset. In *International Conference on Arabic Language Processing*, pages 251–263. Springer.

Han, J., Pei, J., and Kamber, M. (2011). *Data mining: concepts and techniques*. Elsevier.

Inc., S. The most common languages on the internet. https://www.statista.com/statistics/262946/share-of-the-most-common-languages-on-the-internet. Accessed: July, 2019.

Jaki, S. and De Smedt, T. (2018). Right-wing german hate speech on twitter: Analysis and automatic detection. *Manuscript submitted*.

Loper, E. and Bird, S. (2002). Nltk: The natural language toolkit. In *In Proceedings of the ACL Workshop on Effective Tools and Methodologies for Teaching Natural Language Processing and Computational Linguistics. Philadelphia: Association for Computational Linguistics*.

Mubarak, H., Darwish, K., and Magdy, W. (2017). Abusive language detection on arabic social media. In *Proceedings of the First Workshop on Abusive Language Online*, pages 52–56.

Mulki, H., Haddad, H., Ali, C. B., Alshabani, H., and iCompass Consulting, T. (2019). L-hsab: A levantine twitter dataset for hate speech and abusive language. In *Proceedings of the Third Workshop on Abusive Language Online*, pages 111–118.

Ortiz, G., Gómez-Adorno, H., Reyes-Magaña, J., Bel-Enguix, G., and Martinez, G. E. S. (2019). Detection of aggressive tweets in mexican spanish using multiple features with parameter optimization. In *In Proceedings of the First Workshop for Iberian Languages Evaluation Forum (IberLEF 2019), CEUR WS Proceedings*.

Pitsilis, G. K., Ramampiaro, H., and Langseth, H. (2018). Effective hate-speech detection in twitter data using recurrent neural networks. *Applied Intelligence*, 48(12):4730–4742.

Polignano, M. and Basile, P. (2018). Hansel: Italian hate speech detection through ensemble learning and deep neural networks. *EVALITA Evaluation of NLP and Speech Tools for Italian*, 12:224.

Raybaut, P. (2009). Spyder-documentation. *Available online at: pythonhosted. org*.

Rehurek, R. and Sojka, P. (2011). Gensim–python framework for vector space modelling. *NLP Centre, Faculty of Informatics, Masaryk University, Brno, Czech Republic*, 3(2).

Şahi, H., Kılıç, Y., and Sağlam, R. B. (2018). Automated detection of hate speech towards woman on twitter. In *2018 3rd International Conference on Computer Science and Engineering (UBMK)*, pages 533–536. IEEE.

Soliman, A. B., Eissa, K., and El-Beltagy, S. R. (2017). Aravec: A set of arabic word embedding models for use in arabic nlp. *Procedia Computer Science*, 117:256–265.

Watanabe, H., Bouazizi, M., and Ohtsuki, T. (2018). Hate speech on twitter: A pragmatic approach to collect hateful and offensive expressions and perform hate speech detection. *IEEE Access*, 6:13825–13835.

Wijeratne, Y. (2018). The control of hate speech on social media: Lessons from sri lanka. *CPR South*.