Bulletin of Electrical Engineering and Informatics
Vol. 13, No. 2, April 2024, pp. 1351~1361
ISSN: 2302-9285, DOI: 10.11591/eei.v13i2.6530

Enhancing Arabic offensive language detection with BERT-BiGRU model

Rajae Bensoltane, Taher Zaki
Laboratory of Innovation in Mathematics and Intelligent Systems, Faculty of Applied Sciences, Agadir, Morocco

Article Info

Article history:

Received Apr 20, 2023
Revised Sep 24, 2023
Accepted Oct 21, 2023

Keywords:

Arabic
Bidirectional encoder representations from transformers
Bidirectional gated recurrent unit
Natural language processing
Offensive language detection

ABSTRACT

With the advent of Web 2.0, various platforms and tools have been developed to allow internet users to express their opinions and thoughts on diverse topics and occurrences. Nevertheless, certain users misuse these platforms by sharing hateful and offensive speeches, which has a negative impact on the mental health of internet society. Thus, the detection of offensive language has become an active area of research in the field of natural language processing. Rapidly detecting offensive language on the internet and preventing it from spreading is of great practical significance in reducing cyberbullying and self-harm behaviors. Despite the crucial importance of this task, limited work has been done in this field for non-English languages such as Arabic. Therefore, in this paper, we aim to improve the results of Arabic offensive language detection without the need for laborious preprocessing or feature engineering work. To achieve this, we combine the bidirectional encoder representations from transformers (BERT) model model with a bidirectional gated recurrent unit (BiGRU) layer to further enhance the extracted context and semantic features. The experiments were conducted on the Arabic dataset provided by the SemEval 2020 Task 12. The evaluation results show the effectiveness of our model compared to the baseline and related work models by achieving a macro F1-score of 93.16%.

Corresponding Author:

Rajae Bensoltane
Laboratory of Innovation in Mathematics and Intelligent Systems, Faculty of Applied Sciences
Agadir, Morocco
Email: r.bensoltane@uiz.ac.ma

1. INTRODUCTION

Web 2.0 has given rise to numerous platforms and tools that allow internet users to express their viewpoints and ideas on various topics and happenings. Unfortunately, some individuals misuse these platforms to propagate hate speech and offensive content, leading to adverse impacts on the mental well-being of the online community [1], [2]. According to a 2021 pew research center survey, 41% of Americans have experienced online harassment, including offensive name-calling, intentional embarrassment, physical threats, stalking, and sexual harassment. Additionally, the cyberbullying research center reports that over 30% of teenagers in the United States have endured some form of cyberbullying, including hurtful comments, spreading rumors, and threats.

Therefore, the detection of offensive language has become an active research task in natural language processing (NLP). Offensive language can be defined as text that uses abusive slurs or derogatory terms [3]. Different forms of offensive language include hate speech, aggressive content, cyberbullying, and toxic comments. Many workshops and shared tasks have been conducted to encourage research in this field from various perspectives [4]–[7].

Despite the crucial importance of this task, limited work has been done in non-English languages, like Arabic [8], [9]. Arabic occupies the 4th position among the most commonly used languages on the internet. However, the ambiguity and informality of the written format of the Arabic text make the classification of Arabic social media content a very difficult task [10]. Additionally, the Arabic language has multiple varieties with various vocabularies and structures, which make it hard to get high classification results.

Existing studies in Arabic offensive language detetcion have mainly adopted machine learning and deep learning based approaches. Alakrot *et al*. [11] employed a support vector machine (SVM) classifier and experimented with different word-level features, N-gram features, and various pre-processing techniques to detect offensive language on Arabic dataset. They also concluded that it is not beneficial to use both stemming and N-gram features within the same machine learning process. The methodology used in Shannaq *et al*. [12] consisted of two stages of optimization. In during the initial phase, the training dataset was employed to fine-tune multiple word embedding models for the extraction of word characteristics from the ArCybC corpus. In the second stage, a hybrid approach, combining a genetic algorithm (GA) with either SVM or eXtreme gradient boosting (XGBoost), was employed to enhance the model's performance. Abuzayed and Elsayed [13] conducted a comparative analysis of 15 classical and neural learning models, using two different word representations, tf-idf, and word embeddings. The experimnetal results indicated that tf-idf representation yielded better results in classical models compared to word embeddings. However, the most effective neural learning model, a joint convolutional neural network (CNN) and long short-term memory (LSTM) architecture, outperformed all classical models.

Recently, transformer-based models, such as bidirectional encoder representations from transformers (BERT) model have enhanced the results of many tasks [14]–[18], including offensive language detection. The work of Husain and Uzuner [19] investigated the impact of various preprocessing techniques on Arabic offensive language classification. Additionally, different models were examined including traditional machine learning, ensemble machine learning, artificial neural networks, and BERT-based models. The experimental results showed that the BERT-based models achieved better results over all other models. Moreover, the findings of this study indicate that preprocessing has limited gains for BERT-based classifiers in text classification pipelines, suggesting it can be omitted. Althobaiti [20] compared BERT with conventional machine learning techniques like SVM and logistic regression for handling this task. Additionally, they explored using sentiments and textual descriptions of emojis as additional features in the dataset. The experiments demonstrated that the BERT-based model outperformed all other examined models. Another study of El-Alami *et al*. [21] proposed an effective approach for multilingual offensive language detection (MOLD) using transfer learning based on BERT. The system comprises three stages: preprocessing, text representation with BERT, and classification into offensive and non-offensive categories. To address multilingualism, they investigate methods that involve both joint-multilingual and translation-based techniques. They obtained promising results with the translation-based method using the Arabic BERT model (AraBERT) by achieving over 93% F1-score and 91% accuracy on a bilingual dataset composed of English and Arabic reviews. The authors affirmed the robustness of BERT-based models in the MOLD field.

Unlike most existing work in this field, our study aims to provide an enhanced model for Arabic offensive language detetcion without relying on tedious preprocessing or feature engineering tasks. Additionally, we investigate combining the BERT model with a bidirectional gated recurrent unit (BiGRU) layer to further improve the understanding of the context and relationships between words. Moreover, various Arabic BERT models are examined in this paper to select the most suitable one for this task. The main contributions of this study can be summarized as follows:

- We propose an enhanced model for Arabic offensive language detection without the need for hand-crafted features or external linguistic resources, like lexicons.
- We combine BERT with a BiGRU layer to enhance the extracted semantic and contextual features. As far as we know, no prior work has utilized this combination to perform the offensive language detection task in Arabic.
- Extensive experiments on the Arabic SemEval 2020 dataset show the effectiveness of the proposed model in comparison to the baseline and related work models.

The remainder of this paper is structured as follows: section 2 introduces related work to Arabic offensive language detection. The research methodology is provided in section 3. Section 4 presents the experimental setup. The experimental results are discussed in section 5. Finally, section 6 presents the conclusion and outlines directions for future research.

2. RELATED WORK

Compared to the amount of work done in English, only a few studies have been conducted on detecting offensive language in Arabic [8], [9]. One of the earlier studies in this area was conducted by Mubarak *et al.* [22]. The authors built a list of 288 Arabic obscene words and another list of 127 hashtags. They then used this list along with additional patterns to gather Arabic abusive tweets from the Twitter API in 2014. These tweets were classified into two categories: tweets that did not contain any obscene word from the list of seed words, and those that included at least one of the words in the list.

Alakrot *et al.* [11], comments from YouTube were collected and manually labeled by three annotators as either offensive or non-offensive. They trained an SVM classifier with different combinations of word-level features, N-gram features, and various pre-processing techniques. They achieved an F1-score of 82% using pre-processing applied with stemming.

Mohaouchane *et al.* [23] sought to enhance the previous results by using Word2Vec embeddings with different neural network models, including CNN, bidirectional long short-term memory (BiLTSM), and BilTSM with attention. The CNN model achieved the highest accuracy score of 87.84%, and an F1-score of 84.05% over other models.

In 2020, a shared task was conducted by the SemEval workshop [24] that targeted the offensive language detection task. It provided labeled datasets for many languages, including Arabic. The team of Alami *et al.* [25] ranked first in this competition for the Arabic language. The authors used AraBERT to encode the Arabic tweets, followed by a sigmoid layer for classification. They also examined the impact of translating the meaning of emojis on the overall performance of the proposed model. They achieved a macro F1-score of 90.17%.

Hassan *et al.* [26] attained the second rank by combining of CNN-BiLSTM, SVM, and multilingual BERT. The SVM classifier employed character n-grams, word n-grams, and word embeddings as features, whereas the CNN-BiLSTM model learned character embeddings and additionally employed pre-trained word embeddings as input. Their performance yielded a macro F1-score of 90.16%.

Wang *et al.* [27] ranked third for Arabic. They proposed a unified approach to detect the offensive language in all languages, including Arabic. To this end, they used the XML-R model, which was pre-trained to learn all the language representations together. They then fed the output of [CLS] token of the top layer of XLM-R into a fully connected layer, using the same parameter for all languages. The proposed model achieved a macro F1-score of 89.89%.

Safaya *et al.* [28] attained the fourth rank for Arabic. They combined the AraBERT model with a CNN layer to handle this task. The output of the last four hidden layers was fed into several filters and convolution layers of the CNN. Then, the output of CNN was fed into a dense layer with a sigmoid activation function for classification. They reported a macro F1-score of 89.72%.

Another shared task was conducted in 2022 [29] and was divided into three subtasks: i) identify whether a tweet is offensive or not; ii) determine whether a tweet contains hate speech or not; and iii) determine the fine-grained type of hate speech (disability, social class, race, religion, ideology, and gender).

The team of Mostafa *et al.* [30] ranked first in subtask A. Seven language models were examined in this paper. Moreover, an ensemble learning approach was used to further enhance the model performance. Besides, different loss functions were evaluated to address the data imbalance problem. The best results (macro F1-score=85.2%) were achieved using a majority voting technique between three models: i) QARiB trained using Dice loss, ii) MARBERT trained using VS loss, and iii) MARBERTv2 was trained using Focal loss + label smoothing.

AlKhamissi and Diab [31] achieved second place by proposing a multi-task learning approach to handle all three sub-tasks simultaneously. They first encoded input tweets using the fine-tuned MARBERT model, and then passed the output embedding to three task-specific classifiers. Each classifier consisted of a multilayered feedforward neural network with layer normalization. Their method achieved a macro F1-score of 84.5% in subtask A.

3. METHOD

3.1. Task description

The objective of this study is to classify every text into one of two distinct classes: offensive or non-offensive. Therefore, this objective can be approached as a binary classification problem. In pursuit of this goal, the study aims to effectively differentiate texts based on their offensive content, simplifying the task into a two-class classification scenario.

3.2. Model overview

The whole architecture of the proposed model is illustrated in Figure 1. First, a BERT layer is used to generate the vector representations of the text input, followed by a BiGRU layer to further extract context

and semantic features. A fully connected dense layer with sigmoid activation function is then used to classify the text into one possible class.

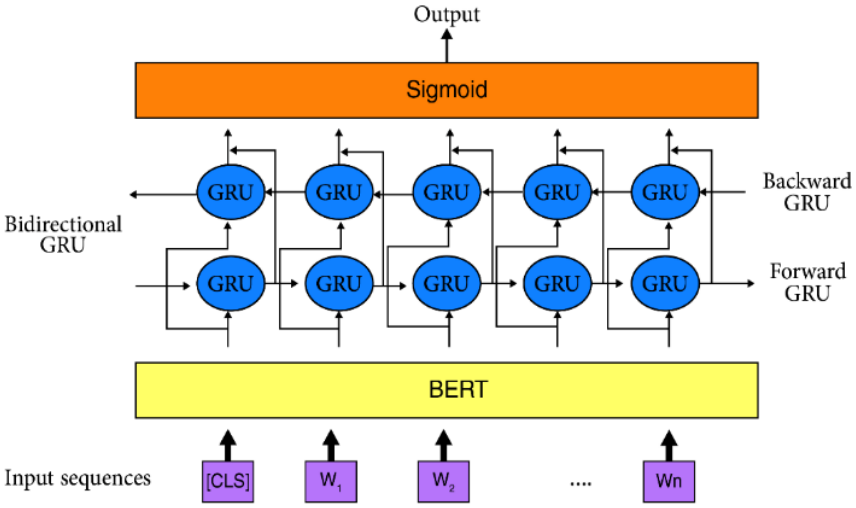

Figure 1. Overall architecture of the proposed model

3.2.1. Bidirectional encoder representations from transformers model

BERT [32] is a pre-trained language model built based on transformers, which is an attention mechanism that employs an encoder to read the input text and a decoder for generating a prediction for the task. BERT uses only the encoder part for providing a language representation model. Besides, BERT makes the training bidirectional by considering context from both left and right directions across all layers.

Moreover, the pre-training process of BERT involved two unsupervised tasks: masked language modeling (MLM) and next sentence prediction (NSP). For the first task, BERT randomly masks a portion of the input tokens and subsequently attempts to predict those hidden tokens. The second task allows the model to predict whether a sentence is the next sentence in a given sequence of sentences. The BERT model has improved the results of many NLP tasks including named entity recognition [33], [34] text classification [35], [36] and sentiment analysis [37], [38]. Figure 2 illustrates the architecture of the BERT model.

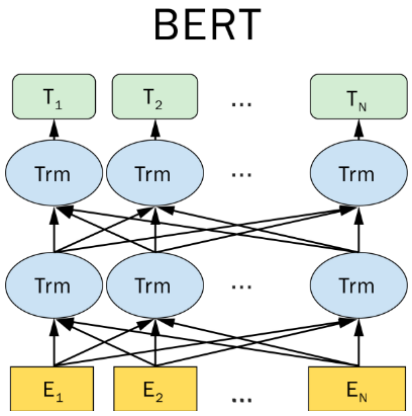

Figure 2. The architecture of BERT model [32]

There are two types of BERT-based models for the Arabic language: monolingual models and multilingual models. The first type pre-trained the BERT architecture on Arabic content only. This content can be written in classical Arabic, modern standard Arabic (MSA), or dialectical Arabic (DA). For the second type, the BERT model is pre-trained on multilingual content, including Arabic. Table 1 describes the main Arabic BERT models that are publicly available for the researchers' community.

Table 1. Main publicaly available Arabic BERT models

Model	Type of Arabic	Source	Size of pre-trained dataset
AraBERTv02 [39]	MSA	Various Arabic corpora like El-Khair [40] and OSIAN [41]	8.6B tokens
AraBERTv02-twitter	MSA+DA	The same dataset as AraBERTv02 in addition to Arabic tweets	8.6B tokens+16M tweets
MARBERTv2 [42]	MSA+DA	MSA corpora such as: OSCAR [43] and OSIAN [41] in addition to Arabic Tweets	29B tokens
Qarib [44]	MSA+DA	news and movie/TV subtitles, while the dialectical text includes tweets	14B tokens
CamelBERT [45]	DA	A range of dialectical corpora like NADI [46] and QADI [47]	5.6B tokens
mBERT [32]	MSA	Wikipedia	7292 tokens

In this study, the BERT model is fine-tuned to learn specific knowledge relevant to the downstream task. Additionally, we employed the final hidden state vector of the special token [CLS] as the representation of the entire input sequence. The output of the BERT model can be represented as (1):

$$x = H_{[CLS]} \in \mathrm{R}^{\mathrm{d}} \quad (1)$$

Where the value of the dimension d is equal to 768.

3.2.2. Bidirectional gated recurrent unit layer

GRU is a variant of the recurrent neural network (RNN) that was created to tackle the issue of long-term dependencies and the gradient vanishing problem. Its structure is simpler than LSTM, as it combines the input and forget gates into a single update gate and merges the hidden and cell states into a single hidden state, as depicted in Figure 3. The update gate, denoted as z_t, regulates the volume of past information that should be transmitted to the next state. On the other hand, the reset gate, indicated as r_t, controls the amount of previous information that should be disregarded. The calculation formula is provided as (2)-(5):

$$z_t = \sigma(W_{zx}x_t + U_{zh}h_{t-1}) \quad (2)$$

$$r_t = \sigma(W_{rx}x_t + U_{rh}h_{t-1}) \quad (3)$$

$$o_t = tanh(W_{cx}x_t + r_t \odot U_{ch}h_{t-1}) \quad (4)$$

$$g_t = (1 - z_t) \odot o_t + z_t \odot h_{t-1}) \quad (5)$$

Where σ represents the sigmoid function and $\odot$ denotes the matrix's element-wise product. The weight matrices W and U must be learned. Since GRU networks can only handle sequences from front to back, we employed a BiGRU layer to process the data from both directions and generate complete contextual features. The output of the hidden layer g_t at time t is the concatneation the backward and forward states as (6):

$$g_t = [\overrightarrow{g_t} \oplus \overleftarrow{g_t}] \quad (6)$$

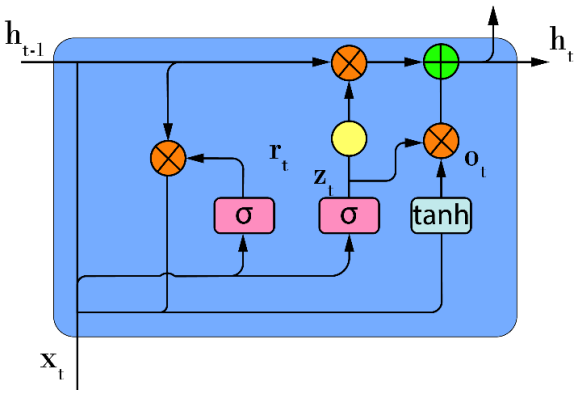

Figure 3. The architecture of GRU

The output of BiGRU can be represented as (7):

$$g = \{g_1, g_2, \dots, g_n\} \quad (7)$$

We then use a fully connected dense layer with a sigmoid function to generate the final predictions, which classify the input text as offensive or not offensive.

$$\hat{y} = Sigmoid(Vg + b) \quad (8)$$

Here, $\hat{y}$ represents the predicted probabilities, V is a weight matrix that can be adjusted during training, and b is a bias term.

4. EXPERIMENTS

4.1. Dataset

The dataset used in this study was released by SemEval 2020 task 12 [24]. It comprises 10,000 tweets gathered during the period of April to May 2019 using the Twitter API and annotated manually as either offensive or non-offensive. More details about the dataset can be found in Mubarak *et al.* [48]. Table 2 illustrates the distribution of the data in terms of training and testing sets.

Table 2. The distribution size of the dataset

Train			Test		
Off	Not	Total	Off	Not	Total
1589	6411	8000	402	1598	2000

4.2. Experimental settings

The proposed model was implemented in Python using TensorFlow and Keras libraries. For the BERT model, we used the base version, containing 12 layers of transformers with 12 self-attention heads and a hidden size of 768. Additionally, we used a max sequence length of 128, a batch size of 32, and a number of epochs of 5.

4.3. Evaluation metrics

To compare our model with the baseline and related work models, we used the accuracy and macro F1-score metrics, computed using as (9):

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \quad (9)$$

Where TP, FP, and FN denote the number of true positives, false positives, and false negatives, respectively.

$$Macro\ F1 = \frac{2 \times MP \times MR}{MP+MR} \quad (10)$$

$$MP = \frac{1}{C}\sum_j^C MP_j \quad (11)$$

$$MR = \frac{1}{C}\sum_j^C MR_j \quad (12)$$

Where C denotes the number of classes. MP_j, MR_j are the precision and recall for class j, respectively.

4.4. Baseline and related work models

The proposed model is compared with the following baseline models:

- Majority baseline [24]: it is the baseline model provided by the SemEval task for the Arabic dataset.
- BERT: we remove the BiGRU layer and fine-tuned the BERT model with a linear layer and a sigmoid activation function.
- BERT-BiLSTM: we replaced the BiGRU layer with BiLSTM in the proposed model to examine its impact on the overall performance.

In addition to these baselines, the results of related work models, that were discussed in section 2, are also included, namely AraBERTEmojisOUT [25], SVM and ValenceList + C-LSTM + Mult-BERT [26], XLM-R [15], and BERT-CNN Kuisal [28].

5. RESULTS AND DISCUSSION

5.1. Selection of the BERT model

There are many BERT-based models that have been implemented to support research in the Arabic language, as illustrated in Table 1. Thus, we first conducted various experiments to select the best BERT model for our proposed system. The experimental results are depicted in Table 3.

It can be noticed that mBERT achieved the worst results, which can be explained by the fact that this model was pre-trained on much less amount of Arabic datasets compared to the monolingual models. Among MARBERTv2, AraBERTv02, AraBERTv02-twitter, and Qarib, MARBERTv2 yielded the best results, likely attributed to the extensive pre-trained dataset (refer to Table 1). Furthermore, the dataset is a combination of MSA and DA tweets, which aligns with the evaluated data in this study. Therefore, we use MARBERTv2 to implement the evaluated models in this paper.

Table 3. Results of our proposed model using different Arabic BERT models

BERT model	Accuracy (%)	Macro F1-score (%)
MBERT	91.5	85.18
CamelBERT	93.4	89.36
Qarib	94.7	91.39
AraBERTv02	94.30	90.83
AraBERTv02-twitter	94.55	91.26
MARBERTv2	95.55	93.16

5.2. Effect of hyper-parameters

To determine the optimal hyper-parameters for our proposed model, we conducted a sensitivity analysis by testing different configurations. We started by tuning the learning rate, which is crucial for weight control during back-propagation and affects training time until convergence. A high initial learning rate can cause unstable learning and divergence, while a low learning rate can result in slow convergence. We illustrated the results of testing different learning rates on the proposed model in Figure 4 and found that a learning rate of 5e-5 produced the best performance. Higher or lower learning rates reduced performance, so we used this learning rate for all implemented models in this study.

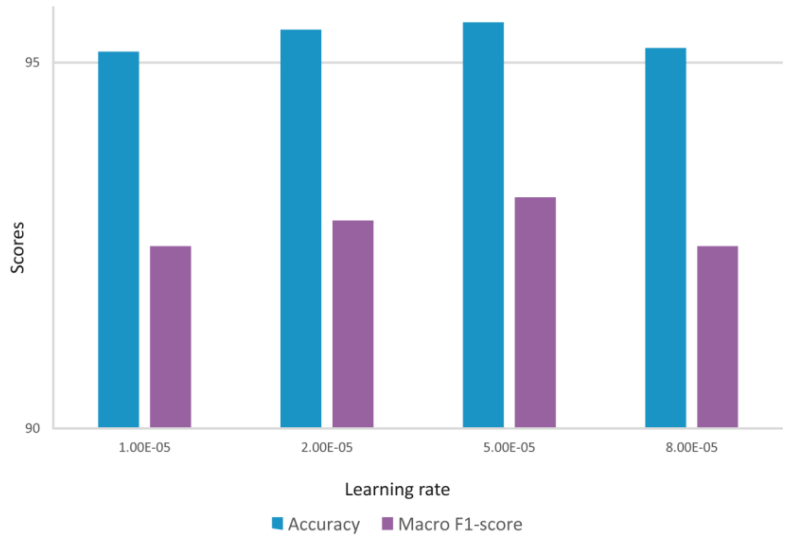

Figure 4. Experimental results using different learning rate values

The second hyper-parameter we optimized was the optimizer. We evaluated our model using different optimization methods, including Adam, Adamax, RMSProp, and SGD, as shown in Figure 5. The evaluation results showed that SGD performed poorly with a macro F1-score less than of 46%, whereas Adam and RMSProp achieved comparable results. Meanwhile, the Adamax optimizer outperformed the other methods in terms of macro F1-score. Therefore, we used it to implement our proposed model.

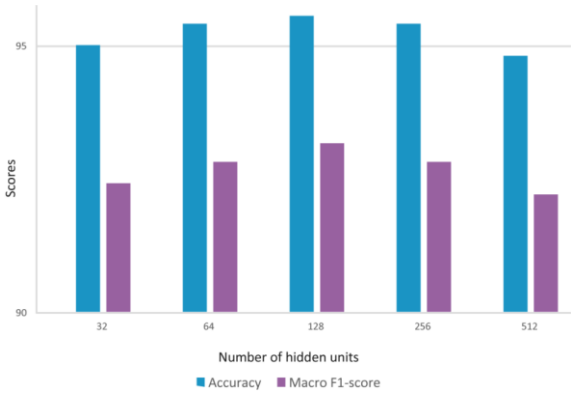

Figure 5. Experimental results using different hidden units' values

Another crucial parameter in network design is the number of hidden units in the GRU layer, which significantly impacts the model's training duration and complexity. Therefore, optimizing this parameter is essential to reduce model complexity and improve its execution performance and predictive capability. We illustrated the results of the sensitivity analysis for this parameter in Figure 6 and found that using 32 or 64 hidden units yielded comparable results, while the best value was achieved when using a number of hidden units of 128. However, when this number increased to 256 and 512, the overall performance decreased. Thus, we set the number of hidden units in the GRU layer to 128 based on the best value of the macro F1-score.

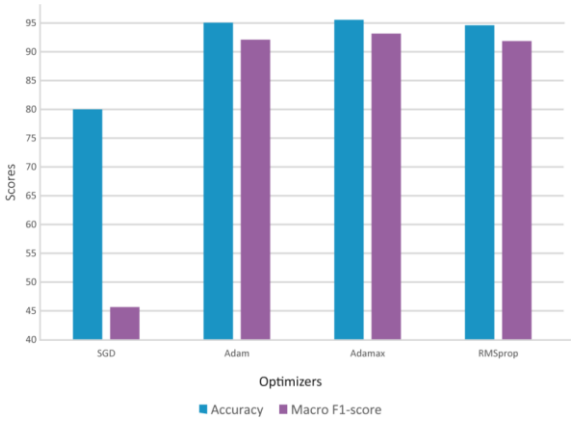

Figure 6. Experimental results using different optimizers

5.3. Comparative analysis

The main experimental results are presented in Table 4. They indicate that the proposed model achieved an overall enhancement of more than 25% in terms of F-1 score compared to the baseline model. Moreover, our model outperforms related work models that used extra features with the BERT model (i.e., AraBERTEmojisOUT) or adopted an ensemble structure (i.e., SVM and ValenceList + C-LSTM + Mult-BERT) to resolve this task using the same dataset as in this study. Additionally, the MARBERTv2-BiLSTM and MARBERTv2-BiGRU achieved better results than the MARBERTv2 model, which was fine-tuned using a linear layer only. This indicates the effectiveness of incorporating the fine-tuned MARBERTv2 model with more powerful neural network layers to further enhance the extracted semantic and contextual features.

Furthermore, our model significantly outeperfoms the BERT-CNN model, which can be justified by the fact that capturing long-range dependencies in the data and considering the order of the words are crucial for understanding the context and improving the classification performance. Besides, our model achieves better results than BERT-GRU, which indicates the effectiveness of using bidirectional layers to encode features from both left and right sides for handling this task. In addition, our model outperforms the BERT-BiLSTM model. This can be explained by the fact that GRU has a simpler architecture than LSTM, potentially simplifying the training process.

Table 4. Main evaluation results. The results with "†" were retrieved from original papers

Model	Accuracy	Macro F1-score (%)
Majority Baseline [24]	-	44.41†
AraBERTEmojisOUT [25]	93.9†	90.17†
SVM and ValenceList + C-LSTM + Mult-BERT [26]	93.85†	90.16†
XLM-R [15]	-	89.89†
BERT-CNN Kuisal [28]	-	89.7†
MARBERTv2-linear	94.55	91.17
MARBERTv2-GRU	94.55	91.66
MARBERTv2-BiSLTM	95.0	92.41
MARBERTv2-BiGRU (ours)	95.55	93.16

6. CONCLUSION

In this paper, an enhanced BERT-based model is proposed to address the offensive language detection task on an Arabic reference dataset. The proposed model employs BERT to generate contextualized vector representations, followed by a BiGRU layer to further improve the extracted context and semantic features. The experimental results showed the effectiveness of our model compared to the baseline and related work models by achieving a macro F1-score of 93.16%. Additionally, the obtained results prove the effeciency of combining BERT with bidirectional sequential layers to further improve its semantic understanding.

Future work directions include evaluating our model on other Arabic NLP tasks, such as fine-grained hate speech detection. Additionally, we intend to implement our model using other pre-trained language models than BERT, such as the XLNET model. Moreover, we plan to adapt our model to handle the task offensive language detection on multilingual corpora. In addition, the dataset used in this study is imbalanced. Thus, future work direction includes investigating various methods to handle the class imbalance issue and examining their impact on the overall performance of our model.

REFERENCES

[1] M. T. Ahmed, M. Rahman, S. Nur, A. Z. M. T. Islam, and D. Das, "Natural language processing and machine learning based cyberbullying detection for Bangla and Romanized Bangla texts," *Telkomnika (Telecommunication Computing Electronics and Control)*, vol. 20, no. 1, pp. 89–97, 2022, doi: 10.12928/TELKOMNIKA.v20i1.18630.

[2] L. Mookdarsanit and P. Mookdarsanit, "Combating the hate speech in Thai textual memes," *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 21, no. 3, pp. 1493–1502, 2021, doi: 10.11591/ijeecs.v21.i3.pp1493-1502.

[3] M. Subramanian *et al.*, "Offensive language detection in Tamil YouTube comments by adapters and cross-domain knowledge transfer," *Computer Speech and Language*, vol. 76, 2022, doi: 10.1016/j.csl.2022.101404.

[4] S. M. R. Kumar, A. K. Ojha, and M. Zampieri, "Proceedings of the First Workshop on Trolling, Aggression and Cyberbullying (TRAC-2018)," *Proceedings of the First Workshop on Trolling, Aggression and Cyberbullying (TRAC-2018)*, 2018.

[5] Z. Waseem, W. H. K. Chung, and D. Hovy, "Proceedings of the First Workshop on Abusive Language Online," *Proceedings of the First Workshop on Abusive Language Online*, 2017, [Online].

[6] D. Fišer, R. Huang, V. Prabhakaran, R. Voigt, Z. Waseem, and J. Wernimont, "Proceedings of the First Workshop on Abusive Language Online," *Proceedings of the 2nd Workshop on Abusive Language Online (ALW2)*, 2018

[7] M. Zampieri, S. Malmasi, P. Nakov, S. Rosenthal, N. Farra, and R. Kumar, "SemEval-2019 task 6: Identifying and categorizing offensive language in social media (OffensEval)," *NAACL HLT 2019 - International Workshop on Semantic Evaluation, SemEval 2019, Proceedings of the 13th Workshop*, pp. 75–86, 2019, doi: 10.18653/v1/s19-2010.

[8] B. Haddad, Z. Orabe, A. Al-Abood, and N. Ghneim, "Arabic Offensive Language Detection with Attention-based Deep Neural Networks," *Lrec*, no. May, pp. 11–16, 2020, [Online]. Available: https://www.dictionary.com/browse/hate-speech

[9] T. Kanan, G. G. Kanaan, R. Al-Shalabi, and A. Aldaaja, "Offensive language detection in social networks for arabic language using clustering techniques," *International Journal of Advances in Soft Computing and its Applications*, vol. 13, no. 2, pp. 95–111, 2021.

[10] R. Bensoltane and T. Zaki, "Aspect-based sentiment analysis: an overview in the use of Arabic language," *Artificial Intelligence Review*, vol. 56, no. 3, pp. 2325–2363, Mar. 2023, doi: 10.1007/s10462-022-10215-3.

[11] A. Alakrot, L. Murray, and N. S. Nikolov, "Towards Accurate Detection of Offensive Language in Online Communication in Arabic," *Procedia Computer Science*, vol. 142, pp. 315–320, 2018, doi: 10.1016/j.procs.2018.10.491.

[12] F. Shannaq, B. Hammo, H. Faris, and P. A. Castillo-Valdivieso, "Offensive Language Detection in Arabic Social Networks Using Evolutionary-Based Classifiers Learned From Fine-Tuned Embeddings," *IEEE Access*, vol. 10, pp. 75018–75039, 2022, doi: 10.1109/ACCESS.2022.3190960.

[13] A. Abuzayed and T. Elsayed, "Quick and Simple Approach for Detecting Hate Speech in Arabic Tweets," *Proceedings of the 4th Workshop on Open-Source Arabic Corpora and Processing Tools, with a Shared Task on Offensive Language Detection*, no. May, pp. 109–114, 2020, [Online]. Available: https://www.aclweb.org/anthology/2020.osact-1.18

[14] M. N. Fakhruzzaman, S. Z. Jannah, R. A. Ningrum, and I. Fahmiyah, "Flagging clickbait in Indonesian online news websites using fine-tuned transformers," *International Journal of Electrical and Computer Engineering*, vol. 13, no. 3, pp. 2921–2930, 2023, doi: 10.11591/ijece.v13i3.pp2921-2930.

[15] M. Boukabous and M. Azizi, "Crime prediction using a hybrid sentiment analysis approach based on the bidirectional encoder representations from transformers," *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 25, no. 2, pp. 1131–1139, 2022, doi: 10.11591/ijeecs.v25.i2.pp1131-1139.

[16] A. Pardamean and H. F. Pardede, "Tuned bidirectional encoder representations from transformers for fake news detection," *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 22, no. 3, pp. 1667–1671, 2021, doi:

10.11591/ijeecs.v22.i3.pp1667-1671.
[17] O. Hourrane and E. H. Benlahmar, "Graph transformer for cross-lingual plagiarism detection," *IAES International Journal of Artificial Intelligence*, vol. 11, no. 3, pp. 905–915, 2022, doi: 10.11591/ijai.v11.i3.pp905-915.
[18] T. Ma, Q. Pan, H. Rong, Y. Qian, Y. Tian, and N. Al-Nabhan, "T-BERTSum: Topic-Aware Text Summarization Based on BERT," *IEEE Transactions on Computational Social Systems*, vol. 9, no. 3, pp. 879–890, 2022, doi: 10.1109/TCSS.2021.3088506.
[19] F. Husain and O. Uzuner, "Investigating the Effect of Preprocessing Arabic Text on Offensive Language and Hate Speech Detection," *ACM Transactions on Asian and Low-Resource Language Information Processing*, vol. 21, no. 4, 2022, doi: 10.1145/3501398.
[20] M. J. Althobaiti, "BERT-based Approach to Arabic Hate Speech and Offensive Language Detection in Twitter: Exploiting Emojis and Sentiment Analysis," *International Journal of Advanced Computer Science and Applications*, vol. 13, no. 5, pp. 972–980, 2022, doi: 10.14569/IJACSA.2022.01305109.
[21] F. Z. El-Alami, S. O. El Alaoui, and N. En Nahnahi, "A multilingual offensive language detection method based on transfer learning from transformer fine-tuning model," *Journal of King Saud University - Computer and Information Sciences*, vol. 34, no. 8, pp. 6048–6056, 2022, doi: 10.1016/j.jksuci.2021.07.013.
[22] H. Mubarak, K. Darwish, and W. Magdy, "Abusive language detection on Arabic social media," *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, pp. 52–56, 2017, doi: 10.18653/v1/w17-3008.
[23] H. Mohaouchane, A. Mourhir, and N. S. Nikolov, "Detecting Offensive Language on Arabic Social Media Using Deep Learning," *2019 6th International Conference on Social Networks Analysis, Management and Security, SNAMS 2019*, pp. 466–471, 2019, doi: 10.1109/SNAMS.2019.8931839.
[24] M. Zampieri *et al.*, "SemEval-2020 Task 12: Multilingual Offensive Language Identification in Social Media (OffensEval 2020)," *14th International Workshops on Semantic Evaluation, SemEval 2020 - co-located 28th International Conference on Computational Linguistics, COLING 2020, Proceedings*, pp. 1425–1447, 2020, doi: 10.18653/v1/2020.semeval-1.188.
[25] H. Alami, S. O. El Alaoui, A. Benlahbib, and N. En-Nahnahi, "LISAC FSDM-USMBA Team at SemEval-2020 Task 12: Overcoming AraBERT's pretrain-finetune discrepancy for Arabic offensive language identification," *14th International Workshops on Semantic Evaluation, SemEval 2020 - co-located 28th International Conference on Computational Linguistics, COLING 2020, Proceedings*, pp. 2080–2085, 2020, doi: 10.18653/v1/2020.semeval-1.275.
[26] S. Hassan, Y. Samih, H. Mubarak, and A. Abdelali, "ALT at SemEval-2020 Task 12: Arabic and English Offensive Language Identification in Social Media," *14th International Workshops on Semantic Evaluation, SemEval 2020 - co-located 28th International Conference on Computational Linguistics, COLING 2020, Proceedings*, pp. 1891–1897, 2020, doi: 10.18653/v1/2020.semeval-1.249.
[27] S. Wang, J. Liu, X. Ouyang, and Y. Sun, "Galileo at SemEval-2020 Task 12: Multi-lingual Learning for Offensive Language Identification using Pre-trained Language Models," *14th International Workshops on Semantic Evaluation, SemEval 2020 - co-located 28th International Conference on Computational Linguistics, COLING 2020, Proceedings*, pp. 1448–1455, 2020, doi: 10.18653/v1/2020.semeval-1.189.
[28] A. Safaya, M. Abdullatif, and D. Yuret, "KUISAIL at SemEval-2020 Task 12: BERT-CNN for Offensive Speech Identification in Social Media," *14th International Workshops on Semantic Evaluation, SemEval 2020 - co-located 28th International Conference on Computational Linguistics, COLING 2020, Proceedings*, pp. 2054–2059, 2020, doi: 10.18653/v1/2020.semeval-1.271.
[29] H. Mubarak, H. Al-Khalifa, and A. M. Al-Thubaity, "Overview of OSACT5 Shared Task on Arabic Offensive Language and Hate Speech Detection," *5th Workshop Open-Source Arabic Corpora and Processing Tools with Shared Tasks on Qur'an QA and Fine-Grained Hate Speech Detection, OSACT 2022 - Proceedings at Language Resources and Evaluation Conference, LREC 2022*, pp. 162–166, 2022.
[30] A. Mostafa, O. Mohamed, and A. Ashraf, "GOF at Arabic Hate Speech 2022: Breaking The Loss Function Convention For Data-Imbalanced Arabic Offensive Text Detection," *5th Workshop Open-Source Arabic Corpora and Processing Tools with Shared Tasks on Qur'an QA and Fine-Grained Hate Speech Detection, OSACT 2022 - Proceedings at Language Resources and Evaluation Conference, LREC 2022*, pp. 167–175, 2022.
[31] B. AlKhamissi and M. Diab, "Meta AI at Arabic Hate Speech 2022: MultiTask Learning with Self-Correction for Hate Speech Classification," *5th Workshop Open-Source Arabic Corpora and Processing Tools with Shared Tasks on Qur'an QA and Fine-Grained Hate Speech Detection, OSACT 2022 - Proceedings at Language Resources and Evaluation Conference, LREC 2022*, pp. 186–193, 2022.
[32] J. Devlin, M. W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," *NAACL HLT 2019 - 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies - Proceedings of the Conference*, vol. 1, pp. 4171–4186, 2019.
[33] N. Alsaaran and M. Alrabiah, "Classical Arabic Named Entity Recognition Using Variant Deep Neural Network Architectures and BERT," *IEEE Access*, vol. 9, pp. 91537–91547, 2021, doi: 10.1109/ACCESS.2021.3092261.
[34] N. Boudjellal *et al.*, "ABioNER: A BERT-Based Model for Arabic Biomedical Named-Entity Recognition," *Complexity*, vol. 2021, 2021, doi: 10.1155/2021/6633213.
[35] A. S. Alammary, "BERT Models for Arabic Text Classification: A Systematic Review," *Applied Sciences (Switzerland)*, vol. 12, no. 11, 2022, doi: 10.3390/app12115720.
[36] R. Qasim, W. H. Bangyal, M. A. Alqarni, and A. Ali Almazroi, "A Fine-Tuned BERT-Based Transfer Learning Approach for Text Classification," *Journal of Healthcare Engineering*, vol. 2022, pp. 1–17, Jan. 2022, doi: 10.1155/2022/3498123.
[37] Z. Gao, A. Feng, X. Song, and X. Wu, "Target-dependent sentiment classification with BERT," *IEEE Access*, vol. 7, pp. 154290–154299, 2019, doi: 10.1109/ACCESS.2019.2946594.
[38] R. Bensoltane and T. Zaki, "Combining BERT with TCN-BiGRU for enhancing Arabic aspect category detection," *Journal of Intelligent and Fuzzy Systems*, vol. 44, no. 3, pp. 4123–4136, 2023, doi: 10.3233/JIFS-221214.
[39] W. Antoun, F. Baly, and H. Hajj, "AraBERT: Transformer-based Model for Arabic Language Understanding," *arXiv*, 2020, doi: 10.48550/arXiv.2003.00104.
[40] I. A. El-Khair, "1.5 billion words Arabic Corpus," *arXiv*, 2016, doi: 10.48550/arXiv.1611.04033.
[41] I. Zeroual, D. Goldhahn, T. Eckart, and A. Lakhouaja, "OSIAN: Open source international arabic news corpus - Preparation and integration into the clarin-infrastructure," *ACL 2019 - 4th Arabic Natural Language Processing Workshop, WANLP 2019 - Proceedings of the Workshop*, pp. 175–182, 2019, doi: 10.18653/v1/w19-4619.
[42] M. Abdul-Mageed, A. R. Elmadany, and E. M. B. Nagoudi, "ARBERT & MARBERT: Deep bidirectional transformers for Arabic," *ACL-IJCNLP 2021 - 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, Proceedings of the Conference*, pp. 7088–7105, 2021, doi:

10.18653/v1/2021.acl-long.551.
[43] P. J. O. Suárez, B. Sagot, and L. Romaty, "Asynchronous pipelines for processing huge corpora on medium to low resource infrastructures," *Leibniz-Institut für Deutsche Sprache*, 2019, doi: 10.14618/IDS-PUB-9021.
[44] A. Abdelali, S. Hassan, H. Mubarak, K. Darwish, and Y. Samih, "Pre-Training BERT on Arabic Tweets: Practical Considerations," 2021, [Online]. Available: http://arxiv.org/abs/2102.10684
[45] G. Inoue, B. Alhafni, N. Baimukan, H. Bouamor, and N. Habash, "The Interplay of Variant, Size, and Task Type in Arabic Pre-trained Language Models," *WANLP 2021 - 6th Arabic Natural Language Processing Workshop, Proceedings of the Workshop*, pp. 92–104, 2021.
[46] M. Abdul-Mageed, C. Zhang, H. Bouamor, and N. Habash, "NADI 2020: The First Nuanced Arabic Dialect Identification Shared Task," *arXiv*, 2020, doi: 10.48550/arXiv.2010.11334.
[47] A. Abdelali, H. Mubarak, Y. Samih, S. Hassan, and K. Darwish, "Arabic Dialect Identification in the Wild," *arXiv*, May 2020, doi: 10.48550/arXiv.2005.06557.
[48] H. Mubarak, A. Rashed, K. Darwish, Y. Samih, and A. Abdelali, "Arabic Offensive Language on Twitter: Analysis and Experiments," *WANLP 2021 - 6th Arabic Natural Language Processing Workshop, Proceedings of the Workshop*, pp. 126–135, 2021, doi: Arabic offensive language on twitter: Analysis and experiments.

BIOGRAPHIES OF AUTHORS

Rajae Bensoltane received her Ph.D in Computer Science (2023) from the university of Ibn Zohr, Morocco. She is currently an assistant professor at the Faculty of Sciences in Agadir, Morocco. Her research interests include natural language processing, sentiment analysis, and information retrieval techniques. She is currently a member of Laboratory of Innovation in Mathematics and Intelligent Systems. She can be contacted at email: r.bensoltane@uiz.ac.ma.

Taher Zaki is an associate professor and Vice Dean of the Faculty of Applied Sciences at Ibn Zohr University. He received his Ph.D in Computer Science from the University of Rouen, France in 2014. He supervises several Ph.D theses in various research areas of computer science, including information retrieval, digital image processing, pattern recognition, text mining, data mining, and knowledge management. He is currently a member of Laboratory of Innovation in Mathematics and Intelligent Systems. She can be contacted at email: t.zaki@uiz.ac.ma.