The International Journal of Engineering and Information Technology

journal homepage:www.ijeit.misuratau.edu.ly

Comparative Study of Fine-tuned BERT-based Models and RNN-Based Models. Case Study: Arabic Fake News Detection

Aljamel, A.
Misurata University
a.aljamel@mu.edu.ly

Khalil, H.
Misurata University
hussein.khalil@misuratau.edu.ly

Aburawi, Y.
Misurata University
yaburawi@it.misuratau.edu.ly

Abstract—**Large Language Models (LLMs) are advanced language models with exceptional learning capabilities. Pre-trained LLMs have achieved the most significant performances in multiple NLP tasks. BERT is one of LLMs that can be easily fine-tuned with one or more additional output layer to create a state-of-the-art model for a wide range of downstream NLP tasks. There are several BERT models which are pre-trained specifically for Arabic Language. They showed high performance results. To investigate the performance of fine-tuned Arabic BERT-based on downstream Arabic NLP tasks, four Arabic BERT-based Models have been selected to be fine-tuned for Arabic Fake News Detection. These Arabic BERT-based models have been compared with five RNN-based architecture models that have been trained for the same downstream Arabic NLP tasks. Then, the DNNL hyper-parameters of those models have been tuned to fit the Arabic Fake News Detection dataset and improve their performance results. For RNN-based models, two embeddings' techniques have been applied, local dataset embeddings generator and pre-trained Arabic Word2Vec embeddings generator. At last, the performance results of the whole 14 models have been compared after a series of experiments to investigate the potential of how to enhance the accuracy and efficiency of LLMs on Arabic fake news detection. The findings indicate that Arabic Language fine-tuned BERT-based models, particularly Camel BERT Base, demonstrate promising results in accurately classifying fake news, outperforming RNN-based models with a (0.86) accuracy rate. This result highlights the significant potential of LLMs applications in Arabic NLP downstream tasks if they are processed in a high computer power infrastructure.**

Index Terms—**Natural Language Processing, Deep Neural Network Learning, Large Language Models, Transformers, Word Embeddings, Hyper-parameters.**

I. Introduction

Natural Language Processing (NLP) has emerged as a foundational pillar of contemporary technological progress. It empowers machines to comprehend, interpret, and generate human language, Facilitating effective communication between humans and machines [1]. This interdisciplinary field grows in relation to linguistics, computer science, and artificial intelligence. NLP leverages the advancement of AI techniques to process and analyze enormous textual data. The incorporation of Machine Learning (ML), Deep Neural Networks Learning (DNNL), and Large Language Models (LLMs) within NLP has fundamentally transformed the potential of language processing systems, paving the way for a new era of ground-breaking applications across diverse domains. ML algorithms including the DNNL algorithms have played a pivotal role in empowering NLP systems to learn from data and make intelligent decisions without explicit programming. DNNL models, including Recurrent Neural Networks (RNNs) and Transformer Neural Networks, have further advanced NLP by capturing complex linguistic patterns and dependencies in text data, leading to significant improvements in language understanding tasks [2]–[4].

With their utilization of extensive datasets and computational power, LLMs have emerged as a significant breakthrough in the field of NLP. Notable examples include OpenAI's GPT-3 and Google's Bidirectional Encoder Representation of Transformers (BERT), which have showcased extraordinary proficiency in comprehending human language. These models have not only established ground-breaking standards for tasks like text generation, Machine Translation, and Sentiment Analysis but have also expanded the possibilities for advancing linguistic capabilities and overcoming intricate NLP obstacles [5], [6].

In order to conduct a series of experiments in work to find out the best performance of RNN-based and LLMs, the authors of this article have decided to collect online Arabic news to be classified as true or fake news. The importance of fake news classification extends beyond academic interest; it has real-world implications for shaping public opinion, influencing political narratives, and maintaining trust in media sources. Employing RNN-based models and LLMs into fake news classification holds immense promise for enhancing the accuracy and efficiency of detection systems [7]–[9].

Received: 29 Apr. 2024; revised 07 May 2024; accepted: 15 Mar. 2024; Available Online: 08 Aug. 2024.

This work will investigate applying transfer learning techniques on fine-tuning the Arabic-language-based pre-trained BERT-based models on detecting fake news written in Arabic. Four types of pre-trained BERT for Arabic Language models will be investigated and compared with five RNN models based, with two different embeddings techniques, after a series of optimization experiments are conducted to optimize the created models. The primary aim of this article is to explore how to leverage the inherent power of LLMs, which possess comprehensive linguistic knowledge and advanced language comprehension in order to improve the classification of downstream Arabic NLP tasks.

The remaining parts of the article proceed as follows: The next section reviews the related works to this work. Section three begins by laying out the theoretical dimensions of the research which are the NLP and DNNL paradigm. The fourth section is concerned with the methodology used for this study. The fifth section presents the findings of the research by presenting the results and the focus on the discussion undertaken for the results. The sixth section gives a brief summary and critique of the findings, the recommendations and further works. The final section is for the list of references used in this article.

II. Related Works

Various studies have been published to address the problem of Arabic fake news using several techniques. One of them used the standard ML technique, and the other researchers used the DNNL technique. In recent years, several researchers have used LLMs such as BERT-based language models. Keya, et al., in [10] proposed text augmentation technique with a BERT language model to generate an augmented dataset composed of synthetic fake data. A pre-trained multilingual BERT strategy was used to enhance the text data and feed the augmented data to a fine-tuned BERT to generate embeddings. The system architecture comprises two methods, the first method is embedding and classification, while the second method is classification method uses labelled texts to derive a classification model. The BanFakeNews' dataset was used to apply the AugFake-BERT technique. The authors state that they recorded a final accuracy score of (92.45%).

Nassif, et al., in [11] have proposed a study which aims to detect fake news in Arabic domain, the study consists of two parts: first, they have constructed a large and diverse Arabic fake news dataset. Second, they have developed and evaluated transformer-based classifiers to identify fake news while utilizing eight state-of-the-art Arabic contextualized embedding models.

Alawadh, et al., in [12] have proposed a study to utilize Arabic fake news datasets by applying Arabic embeddings for ML classifiers. They enhance the performance by using mini-BERT with attention mechanisms. Holdout validation schemes were applied to both ML and mini-BERT-based DNNL classifiers. They have found that Mini-BERT-based classifiers outperformed ML classifiers. Their results show a consistent improvement with increased training data.

Another study established by Kumari in [13] which employed the BERT-based classification model to detect a fake news, specifically in predicting the domain and classification of fake news articles. The proposed model performance scores achieved are, a macro F1 score is 83.76% for Task 3A and 85.55% for Task 3B. According to them, these scores demonstrate the effectiveness in accurately identifying fake news.

A key study of Al-Yahya, et al., in [14], which has compared DNNL with Transformer-based language models by applying these models on detecting Arabic Fake news. Several Arabic Transformers have been employed, including AraBERT, AraELECTRA, AraGPT2, and Arbert. They set up a series of experiments that have been designed in word and document level embeddings for linear and deep learning models and transformer-based models. The models were applied to three datasets (ArCOV19-Rumors, AraNews, and ANS). Their experimental results show that the best performing model is a transformer-based model rather than a neural network-based solution, and several limitations and challenges have been reported, such as the use of a small dataset and repetition of tweets and unavailable tweets.

Collectively, these studies outline a critical role for LLMs such as AraBERT, AraGPT-2, AraELECTRA, and others in the field of NLP downstream tasks such as Arabic fake news detection. The performance results of these models have been improved by applying advanced techniques. They. demonstrate excellent effectiveness in analyzing and understanding texts and accurately identifying the demands of Arabic NLP downstream tasks. In this work, a series of experiments and comparisons will be conducted between RNN-based models and Arabic language BERT-based models. These experiments will emphasis on tuning DNNL hyper-parameters to demonstrate their effectiveness in detecting fake news in the Arabic language.

III. Natural Language Processing (NLP) and Deep Neural Networks Learning (DNNL)

DNNL algorithms have already made remarkable advances in NLP tasks such as Machine Translation (MT) and Speech Recognition. In fact, DNNL takes advantage of big datasets to improve their results. This fact is guaranteed by most of the researchers who presume that the predictions' accuracies of DNNL models increase with more data training. As a result, numerous complex DNNL based algorithms and models have been proposed to solve difficult NLP tasks. For example, Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs). Superior results have been achieved after the revolutionary solutions in word embeddings and vector representations [15]–[18].

A. RNN-based Models

The main idea behind RNN-based models is processing a sequential information to produce a fixed-size vector to be represented in a sequence by feeding words sequentially to a recurrent unit. These models are capable to capture the

inherent sequential nature which are presented in language, where units could be sentences, words, and characters. RNN-based models are tailored for modelling such context dependencies in language and similar sequence modelling tasks, which resulted to be a strong motivation for researchers to use RNN-based models over other Neural Networks models in NLP tasks [15].

However, simple RNN has an issue related to the vanishing gradient. This issue makes it difficult to learn and tune the parameters of the earlier layers in the network. This restriction was overcome by various different RNN-based structures such as Long Short-Term Memory (LSTM), Gated Recurrent Units (GRUs), Bidirectional LSTM and Bidirectional GRU which are the most used RNN variants in NLP applications. The main design purpose of those RNN-based models is to enhance the mechanism of simple RNN model and solve both problems of vanishing and exploding gradient. They allow the error to be back-propagated through an unlimited number of time steps [15]. The details about the model can be found in the work of Zargar in [19].

B. Transformers Neural Networks (NN) Models

Transformers NN models have revolutionized the field of NLP with their ability to handle sequential data efficiently. They allow for parallelization and capturing long-range dependencies in the text. Transformers NN utilize the attention mechanisms in conjunction with a recurrent network to avoid recurrence by relying entirely into draw global dependencies between input and output. The performance of Transformer models can reach a significant values in many NLP tasks such as translation and summarization quality [20].

Large Language Models (LLMs) are advanced language models with massive parameter sizes and exceptional learning capabilities. The core module behind these LLMs is the attention mechanism in Transformer NN that serves as the fundamental building block for language modelling tasks. Pre-trained LLMs have enabled massive advances in NLP tasks and achieved the most significant performances in multiple NLP tasks. Recently, there has been a focus on applying transfer learning by fine-tuning these pre-trained LLMs for downstream NLP tasks with a relatively small number of examples, resulting in notable performance improvement for those tasks. There are several LLMs; for instances, BERT from google, Generative Pre-trained Transformer (GPT-2, 3, 3.5 and 4) form OpenAI and Bidirectional and Auto-Regressive Transformers (BART) from Facebook. [5], [20]–[22].

C. BERT Model

BERT is a multi-layer bidirectional Transformer encoder which is based on the original implementation described in Vaswani et al. in [20]. Transformer-based BERT architecture uses bidirectional self-attention. BERT can be easily fine-tuned with just one additional output layer to create a state-of-the-art model for a wide range of downstream tasks. In BERT process, a "sequence" refers to the input token sequence to BERT, which may be a single sentence, or two sentences packed together. The key feature of BERT is that it can handle smoothly, left-to-right, or right-to-left language models because it had been pre-trained using two unsupervised tasks, masked LM and Next Sentence Prediction (NSP). Fine-tuning BERT is straightforward since the self-attention mechanism in the Transformer allows BERT to model many downstream tasks, whether they involve single text or text pairs [23], [24].

D. Arabic Language Based BERT Pre-Trained Models

Although BERT for Arabic Language still under investigation and improvement, there are already several BERT models have been produced for Arabic Language NLP tasks. According to Alammary in [25], there are many BERT models which are pre-trained specifically for Arabic Language NLP tasks, for example, AraBERT by Antoun, et. al., in [21], ARBERT and MARBERT from Abdul-Mageed, Elmadany and Nagoudi in [26], QARiB from Abdelali, et al. in [27], CAMeLBERT from Inoue, et al. in [28] and Arabic-BERT from Safaya, Abdullatif and Yuret in [29]. Alammary in [25] states that these Arabic BERT models showed high performance comparable to that of the English BERT models. Some of these models were pre-trained on both types of Arabic Languages, Modern Standard Arabic (MSA) and a Dialectical Arabic (DA) corpus, which might have improved their performance further.

Pre-trained LLMs in general are described by their number of layers or Transformer blocks, hidden units' size, and the number of self-attention heads. According to these attributes, there are several model sizes. For Arabic BERT models applied in this work, four types have been selected of Arabic BERT types. They are AraBERT Base Model, ARBERTv2 Base Model, Camel BERT Base Model and Qarib BERT Base Model. The details of these models are found in Table 1 below.

Table 1: The details of the Arabic Pre-Trained BERT-based Models Applied in this work, Where, DS: Data Size (Tokens), VN: Vocabularies Number, IS:Iteration (Steps), HD: Hidden Units, MSA: Layer Number, TP: Trainable Parameters, DTR: Data Type References, Modern Standard Arabic, AD: Arabic Dialects, CA: Classical Arabic: CA

Model Name	DS	VN	IS	HU	LN	TP	DTR
AraBERT Base	8.6B	60k	3M	768	12	≅135M	MSA, AD [21]
Camel BERT Base	12.6B	30k	1M	768	12	≅100M	MSA, AD, CA [28]
ARBERTv2 Base	27.8B	32k	4M	768	12	≅163M	MSA [26]
Qarib BERT Base	14B	64k	10M,	768	12	≅135M	Arabic X API, Arabic Corpora [27]

IV. THE METHODOLOGY

The methodology in this article involves applying transfer learning techniques by fine-tuning the pre-trained BERT-based models for Arabic language. Four types of pre-trained Arabic Language BERT-based models have been selected. These models are AraBERT Base Model, ARBERTv2 Base Model, Camel BERT Base Model and Qarib BERT Base Model. Then, the same dataset will be trained with five RNN-based models, which are Simple RNN, LSTM, GRU, Bidirectional LSTM and Bidirectional GRU. A series of optimization experiments

will be conducted to optimize the created models. The results of the fine-tuned BERT-based and RNN-based models will be compared to find-out the better model.

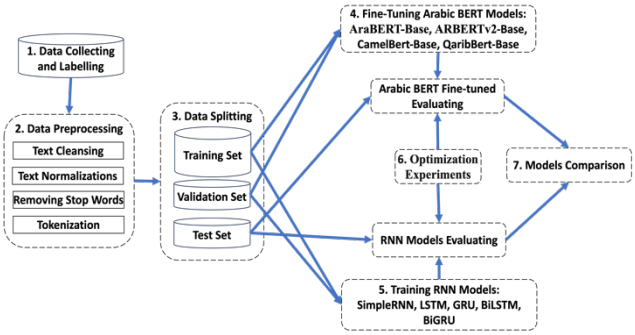

Figure 1: The Methodology

As shown in Figure 1 above, the seven stages of the methodology are, Data colleting and labelling, data pre-processing, data splitting, fine-tuning Arabic BERT-based model, RNN-based models training and evaluating, and optimization experiments, Models results comparison . The details of these stages are provided in the next sub-sections.

Stage 1: *Collecting and Labelling the Fake News Dataset.*
The pervasive spread of fake news within the digital landscape poses a significant threat to the integrity of public discourse and societal well-being. The online fake and true news classification which will be employed to conduct the optimization experiments were collected from several online news agents. The number of news samples are 1155 instances. The news is explored by media experts to label the fake and true news. The labelled Arabic news dataset will be exploited to train and optimize the BERT-based and RNN-based models to classify the news as fake or true news. The outcome of media experts labelling of the dataset Arabic news is that the news contains 650 samples of true news and 505 samples of fake news as depicted in Table 3.

Stage 2: *Pre-processing the Fake News Dataset.*
After collecting the online news, normal pre-processing was applied to clean and convert it into a coherent form to be easily handled and relevant for model training. The text normalization pre-process will reduce the dimensionality of the data which in turn accelerate the training process. These pre-processing tasks are start with filling or removing missing values in the data, removing duplicate rows, removing stop words, removing the punctuation characters, removing the English letters and words, removing any redundant characters, applying letter normalization to unify letters that appear in different forms such as replacing in {أ ، إ ، آ} with {ا}, {ة ، ـة } with { ه ، ه}, and {ى ، ئ ، ؤ} with {ء}. In addition, the operations of lemmatizing and stemming the words are applied. After completing the pre-processing stage, the specification of the pre-processed news dataset becomes as in Table 2 below.

Table 2: Dataset specifications

Counters	Before pre-processing	After pre-processing
Words Counter	21544	18290
Unique Words Counter	5033	4075

Stage 3: *Data splitting*
The pre-processed dataset has been split into three sets, the training set, validating set, and testing set. The training set has been used for training the target model, the validation set has been used for validating the model, and the testing set has been utilized for testing and evaluating the target model. In this work, the ratio splitting of these parts is: 80% for training, 10% for validation part and 10% for testing part. The summary statistics for these three datasets are presented in Table 3 below:

Table 3: Dataset splits specifications

Dataset	Samples Number	True News Samples	Fake News
Training	924	520	404
Validation	115	65	50
Testing	116	65	51
Total	1155	650	505

Stage 4: *fine-tuning Arabic BERT model and evaluating.*
For BERT models, an acquired data-based pre-processing was carried out to obtain the unique input tokens and attention masks to be ready for fine-tuning Arabic BERT-based models. Special tokens were added to each assigned token of the BERT tokenizer, <CLS>, <SEP>, <UNK>, <MASK>, <PAD>. These tokens were added at the start and end of a sentence. The maximal length was estimated, and paddings were added to shorter sentences. The attention masks were added against each created token. These contained special weights that were assigned to each token produced by the BERT tokenizer. The attention mask vector contains 0 and 1 values, in which 1 indicates the need to select a corresponding index in the embedding vector to pay attention, and 0 indicates that the corresponding keyword for attention should not be selected. This will let more attention be paid to more prominent keywords in the sentence. The pre-trained BERT model needs to add a few layers to perform fake news classification. The DNNL layers which are added on top of the pre-trained structure of each pre-trained BERT-based model are [11] (see Figure 2):

- One Dimension Global Maximum Pooling layer to optionally pooling the outputs of the embedding layer to reduce the dimensionality of the tensor which is passed to the output layer.
- Two Fully Connected layers for finding a strong correlation between the input text and the output labels. The following diagram summarizes our algorithm.
- Dropout layer to improve the model fitting resulting in increased accuracy.
- SoftMax layer with two dimension is added as an output to the whole network.

The evaluation process of the fine-tuned BERT-based models is conducted based on the training and validation phases. In the training, the loss values are in every step of the training. Loss values are an evaluation of the

effectiveness of each model's prediction. The closer the value is to zero, the more accurate the model's predictions are. In the testing phase, the Precision, Recall, F1-measure, and Accuracy values are calculated to measure the generality and performance of the models.

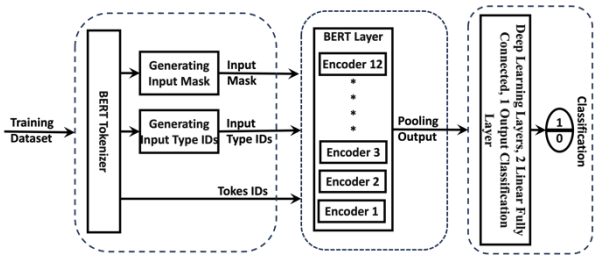

Figure 2: BERT-Based Fine-tuning Model Structure [30]

Stage 5: *RNN models training and evaluating.*

The aim of developing several RNN Models is to investigate which of them achieves the best performance, then compare it with pre-trained BERT for Arabic Language. These RNN models including, Simple RNN, LSTM, GRU, Bidirectional LSTM (BiLSTM) and Bidirectional GRU (BiGRU). The architecture of the models are shown in Figure 3 below.

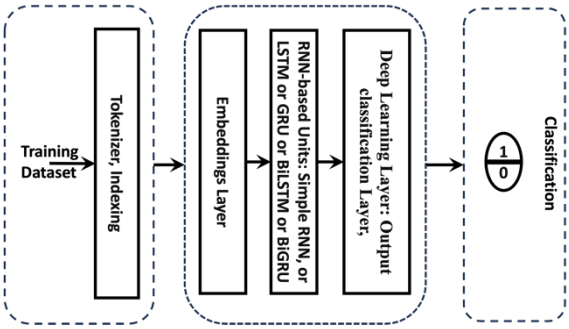

Figure 3: RNN-based Models Architecture

The architecture of the sequence classification model is many-to-one RNN type. The input gets a sequence of data as an input and generates some informatic data like labels. In this article work, after pre-processing the raw data which are sentences of Arabic news, tokenizing these sentences, submit them to the embeddings layer to generate the embeddings weights, the output of the embeddings layer is submitted to the simple RNN layer or LSTM layer or GRU layer or BiLSTM layer or BiGRU Layer. Then the output of the RNN-based models is submitted to a full connected layer with SoftMax activate function to classify the news to fake or true news.

As for the fine-tuned BERT-based models, the evaluation process for RNN-based models is conducted based on the training and validation phases. The training and validation loss values of the models at every step of the training process are measured to evaluate the effectiveness of each model's prediction. In addition, the Precision, Recall, F1 measure, and Accuracy values are measured to evaluate the performance and generality of the models.

Word Embeddings Experiments:

As part of DNNL process in NLP, words are mapped into vectors using word embedding models by using a variant of statistical and language techniques to map words into vectors of real numbers. The word embedding models attempt to capture semantic, syntactic, and contextual representation of words in the text. Word embedding can be used to calculate word similarity which can be used in many tasks such as text classification. Recently, a number of models have been proposed for contextualized word embeddings. These models run a pretrained encoder network over the sentence to produce contextual embeddings of each token. Encoders, such as RNN-based models or a Transformer-based models, can be trained on several NLP tasks for which large amounts of data are available to produce a collection of one vector per token. The output of these embedding models can be used as a drop-in replacement input to any model for any NLP task [31].

For RNN-based models, there are two embedding methods are applied for RNN-based architecture, the first is by a Neural Network training on this research's dataset. The second embeddings model is a pre-trained encoder network embeddings model, which is AraVec model. This embeddings model applies the Word2Vec techniques. For more information about this Arabic embeddings model, review the work of Soliman, Eissa and El-Beltagy in [32]. For BERT models, their build-in embeddings models are applied. In BERT models, a Word Piece tokenization is utilized where each word of the input sentence breaks down into sub-word tokens [21].

Stage 6: *Optimization Experiments.*

In this work, a series of experiments are conducted on models for tuning several hyper-parameters to optimize and obtain the better models' performance. In DNNL algorithms, there are various hyper-parameters that can be adjusted or tunned, such as the optimizers, learning rates, number of epochs and the dropout values. The values of these hyper-parameters can be determined by employing default values, or by employing recommended values or by searching for the best values. Generally, there are two common methods to find the ML algorithms' hyper-parameters optima. First, by applying the Grid Search method, second, by applying the automatic Heuristic Search techniques. There are two types of Grid Search method, a complete search or recursive backtracking search and the random grid search. In this research, the values of DNNL hyper-parameters are determined by three means, employing default values, employing the recommended values, and applying the grid-search. For grid-search approach, the random Hyper-parameters sets grid-based search is applied. These hyper-parameters values are selected manually and then the hyper-parameters grid sets of these values are selected automatically and randomly. The Hyper-parameters values in grid sets are used to configure the models to be trained and evaluated in a k-fold cross-validation process. Finally, the parameters that achieve the highest model performance are chosen. Usually, The random grid search method is

applied when the hyper-parameter search space is large [33]. In this work, we adopted random grid-based search to perform hyper-parameter tuning as it is sufficient to satisfy the requirements of the deployed ML techniques. In addition, it is simple to implement in comparison with the computationally expensive automatic optimization techniques. In this work, for RNN-based models, a python ready Grid Search packages have been applied; however, for BERT-based models, the authors of this article designed the Random Grid Search method to search the optima hyper-parameter values.

The recommended and allocated hyper-parameters values by some expertise in the area which are employed in this research, are illustrated in Table 4 below.

Table 4: Not Tuned Hyper-Parameters

#	The Hyper-Parameter	Value
1	Bach Size	32
2	Maximum Sequence length	20
3	Start monitoring the Epochs and Patience before early stop the training	250, 10
4	L1 and L2 norm regularization	L1=1e-5, L2=1e-4

However, the Hyper-parameters and their grid sets values which are tuned by using the Random Grid Search method are shown in Table 5 below. Taking into consideration that the embeddings length size hyper-parameter is only for RNN-based models with local dataset trained embeddings.

Table 5: Hyper-Parameter Values

The Hyper-Parameter	Value 1	Value 2	Value 3	Value 4
Optimizer	Adam	AdaGrad	AdaMax	RMSprop
Learning rate	0.01	1e-3	1e-4	1e-5
Dropout rate	0.2	0.25	0.4	0.5
Number of epochs	1-1000 epochs, Early Stopping with different starting from and patience epochs			
Activation Function	Relu	Tanh	--	--
Hiding Lyres Units	32	64	128	256
Embeddings Length Size	16	32	64	128

Table 6: The Models' Grid Searching Hyper-Parameters. Where ET: Embeddings Type, Opt: Optimizer, LR: Learning Rate, DR: Dropout Rate, HLU: Hiding Layers Units, AF: Activation Function, EL: Embeddings Length

The Model	ET	Opt	LR	DR	HLU	AF	EL
Simple RNN	Trained	adamax	1e-4	0.4	256	tanh	16
	W2V	adamax	1e-4	0.2	128	tanh	300
LSTM	Trained	adam	1e-5	0.4	256	tanh	128
	W2V	adam	0.001	0.25	128	tanh	300
GRU	Trained	rmsprop	0.01	0.2	32	relu	128
	W2V	adam	1e-4	0.5	64	tanh	300
BiLSTM	Trained	rmsprop	0.01	0.2	32	relu	128
	W2V	adamax	1e-4	0.5	128	tanh	300
BiGRU	Trained	rmsprop	0.01	0.5	256	tanh	32
	W2V	adam	1e-4	0.5	64	tanh	300
AraBERT Base	Bert	adamax	0.001	0.2	32	relu	786
Camel BERT Base	Bert	adamax	0.001	0.2	32	tanh	768
ARBERTv2 Base	Bert	adam	1e-4	0.2	256	tanh	768
Qarib BERT Base	Bert	adam	1e-5	0.4	64	tanh	768

The rest of the DNNL hyper-parameters' values are specified by employing the default values.

The hyper-parameter values of grid-search sets in Table 5 were heuristically selected by studying the specifications and recommendations of those algorithms. The hyper-parameters' values selected by random grid search proved favorable to the traditionally accepted default values for the DNNL models. Table 6 illustrates this work's random grid search experiments best hyper-parameters values including the three types of embeddings.

Stage 7: *Models Results and Comparison*

The hyper-parameter values in Table 6 are the results of the random grid search experiments of all DNNL models used in this work. These hyper-parameter values and the values in Table 4 have been used to configure the DNNL models to get their performance in Arabic fake news classification. Then a comparison between the performance results of these classifiers is accomplished to find-out which model is the best and why? Answering these questions will be provided in the next section with the discussion about the classifiers used in this article. conduct all the experiments of this work to optimize those models.

V. Results and Discussion

As aforementioned, the experiments of optimizing five RNN-based DNNL architectures and four BERT-based DNNL are conducted on classifying the Arabic fake news dataset. The experiments are categorized into three groups, the first experiments group is for the five RNN-based DNNL architectures. In this group, the embeddings technique is by training a local dataset to generate the embeddings. The second experiments group is also for the five RNN-based DNNL architectures; however, for these architectures, the embeddings are generated by inserting a layer of per-trained embeddings model, AraVec model. The third experiments group is by fine-tuning four types of Arabic BERT models. Figure 6 below shows the training behavior comparison between training dataset accuracy and the validation dataset accuracy of all models in terms of epochs numbers.

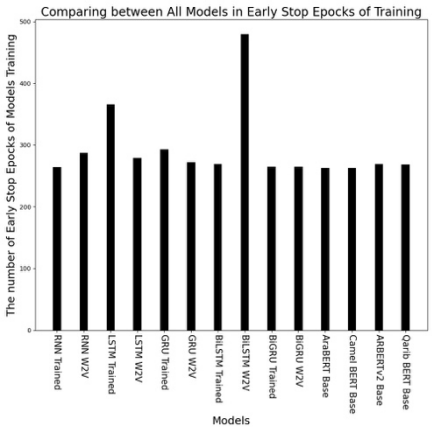

Figure 4: A compression between all models in terms of the number of early stopping epochs.

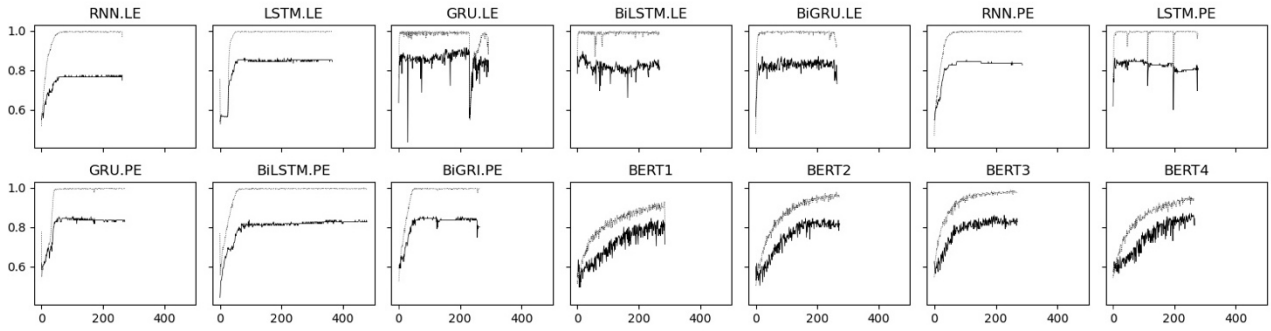

Figure 5: The training behaviors comparison between training dataset accuracy and the validation dataset accuracy of all models in terms of epochs numbers, where LE is local embeddings and PE is pre-trained embeddings and BERT1 is AraBERT Base, BERT2 is Camel BERT Base, BERT3 is ARBERTv2 Base and BERT4 is Qarib BERT Base

In addition, Figure 5 above demonstrations a comparison between early stopping epochs number of all models.

The performance results obtained from the preliminary first group experiments of RNN-based models' optimization are set out in Table 7 below.

These RNN-based models are configured by using the specified hyper-parameters values in Table 4 and Table 5 above besides the local dataset training embeddings layer.

Table 7: RNN-based Models Performance Results with local dataset embeddings generating

The Model	P	R	F1	Accuracy	Number of Epochs
RNN	0.72	0.71	0.71	0.72	264
LSTM	0.84	0.83	0.83	0.84	366
GRU	0.81	0.78	0.78	0.79	293
BiLSTM	0.81	0.78	0.78	0.79	269
BiGRU	0.72	0.71	0.69	0.69	265

For the results of the second group experiments, Table 8 below presents these performance results of RNN-based models which are configured by using the specified hyper-parameters values in Table 4 and Table 5 above besides the pre-trained embeddings layer.

Table 8: RNN-based Models Performance Results with pre-trained model embeddings generating

The Model	P	R	F1	Accuracy	Number of Epochs
Simple RNN	0.81	0.80	0.80	0.81	287
LSTM	0.84	0.80	0.81	0.82	279
GRU	0.82	0.82	0.82	0.82	272
BiLSTM	0.84	0.83	0.83	0.84	479
BiGRU	0.83	0.81	0.81	0.82	265

For the results of the third group experiments, Table 9 below presents these performance results of BERT-based models which are configured by using the specified hyper-parameters values in Table 4 and Table 5 above.

Table 9: BERT-based fine-tuned Models Performance Results

The Model	P	R	F1	Accuracy	Number of Epochs
AraBERT Base	0.79	0.77	0.77	0.78	263
Camel BERT Base	**0.87**	**0.85**	**0.86**	**0.86**	**263**
ARBERTv2 Base	0.81	0/79	0.79	0.80	269
Qarib BERT Base	**0.65**	**0.64**	**0.64**	**0.66**	**268**

Figure 6 below compares between all models in the three experiments groups in terms of testing accuracy results.

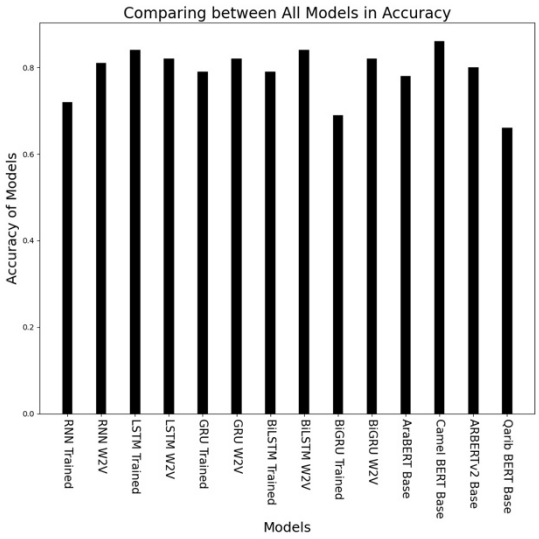

Figure 6: A comparison between all models in terms of testing dataset accuracy

Discussion

It can be seen from the data in tables and figures above that the results of most of the models are close to each other; however, Table 7, Table 8, Table 9 and Figure 6 above show that the maximum testing performance rates are for Camel BERT Base model with a precision rate of (0.87), recall rate of (0.85), f1-score rate of (0.86) and accuracy rate of (0.86). These rates are obtained after (263) epochs. In addition, the minimum performance rate are for Qarib BERT Base model with precision rate of (0.65), recall rate of (0.64), f1-score rate of (0.64) and accuracy rate of (0.66). These rates are obtained after (268) epochs. Obtaining the maximum performance rates from BERT-based models is expected; nevertheless, obtaining the minimum performance rates is somewhat counterintuitive. A possible explanation for these results may be the lack of adequate computing infrastructure for enough training BERT-based models. This lack of adequate computing infrastructure leads to selecting base or small Arabic BERT-based models rather than many other large Arabic BERT-based models. Another possible

explanation for this is that the number of iterations for random grid search cross-validation is not enough to find out the optima hyper-parameter values for BERT-based models.

From the charts in above, it can be seen that the fluctuation in accuracy values for RNN-based models is higher than the fluctuation in BERT-based models; in addition, there is a sharp rise in accuracy of RNN-based models compared to the BERT-based models. It is obvious that the training accuracy and validation accuracy in BERT-based models are gradually increasing apart from the oscillation caused by small batch size during the training. We can conclude from these models' training behaviors that increasing the epochs number of training will increase the validation accuracy results and improve the performance of fine-tuned Arabic BERT-based models, nevertheless, it can be revealed from the training behaviors of RNN-based models that the performance improvement is uncertain. This conclusion can be confirmed by the size of the training dataset which is utilized to train RNN-based models is small compared to the pre-trained BERT-based models which are pre-trained on a huge size of datasets and fine-tuned with a massive number of parameters for downstream task in this research.

Furthermore, Figure 4 above depicts a similarity in the number of early stopping epochs in the training process of all models. Most of early stopping epochs are less than (300) except the trained embeddings LSTM model which is (366) epochs and pre-trained embeddings Bidirectional LSTM model which is (479) epochs.

From the results of hyper-parameter optimization random grid search which are demonstrated in Table 6 above, the most selected hyper-parameters values can be as in Table 10 below.

Table 10: The Most Selected Hyper-Parameters by RNN-Based Models and BERT-based Models. The hyper-Parameter Embeddings Length is for the 5 local trained RNN-based models, Where Sel.: Selected Hyper-Parameters, LR: Learning Rate, AF: Activation Function, DR: Drop Rate, HLU: Hidden Layers Units, EL: Embeddings,

Opt.	Sel.	LR	Sel.	AF	SEL	DR	Sel.	HLU	Sel.	EL	Sel.
AdaMax	5	0.01	3	Relu	3	0.2	6	32	4	16	1
Adam	6	0.001	3	Tanh	11	0.25	1	64	3	32	1
RMSprop	3	1e-4	6	--	--	0.4	3	128	3	64	0
AdaGrad	0	1e-5	2	--	--	0.5	4	256	4	128	3
	14		14		14		14		14		5

The most selected hyper-parameter values set is:

{Optimizer: Adam, Learning Rates: 1e-4, Activation Function: Tanh, Dropout Rate: 0.2, Hidden Layers Units: 32 or 256, Embeddings Length: 128 local trained or 300 W2V-based or 768 BERT-based}

This set is companied with the following specified not-searched hyper-parameters set:

{Bach Size:32, Maximum Sequence length:20, Start monitoring the Epochs: 250, Patience before early stop the training: 10, L1 norm regularization: 1e-5, L2 norm regularization: 1e-4}

We believe that these optima hyper-parameter values can be utilized to optimize the models more; nevertheless, Table 10 above depicts the best hyper-parameter values for every model separately.

Several reports have shown that the NLP transfer learning by using pre-trained LLMs are worthfully to perform downstream NLP tasks. In this work we successfully fine-tuned 4 different Arabic BERT-based to perform a classification of small dataset of Arabic fake news. We believe that a reliable computing infrastructure is crucial to adequately training a fine-tuned LLMs for NLP downstream tasks.

VI. Conclusion Recommendations and Further Works

This study sets out to compare two DNNL models techniques, RNN-based models, and BERT-based models. The comparison has been performed on Arabic online news to detect the fake news. The five RNN-based models are Simple RNN, LSTM, GRU, Bidirectional LSTM and Bidirectional GRU. These language models are performed by applying two kinds of embeddings techniques, local dataset embeddings representation generator and pre-trained Arabic W2V embeddings representation. The four BERT-based models are AraBERT Base Model, ARBERTv2 Base Model, Camel BERT Base Model and Qarib BERT Base Model.

This research underscores the usefulness of employing LLMs, particularly pre-trained Arabic BERT models, in the realm of Arabic context fake news classification. The study demonstrates that fine-tuning these models through transfer learning techniques leads to notable improvements in classification accuracy compared to traditional RNN models. The superior performance of BERT models, especially Camel BERT Base Model, can be attributed to their ability to capture intricate linguistic patterns and semantic relationships within textual data, leveraging their extensive pre-training on vast amounts of Arabic text. This research emphasizes the potential of LLMs as a powerful tool in combating the spread of misinformation and fostering a more informed and reliable online environment for Arabic-speaking communities.

Through a series of hyper-parameter values optimization experiments, Camel BERT Base Model was assessed as the best performing classifier by precision rate of (0.87), recall rate of (0.85), f1-measure rate of (0.86) and accuracy rate of (0.86).

Based on the findings of this research, the following recommendations are proposed:

- These kinds of experiments require high computing power infrastructure to explore more hyper-parameters and more values for these hyper-parameters.
- We early stopped the training; we believe that increasing the epochs number of traing will increase the accuracy.
- Some of DNNL hyper-parameters do not require to be tuned, their default values are enough.
- Labelling the texts in the supervised learning should be conducted by comparing more than one expert to grantee the labelled texts of each other.

- It is better to fine-tune the LLMs which are pre-trained on different types of Arabic Languages such as the Libyan Dialectical Arabic (DA) texts.

Building on the foundation established in this research, several avenues for further work are proposed:

- Explore More LLM Models.
- Tune more Hyper-parameters for DNNL and LLM Models by building them on a high computing power infrastructure.
- Fine-tuning pre-trained Arabic LLMs for sequence-to-sequence downstream Arabic NLP tasks such a summarization and translation.
- Instead of adding full connected hidden layers to the fine-tuned LLMs, it is better try adding RNN-based layers for a more comprehensive approach to Arabic NLP downstream tasks such as fake news detection.
- Explore the practical challenges and considerations involved in deploying LLM-based NLP downstream tasks.

By following these research directions, we can further advance the field of Arabic NLP and contribute to the development of more effective and reliable solutions for combating the spread of fake news and promoting information integrity within the Arabic-speaking world.

REFERENCES

[1] R. Rocca, N. Tamagnone, S. Fekih, X. Contla, and N. Rekabsaz, 'Natural language processing for humanitarian action: Opportunities, challenges, and the path toward humanitarian NLP', Front. Big Data, vol. 6, p. 1082787, Mar. 2023.

[2] W. Khan, A. Daud, K. Khan, S. Muhammad, and R. Haq, 'Exploring the frontiers of deep learning and natural language processing: A comprehensive overview of key challenges and emerging trends', Nat. Lang. Process. J., vol. 4, p. 100026, Sep. 2023.

[3] I. H. Sarker, 'AI-Based Modeling: Techniques, Applications and Research Issues Towards Automation, Intelligent and Smart Systems', SN Comput. Sci. 2022 32, vol. 3, no. 2, pp. 1–20, Feb. 2022.

[4] S. Minaee et al., 'Large Language Models: A Survey', Feb. 2024.

[5] B. Min et al., 'Recent Advances in Natural Language Processing via Large Pre-trained Language Models: A Survey', ACM Comput. Surv., vol. 56, no. 2, Sep. 2023.

[6] M. U. Hadi et al., 'A Survey on Large Language Models: Applications, Challenges, Limitations, and Practical Usage', Authorea Prepr., Oct. 2023.

[7] A. C. Hinojosa, D. Álvaro, R. Yuste, D. Roberto, and C. Sánchez, 'Exploring the Power of Large Language Models: News Intention Detection using Adaptive Learning Prompting', 2023.

[8] Y. Yan, P. Zheng, and Y. Wang, 'Enhancing large language model capabilities for rumor detection with Knowledge-Powered Prompting', Eng. Appl. Artif. Intell., vol. 133, p. 108259, Jul. 2024.

[9] C. Trattner et al., 'Responsible media technology and AI: challenges and research directions', AI Ethics 2021 24, vol. 2, no. 4, pp. 585–594, Dec. 2021.

[10] A. J. Keya, M. A. H. Wadud, M. F. Mridha, M. Alatiyyah, and M. A. Hamid, 'AugFake-BERT: Handling Imbalance through Augmentation of Fake News Using BERT to Enhance the Performance of Fake News Classification', Appl. Sci., vol. 12, no. 17, 2022.

[11] A. B. Nassif, A. Elnagar, O. Elgendy, and Y. Afadar, 'Arabic fake news detection based on deep contextualized embedding models', Neural Comput. Appl., vol. 34, no. 18, pp. 16019–16032, 2022.

[12] H. M. Alawadh, A. Alabrah, T. Meraj, and H. T. Rauf, 'Attention-Enriched Mini-BERT Fake News Analyzer Using the Arabic Language', Futur. Internet, vol. 15, no. 2, pp. 1–14, 2023.

[13] S. Kumari, 'NoFake at CheckThat! 2021: Fake News Detection Using BERT', 2021.

[14] M. Al-Yahya, H. Al-Khalifa, H. Al-Baity, D. Alsaeed, and A. Essam, 'Arabic Fake News Detection: Comparative Study of Neural Networks and Transformer-Based Approaches', Complexity, vol. 2021, 2021.

[15] T. Young, D. Hazarika, S. Poria, and E. Cambria, 'Recent trends in deep learning based natural language processing [Review Article]', IEEE Comput. Intell. Mag., vol. 13, no. 3, pp. 55–75, 2018.

[16] Y. M. Wazery, M. E. Saleh, A. Alharbi, and A. A. Ali, 'Abstractive Arabic Text Summarization Based on Deep Learning', Comput. Intell. Neurosci., vol. 2022, pp. 1–14, 2022.

[17] A. L'Heureux, K. Grolinger, H. F. Elyamany, and M. A. M. Capretz, 'Machine Learning with Big Data: Challenges and Approaches', IEEE Access, vol. 5, pp. 7776–7797, 2017.

[18] P. Xu, X. Ji, M. Li, and W. Lu, 'Small data machine learning in materials science', npj Comput. Mater., vol. 9, no. 1, pp. 1–15, 2023.

[19] S. Ashraf Zargar, 'Introduction to Sequence Learning Models: RNN, LSTM, GRU', no. April, 2021.

[20] A. Vaswani et al., 'Attention is all you need', in 31st Conference on Neural Information Processing Systems (NIPS 2017), Long Beach, CA, USA., 2017, vol. 2017-Decem, no. Nips, pp. 5999–6009.

[21] W. Antoun, F. Baly, and H. Hajj, 'AraBERT: Transformer-based Model for Arabic Language Understanding', 2020.

[22] Y. Chang et al., 'A Survey on Evaluation of Large Language Models', ACM Trans. Intell. Syst. Technol., pp. 1–43, 2024.

[23] J. Devlin, M. W. Chang, K. Lee, and K. Toutanova, 'BERT: Pre-training of deep bidirectional transformers for language understanding', NAACL HLT 2019 - 2019 Conf. North Am. Chapter Assoc. Comput. Linguist. Hum. Lang. Technol. - Proc. Conf., vol. 1, no. Mlm, pp. 4171–4186, 2019.

[24] X. Ma, Z. Wang, P. Ng, R. Nallapati, and B. Xiang, 'Universal Text Representation from BERT: An Empirical Study', 2019.

[25] A. S. Alammary, 'BERT Models for Arabic Text Classification: A Systematic Review', Appl. Sci., vol. 12, no. 11, 2022.

[26] M. Abdul-Mageed, A. R. Elmadany, and E. M. B. Nagoudi, 'ARBERT & MARBERT: Deep bidirectional transformers for Arabic', ACL-IJCNLP 2021 - 59th Annu. Meet. Assoc. Comput. Linguist. 11th Int. Jt. Conf. Nat. Lang. Process. Proc. Conf., no. i, pp. 7088–7105, 2021.

[27] A. Abdelali, S. Hassan, H. Mubarak, K. Darwish, and Y. Samih, 'Pre-Training BERT on Arabic Tweets: Practical Considerations', 2021.

[28] 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 Arab. Nat. Lang. Process. Work. Proc. Work., pp. 92–104, 2021.

[29] A. Safaya, M. Abdullatif, and D. Yuret, 'KUISAIL at SemEval-2020 Task 12: BERT-CNN for Offensive Speech Identification in Social Media', 14th Int. Work. Semant. Eval. SemEval 2020 - co-located 28th Int. Conf. Comput. Linguist. COLING 2020, Proc., no. Ml, pp. 2054–2059, 2020.

[30] H. Chouikhi, H. Chniter, and F. Jarray, 'Arabic Sentiment Analysis Using BERT Model', Commun. Comput. Inf. Sci., vol. 1463, no. November, pp. 621–632, 2021.

[31] I. Tenney et al., 'What do you learn from context? Probing for sentence structure in contextualized word representations', 7th Int. Conf. Learn. Represent. ICLR 2019, pp. 1–17, 2019.

[32] A. B. Soliman, K. Eissa, and S. R. El-Beltagy, 'AraVec: A set of Arabic Word Embedding Models for use in Arabic NLP', Procedia Comput. Sci., vol. 117, pp. 256–265, 2017.

[33] A. Aljamel, T. Osman, G. Acampora, A. Vitiello, and Z. Zhang, 'Smart Information Retrieval: Domain Knowledge Centric Optimization Approach', IEEE Access, vol. 7, no. Ml, pp. 4167–4183, 2019.