Hindawi
Complexity
Volume 2021, Article ID 5516945, 10 pages
https://doi.org/10.1155/2021/5516945

Research Article

Arabic Fake News Detection: Comparative Study of Neural Networks and Transformer-Based Approaches

Maha Al-Yahya,[1] **Hend Al-Khalifa**,[1] **Heyam Al-Baity**,[1] **Duaa AlSaeed**,[1] **and Amr Essam**[2]

[1] *Information Technology Department, College of Computer and Information Sciences, King Saud University, P.O. Box 145111, 4545, Riyadh, Saudi Arabia*
[2] *Computer Department, College of Engineering, Helwan University, Cairo 11795, Egypt*

Correspondence should be addressed to Maha Al-Yahya; malyahya@ksu.edu.sa

Received 19 February 2021; Revised 31 March 2021; Accepted 6 April 2021; Published 19 April 2021

Academic Editor: M. Irfan Uddin

Fake news detection (FND) involves predicting the likelihood that a particular news article (news report, editorial, expose, etc.) is intentionally deceptive. Arabic FND started to receive more attention in the last decade, and many detection approaches demonstrated some ability to detect fake news on multiple datasets. However, most existing approaches do not consider recent advances in natural language processing, i.e., the use of neural networks and transformers. This paper presents a comprehensive comparative study of neural network and transformer-based language models used for Arabic FND. We examine the use of neural networks and transformer-based language models for Arabic FND and show their performance compared to each other. We also conduct an extensive analysis of the possible reasons for the difference in performance results obtained by different approaches. The results demonstrate that transformer-based models outperform the neural network-based solutions, which led to an increase in the F1 score from 0.83 (best neural network-based model, GRU) to 0.95 (best transformer-based model, QARiB), and it boosted the accuracy by 16% compared to the best in neural network-based solutions. Finally, we highlight the main gaps in Arabic FND research and suggest future research directions.

1. Introduction

Fake news or rumors are defined as "a claim or information that is verified to be not true" [1]. False information posted on social media platforms is a significant problem because it can spread rapidly and reach tens of thousands of people extremely quickly. Thus, manual methods for detecting fake news are not feasible in terms of time and cost. Therefore, to limit the spread of questionable content and alert the public of the possibility that the news they are reading is not real, methods that can automatically identify fake news are required. Moreover, with the ongoing COVID-19 pandemic, misleading or false COVID-19 information is becoming a serious problem that can impact people's health.

Fake news detection (FND) is defined as "the prediction of the chances of a particular news article (news report, editorial, expose, etc.) being intentionally deceptive" [2]. Other terms referring to tasks similar or closely related to FND include [2] rumor detection, rumor veracity classification, misleading information detection, stance classification of news articles [3], credibility assessment, fact checking "the assessment of news authenticity" [4], and claim verification. A comparison of these terms can be found in the literature [2]. Recently, FND tasks have attracted considerable interest in the NLP research community. In recent years, the use of machine learning, particularly deep learning-based methods, to identify such phenomena has attracted the attention of the research community. The first rumors evaluation shared task took place in RumourEval-2017 as part of the SemEval-2017 conference [5]. Since then, the field has attracted much attention. Two recent challenges have addressed this task, RumourEval-2019 [6] and the Constraint@AAAI2021-COVID19 FND challenge [1].

The goal of this study is to empirically analyze whether current advances in deep learning models and large-scale language models for the Arabic language can be effectively applied to the task of Arabic FND. We consider the problem of identifying fake news as a classification problem; i.e., our goal is to classify a given tweet as fake or real. In this study, the FND task can be defined as follows: "Given a tweet from Twitter on COVID-19, we would like to predict if this piece of news is fake news or real news." This study will investigate the use of deep learning and transformer-based language models for the task of Arabic FND using four available datasets ArCOV19-Rumors [7], COVID-19-Fakes [8], AraNews [9], and the Arabic News Stance corpus (ANS) [10]. We conduct a comparative study that investigates widely popular deep learning architectures and transformer-based models for the FND task. We hope to provide the research community with insights to help better understand the behavior of these models when applied to COVID-19 Arabic news.

The remainder of this paper is organized as follows. Section 2 presents work related to FND. In Section 3, we describe our approach and present details of the experiments performed in this study. Analysis and discussion of the experimental results are presented in Section 4. Conclusions and suggestions for future work are provided in Section 5.

2. Related Work

Some studies consider rumor detection as a rumor resolution task with a pipeline that includes several components, such as rumor detection, rumor tracking, and stance classification, that contribute to determining the veracity classification of a rumor [6]. FND methods may vary depending on the data they target (short social media data, such as tweets and posts, or long website articles), and the ML method used. Some studies only consider the main tweet or post; other studies consider additional aspects of the news item, such as the discussion, replies, and comments [11].

Task 7 at SemEval-2019 was dedicated to rumor evaluation. A number of FND systems were submitted to the RumourEval-2019 [6] challenge. The data for the challenge was obtained from both Twitter and Reddit, and the challenge involves two subtasks, A and B. For subtask A, given a tweet and its conversation thread, the task is to classify the tweet as either Supporting, Denying, Querying, or Commenting on the rumor mentioned by the tweet.

Subtask B is concerned with veracity prediction, i.e., whether the rumor in the tweet is classified as true, false, or unverified. The macro F1 score was used to evaluate the models, and the top three scores were 0.5765, 0.2856, and 0.2620. There was a trend toward using neural network approaches in this challenge. The best performing model was an ensemble of classifiers (SVM, RF, LR) including a NN with three connected layers, where individual post-representations were created using an Long Short Term Memory (LSTM) with attention At RumourEval-2019, there was also a trend toward using neural network-based approaches and pretrained models, such as BERT [6].

At Constraint@AAAI2021, the objective of the COVID19 FND challenge [1] was to create a model that would help determine whether a message about COVID-19 is fake or real news. The challenge organizers created an annotated dataset of 10,700 real and fake social media posts and news articles about COVID-19 in English. The collected dataset was split into a training set (60%), validation set (20%), and test set (20%). At Constraint@AAAI2021, for the COVID19 FND challenge, the TUDublin team constructed an ensemble consisting of bidirectional LSTM, SVM, Logistic Regression, Naive Bayes, and a combination of Logistic Regression and Naive Bayes. The proposed model [12] achieved an F1 score of 0.94, which is within 5% of the best result. Another team participating in the Constraint@AAAI2021-COVID19 FND English challenge used a transformer model for FND [13]. In reporting on this task, the authors [9] describe using transformer-based pretrained models with additional layers to construct a stacking ensemble classifier. The pretrained models were fine-tuned for the FND task. On the challenge test dataset, the models achieved accuracy, precision, recall, and F1 scores of 0.979906542, 0.979913119, 0.979906542, and 0.979907901, respectively.

An FND method that combined Latent Dirichlet Allocation (LDA) topical distributions with contextualized representations from XLNet participated in the Constraint@AAAI2021-COVID19 FND challenge (in English) [14]. Comparing this method with existing baseline methods indicates that topic distributions with XLNet, which achieved an F1 score of 0.967, outperformed other approaches.

Baris and Boukhers [11] presented an FND approach that used a Bidirectional Encoder Representations from Transformers (BERT) language model that considers content information, prior knowledge, and the credibility of sources to detect fake news. The authors conducted a number of experiments on the Constraint@AAAI2021-COVID19 FND challenge datasets (in English) [14]. The highest F1 score obtained ranged between 97.57 and 98.13. Similarly, a previous study [15] evaluated deep learning approaches on the FND task. They evaluated a number of supervised text classification algorithms on the dataset provided by Constraint@AAAI2021-COVID19 FND in an English challenge [14]. The algorithms included Convolutional Neural Networks (CNN), LSTM, and BERT. The best accuracy of 98.41% was obtained on the Covid-19 FND dataset. Another solution, which was ranked within 1.5% of the best performing solutions for the Constraint@AAAI2021-COVID19 FND in an English challenge [14], uses Neural Stacking for FND [16]. Here, the authors employed a heterogeneous representation ensemble adapted for the classification task via an additional neural classification head comprising multiple hidden layers. They conducted ablation studies to understand the behavior of the proposed methods.

Another study [11] investigated a semantic graph approach for rumor detection based on the modeling of the semantic relations between the main posts and replies. This model learns the implicit relations among the main tweet and its replies based on their content. They compared the results to state-of-the-art rumor detection methods on the Twitter datasets described in the literature [17]. They compared the proposed model to feature-based models and deep learning models. The experimental results demonstrated that deep learning models outperformed feature-based models for rumor detection. They also demonstrated that by incorporating implicit semantic relations among all tweets in a thread, the semantic graph approach achieved state-of-the-art performance on both datasets in terms of accuracy.

Another approach that uses a heterogeneous information graph neural network has been proposed previously [18]. The authors used an Adversarial Active Learning-based Heterogeneous Graph Neural Network (AA-HGNN), which employs a novel hierarchical attention mechanism to perform node representation learning in the HIN. The results obtained on two fake news datasets provided F1 scores of 0.57 and 0.70, and these results outperform those of text-based and other graph-based models.

The FakeFlow [19] approach models the flow of affective information in the news to detect if a news item is fake. It targets news articles with longer text, and it is based on the idea that fake news articles often receive reader attention by means of emotional appeals. The authors used neural architectures, i.e., a CNN and Bidirectional Gated Recurrent Units (Bi-GRUs), to model the flow of affection in the news article, and they evaluated the models on three datasets (two available datasets and one dataset created by the authors). They compared their results to several baseline models (CNN, LSTM, HAN, BERT, and Longformer), and the scores achieved by FakeFlow are as follows: accuracy, 0.96; precision, 0.93; recall, 0.97; macro F1 score, 0.96. Note that this model was outperformed slightly by the Longformer model (with a macro F1 score of 0.97).

Another method [20] uses an ensemble learner approach to FND for English. Experimental results demonstrated the ensemble-based approach outperforms individual learners in the FND task.

A previous study [21] employed the pretrained end-to-end BERT model [22]. This system achieved a macro F1 score of 61.67. They reported that adding jointly learned POS, NER, dependency tag embeddings, and third segment embedding or an explicit [SEP] token to separate source and previous posts in BERT's input did not yield improvement.

Another study [23] classified rumors from Twitter and Reddit based on rumor text and the associated discussion thread, i.e., the performed rumor stance (fake/real) classification and veracity prediction. The author used the RumorEval-2019 dataset for this purpose, and the study proposed a method based on classifying the stance of each post in the discussion thread (discussing a rumor). This method is based on multiturn conversational modeling using a transformer-based model, extracting the NLP features of conversations, jointly learning rumor stance, and veracity classification. The architecture includes a base model, Longformer [24], and a number of sentence encoders to learn the different features for stance classification and veracity classification. The author trained different models by varying the type of sentence encoder and learning rate for each configuration. To increase the F1 measure and reduce overfitting, the author employed the Top-Ns fusion strategy [21] to select the best models from the pool of saved models. The resulting models were evaluated using the same guidelines used in the RumorEval-2019 task [6]. They achieved a macro F1 score of 0.5868.

A previous study [25] described an approach to fine-tune transformer-based language models (RoBERTa and CT_BERT) for the FND task. Here, adversarial training was used to improve model robustness. The models were evaluated on an existing COVID-19 fake news dataset [26] and compared to state-of-the-art methods. The results demonstrated superior performances relative to various evaluation metrics, and the best weighted average F1 score was 99.02%.

Transformer models have been used successfully for classification tasks, e.g., the classification of spam reviews. Such models have demonstrated encouraging results. For example, in the literature [27], the authors presented an experiment utilizing Generative Pre-Training 2 (GPT-2) language models to classify spam reviews. They evaluated the approach on TripAdvisor and YelpZip datasets, and the results demonstrated that this method performs 7% better than state-of-the-art methods. They also demonstrated that the model can support data augmentation when labeled data are limited and can generate synthetic spam/nonspam reviews with reasonable perplexity.

Arabic FND is in its infancy compared to English FND; however, it is growing rapidly. For example, a previous study [28] introduced two new datasets of fake and real political news in the Middle East. The fake news dataset includes 3185 articles collected from two Arabic satirical news websites, and the real news dataset includes 3710 articles from credible news websites. They performed an initial exploratory analysis to identify the linguistic properties of Arabic fake news, and then they used these features to construct traditional ML classifiers and neural models to identify the class of the news article. They compared these approaches to a baseline and reported an accuracy of 98.6%.

A feature-based approach to FND using traditional ML methods was presented in the literature [29]. Here, the authors utilized content-related and user-related features and sentiment analysis to generate new features for fake Arabic news detection. They concluded that sentiment analysis improved the prediction accuracy. They experimented with the Random Forest, Decision Tree, AdaBoost, and Logistic Regression algorithms, and the results indicated 76% accuracy for FND.

Another study investigated the credibility of Twitter news items [30]. The authors described a hybrid machine learning approach with a set of topic-related and user-related features to evaluate the news credibility of Arabic news on Twitter. They applied the traditional Decision tree, SVM, and Naive Bayesian ML classifiers on a dataset of 800 Arabic news tweets that were manually labeled. The results demonstrated that the decision tree achieves almost 2% higher than SVM and 7% higher than NB.

Arabic transformer models are increasing interest in the Arabic NLP community. The transformer architecture was introduced in 2017 [31] for language translation based on attention mechanisms without recurrence and convolution layers. Here, the transformer comprises encoder and decoder components, each including self-attention modules, which results in a highly parallelizable architecture that can handle longer sentences [27].

A task relevant to FND is the detection of autogenerated content to determine if sentence is written by humans or generated automatically by a machine. A previous study [32] used a transfer learning-based model to determine whether an Arabic sentence was written by a human or was generated automatically by a machine. The authors combined AraBERT and GPT2 to detect and classify the Arabic autogenerated texts, and they used a Twitter-based dataset and a GPT2-Small-Arabic model to generate fake Arabic sentences. They evaluated their model by comparing recurrent neural network (RNN) word embeddings-based baseline models (LSTM, BI-LSTM, GRU, and BI-GRU) to a transformer-based model. They reported accuracy of up to 98%.

Similarly, another study [9] utilized transformers to generate Arabic fake news. This approach used true online stories and a part of the speech tagger to develop AraNews, a large POS-tagged news dataset that can be used off-the-shelf. The authors also presented models to detect manipulated Arabic news, and they achieved state-of-the-art results on the Arabic FND task with a macro F1 score of 70.06. Note that the models and data used in that study are publicly available.

From our review of related work in the area of FND, it is clear that work on Arabic FND using neural approaches is limited; thus, further research and investigation are required. In addition, to the best of our knowledge, no previous study has experimented with transformers for the FND task for Arabic. Therefore, this study aims to fill this gap and shed some light on neural-based and transformer-based approaches for the FND task.

3. Materials and Methods

The availability of Arabic FND datasets [33] and recent advancements in Arabic transformers and transformer-based approaches have encouraged the Arabic NLP community to further the development of Arabic Transformers, e.g., AraBERT [34], AraELECTRA [35], AraGPT2 [36], QARiB [37] Arbert, and Marbert [38]. The transformers used in our experiments are summarized in the following.

(i) AraBERT [34] is a pretrained contextualized text representation model for the Arabic language. There are a number of versions of AraBERT, including AraBERT v1, AraBERT v02, and AraBERT v2. The popularity of these models has increased recently because they employ transfer learning by fine-tuning a large pretrained language model (self-supervised) for NLP tasks with a small number of labeled examples to obtain good results. AraBERT is pretrained using Modern Standard Arabic (MSA) data, which limits AraBERT's applicability to tasks involving dialects. AraBERT was evaluated on three tasks, i.e., sentiment analysis, named entity recognition, and question answering.

(ii) AraELECTRA [35] is based on the Efficiently Learning an Encoder that Classifies Token Replacements Accurately (ELECTRA) approach [39].

(iii) AraGPT2 [36] is a pretrained transformer for Arabic language generation. AraGPT2 is trained on large Arabic corpora of Internet text and news articles. There are a number of variants available (i.e., base, medium, large, and mega). The largest model (AraGPT2-mega) has 1.46 billion parameters.

(iv) Arbert and Marbert [38] are transformer-based models that exploit large-to-massive scale datasets. These models have been evaluated on several NLP tasks, including sentiment analysis, social meaning prediction, topic classification, dialect identification, and named entity recognition.

We designed a number of experiments using word and document level embeddings for linear and deep learning models (CNN, RNN, GRU) and transformer-based models (AraBERT v1, AraBERT v02, AraBERT v2, ArElectra, QARiB, Arbert, and Marbert).

We first retrieved and compiled each dataset using tweet IDs. We then performed text preprocessing on the datasets. The datasets were then split into training (80%) and validation (20%) sets. The feature extraction included both word and character levels. Finally, the models were constructed and evaluated.

In the following, we discuss the datasets, evaluation metrics, text preprocessing steps, features, model architectures, and the experimental setup.

3.1. Dataset and Evaluation Metrics. In this study, we used the Arabic COVID-19 pandemic tweets collected and published in the ArCOV19-Rumors [7] and Covid-19-Fakes [8] datasets. We also use two general Arabic fake news datasets, i.e., the AraNews dataset [9] and ANS corpus [10].

ArCOV19-Rumors [7] is a human-annotated Arabic COVID-19 Twitter dataset for FND. It contains two subsets, i.e., the claims subset, which includes all relevant tweets of the claims (labeled as true, false, or other), and a tweet verification subset, which only includes relevant tweets that are either expressing or denying. In our experiments, we only utilized the claims subset. The Covid-19-Fakes [8] is an automatically annotated bilingual (Arabic/English) COVID-19 Twitter dataset used for misleading information detection.

The AraNews [9] dataset is a general Arabic misinformation dataset collected from multiple newspapers on multiple topics from 15 Arabic countries, the United Kingdom, and the United States of America. The ANS [10] is a corpus comprising Arabic news titles. The data were collected from several news outlets, e.g., the BBC and CNN, for use in claim verification tasks.

We used the three datasets for training and validation, and the fourth dataset was used for evaluation. The details of these datasets and the distribution of training and validation sets are given in Table 1. For the rumor datasets, we determined the size of the training and validation datasets randomly with a ration 80–20%; therefore, the number of positive and negative labels in each dataset was not constant. For the ANS and AraNews, the training and validation datasets were provided as separate datasets.

The evaluation metrics considered in this study (precision, recall, accuracy, and F1 score) are similar to those used in the literature for the task of FND.

3.2. Text Preprocessing. In our experiments, we used a common pipeline for text preprocessing and two other pipelines, i.e., one for the embedding-based model and the other for the transformer-based model. These pipelines are described below.

(1) Common Pipeline.

(a) Replace hashtags with relevant tokens (xxHash).
(b) Replace emojis with relevant tokens (xxemoji).
(c) Replace HTML.
(d) Replace repetition of words, characters, and successive spaces.
(e) Replace capital letters with lowercase and add special token (xxmaj).
(f) Embed beginning of sentence token (xxbos).

(2) Embedding-based model (after common pipeline).

(a) Segmentation using farasa [40].
(b) Lemmatization using farasa [40].
(c) Replace HTTPS with a relevant token (xxhttps).
(d) Replace mentions with relevant tokens (xxMention).
(e) Remove stop words, punctuation, diacritization, normalization, and non-Arabic letters.
(f) Splitting by character for character level embeddings.
(g) spaCy tokenizer.

(3) Transformer-based model (after common pipeline).

(a) Used official repository preprocessing methods (when mentioned).
(b) Load transformer tokenizer.
(c) Sort tokenizer vocabulary.

(d) Feed tokenizer and sorted vocabulary to Text Block component from fastai (https://github.com/fastai/fastai).

Here, the transformer model has its own tokenizer, which can handle raw data; thus, we reduced the pipeline for the tokenizer such that the tokenizer outputs texts that are similar to what the model is trained on. We used the fastai library to load the data. The Text Block component obtains texts from files or a data frame, applies tokenization and numericalization to the given texts, and provides a simple API for the data loader creation.

The Sorted Data Loader sorts the texts based on their length to reduce padding units as much as possible.

3.3. Model Architectures and Settings. Figure 1 shows the architecture of the customized models. Note that all models share the same embeddings of vector size 100. The linear models comprise two linear layers, the convolution (Conv) layers, the batch normalization layer, and ReLU activation. The sequence model comprises a unidirectional layer of hidden size (100). The final linear layer is appended for all models for classification.

By definition, embeddings are a common block in all model architectures. We set the vector size of the embeddings to 100 so that we can maintain the feasibility of the linear models and unify the embeddings with different architectures.

The linear models comprise two linear layers: linear (5700,1024), ReLU, linear (1024,1), where 57 is the longest text in the corpus. In addition to word embeddings, document embeddings are added in the case of doc2vec, followed by ReLU, linear (5700,100), and linear (200,1). Here, 200 represents the concatenation between document vectors and the output of the first linear layer. In the CNN models, we used four Conv_layers (1,4), Conv_layers (4,8), Conv_layers (8,16), Conv_layers (16,32), an adaptive average pooling layer, and linear (100,1). Here, Conv_layers each comprise a 2D convolution layer of kernel size 3 and stride 2, followed by a batch normalization layer and finally ReLU activation. In the sequence models, we used a unidirectional RNN (100,100), ReLU, and linear (100,1). Note that the same architecture was applied to the GRU.

(i) Linear models Word Level (WL) and Character Level (ChL) with the four settings (Word2Vec-W2V, Glove-G, fastText-F, Doc-D)

(ii) Similarly, DL (CNN, RNN, GRU) models with the four settings (Word2Vec-W2V, Glove-G, fastText-F, Doc-D)

(iii) Transformer-based models (AraBERT v1, AraBERT v02, AraBERT v2, ArElectra, QARiB, ArBert, and MarBert) were run with three different experimental settings: (1) with gradual unfreezing, special learning rate, and learning rate scheduling, (2) with special learning rate and learning rate scheduling, and (3) with a constant learning rate of 1e-5.

3.4. Experimental Setup. In these experiments, we used twarc (https://github.com/DocNow/twarc) to get the details of the tweets (hydrate) Twitter tweets using their IDs, and then we linked the tweet to its designated class. Here, we retrieved 85% from the rumors class and 37% from Covid-19-Fakes. For lemmatization and segmentation, we used farasa [40]. We then used our custom with the fastai library's default preprocessing methods. If the model has special preprocessing steps, we appended these steps to the preprocessing pipeline. To build our embeddings, we used gensim, which is a python library for NLP that includes implementations for word2vec [41], fastText [42], and doc2vec [43] embeddings. In addition, we used the glove library [44] to train the glove embeddings. We used a similar configuration for all embeddings. Here, the vector size was

Table 1: Details of Arabic dataset statistics used in this study.

Dataset		Real positive	Fake negative	Total
	All	1635	1397	3032
ArCOV19-rumors	Training	—	—	2424
	Validation	—	—	608
	All	52648	55546	108194
AraNews	Training	46193	48845	95038
	Validation	6455	6701	13156
	All	1325	2766	4091
Arabic news stance (ANS)	Training	1035	2150	3185
	Validation	290	616	906
Covid-19-fakes evaluation	All	69126	1833	70959

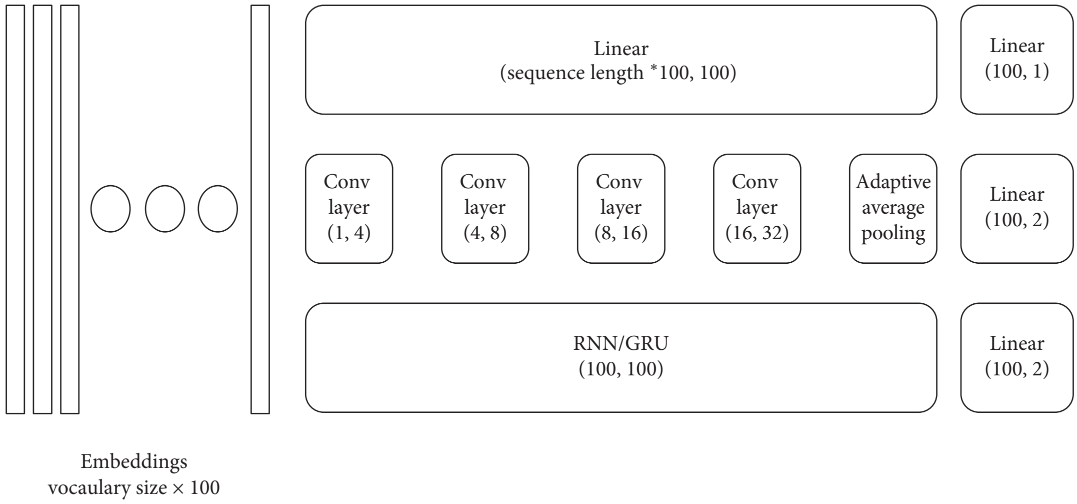


Figure 1: FND model architecture.

set to 100, the minimum frequency was set to 3, and trained was performed on Covid-19-Fakes for 10 epochs. In this training, we enforced the addition of extra tokens for unknown.

We created an embedding-based model with the identified dimensions. Using fastai, we constructed a new vocabulary based on the ArCOV19-Rumors dataset, and we determined the size of the models' embeddings to have the same size as the vocabulary. We then mapped the embeddings from the pretrained vocabulary. Here, for each word in the rumors class that was not present in the pretrained embeddings, we loaded the weights of the unknown token. We trained the model for 10 epochs. 4 with freezing the embeddings, then 2 without any freezing, and at last, 4 epochs with changing the optimizer from Adam to SGD. For the doc2vec embeddings, we concatenated the document embedding input of the final linear layer and doubled the size of this layer.

For the tokenizer-based model, we loaded the tokenizer using the transformer library (https://huggingface.co/). After that, we added our own special token to the tokenizer and resized the embeddings of the model. This was followed by sorting the tokenizer vocabulary based on its index and feeding the results into the fastai text block component. This step is important to ensure correct numericalization. Note that we trained the transformer models for only five epochs. Three modes of training were applied, i.e., using a constant learning rate without freezing, using a learning rate finder with a learning scheduling and applying gradual unfreezing and the learning rate finder technique with learning rate scheduling.

We used the Adam optimizer [45] with momentums equal to 0.9 and 0.99, epsilon equal to 11-05, and a weight decay of 0.01. For SGD, we used zero for both the weight decay and momentum values. For the learning rate, we used both grid search and the learning rate finder technique from the fastai library. Here, we used cosine annealing for learning rate scheduling, ReLU as the activation function, and binary cross entropy for the loss function. To evaluate the models, we considered the F1 score, precision, recall, and accuracy, which were implemented using the sci-kit-learn library [46].

4. Results and Discussion

The results obtained by training the models on the three different datasets (ArCOV19-Rumors, AraNews, and ANS) are shown in Table 2. The transformer-based models were trained using various configurations in terms of learning rate and gradual unfreezing. Figure 2 shows a bar chart comparison of the models. The ROC curves for the three models are plotted in Figure 3.

Table 2: FND model performance on ArCOV19-Rumors, ANS, and AraNews datasets.

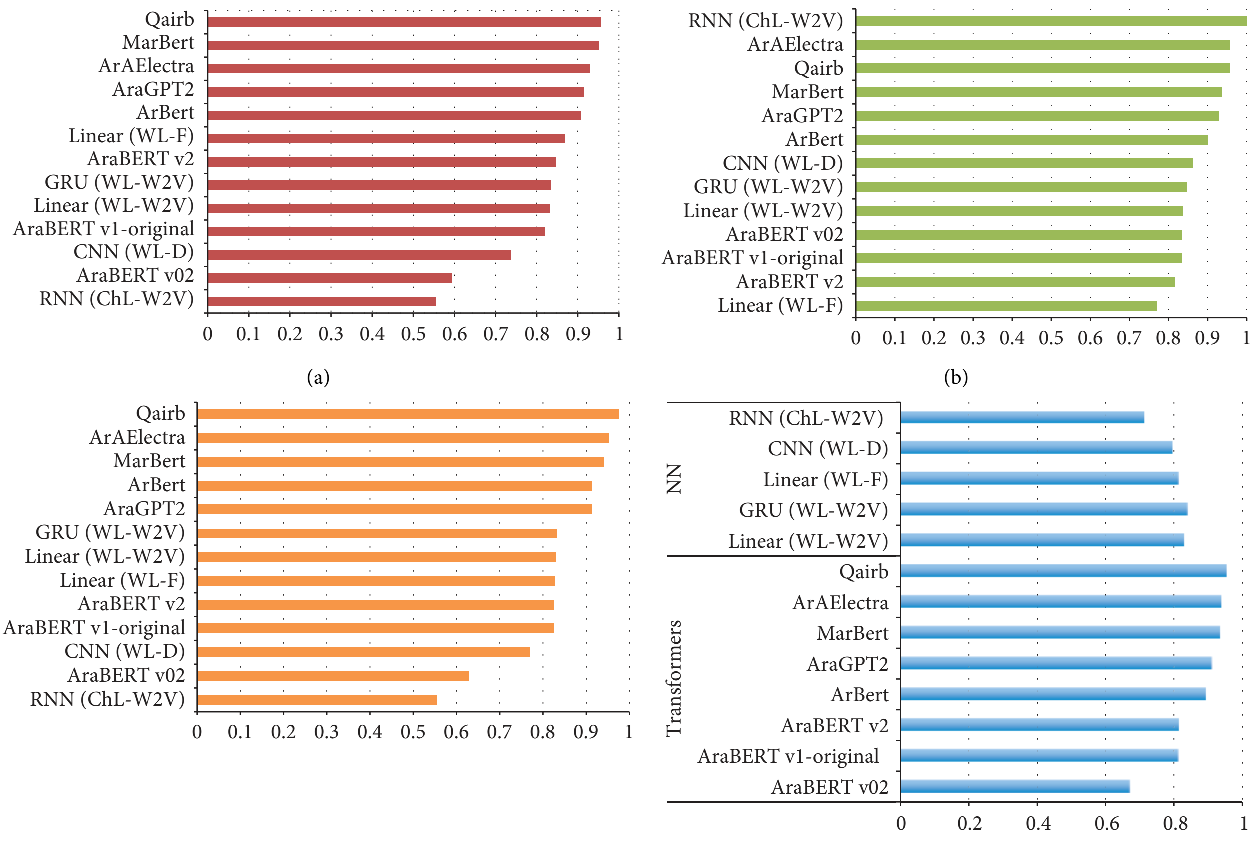
Model \ measure	F1	Precision	Recall	Accuracy
Linear and deep learning (ArCOV19-Rumors)				
Linear (WL-W2V)	0.827886	0.831215	0.836915	0.829752
CNN (WL-F)	0.782586	0.881676	0.70797	0.791736
RNN (ChL-F)	0.712743	0.555372	1	0.555372
GRU (WL-W2V)	**0.838259**	0.833348	0.846716	**0.831405**
Transformers with gradual unfreezing, special learning rate, and learning rate scheduling (ArCOV19-Rumors)				
AraBERT v1-original	0.811362	0.819645	0.83294	0.825083
AraBERT v02	0.669801	0.594531	0.835298	0.628713
AraBERT v2	0.812187	0.846519	0.816502	0.825083
ArAElectra	0.936161	0.929813	0.95675	0.952145
AraGPT2	0.908726	0.914789	0.927801	0.912541
QARiB	**0.953345**	0.956216	0.956404	**0.975248**
ArBert	0.891291	0.907135	0.90165	0.914191
MarBert	0.933561	0.950039	0.934968	0.940594
Transformers with a constant learning rate of 1e-5 (ArCOV19-Rumors)				
AraBERT v02	0.705171	0.722772	0.69802	0.945545
AraGPT2	0.776255	0.774085	0.806601	0.813531
ArBert	**0.953423**	0.947729	0.966777	**0.958746**
Transformers with special learning rate and learning rate scheduling (ArCOV19-Rumors)				
AraBERT v02	0.928597	0.960616	0.911881	0.948845
QARiB	**0.930478**	0.941914	0.92557	**0.952145**
AraGPT2	0.90882	0.919802	0.927927	0.925743
Transformers with special learning rate and learning rate scheduling (ANS dataset)				
AraBERT v02	0.02752	0.06181	0.018186	0.675497
QARiB	0.058888	0.10596	0.045412	0.688742
AraGPT2	**0.203931**	0.278146	0.191023	0.642384
Transformers with special learning rate and learning rate scheduling (AraNews dataset)				
AraBERT v02	0.886098	0.800455	0.999121	0.800264
QARiB	0.887142	0.800422	0.999093	0.800264
AraGPT2	0.659546	0.509349	1	0.509349

The results shown in Table 2 demonstrate that the transformer-based models generally outperformed the basic deep learning models, which are based on linear, CNN, GRU, or LSTM blocks. This can be explained by multiple factors, e.g., the huge language knowledge gained by the transformers by training them on language modeling objectives. Another factor is that the embedding = based models were trained on a limited dataset compared to the transformers. Even though our embeddings were trained on a dataset from the same domain, they were not able to obtain high scores. In contrast, the transformer models were trained on multiple topics and were more efficient at achieving good results on the limited dataset. This result emphasizes the importance of training a language model on various topics and indicates the superiority of transformers over embeddings for Arabic FND.

By analyzing the data of the embedding-based models, we found that models can repeatedly fall into predicting a single class. However, few experiments resulted in unexpected results, e.g., the linear models with word2vec and fastText (≅ 0.83 accuracy), which demonstrates the superiority of LSTM-based and CNN-based models. In contrast, WL-GRU-W2vec obtain an accuracy of 0.83. Our intuition is that linear models are the best to deal with such a small dataset without overfitting.

In terms of the transformer-based models, we found it difficult to determine the best performing models because models can behave differently depending on the training methods. However, we found that QARiB obtained high scores under various training settings, exceeding an accuracy of 0.95. AraBERT v02 was one of the best models but only when the learning rate was well determined. In addition, AraGPT2 obtained interesting results despite the fact that it was originally trained on text generation. In addition, AraGPT2 performed better with a higher learning rate. This is illustrated in Table 2, which shows that AraGPT2 obtained better results with a learning rate of 1e-4 compared to a learning rate of 1e-5 or 1e-6.

As mentioned previously, we applied various training modes. Based on the results, we failed to determine a rule of thumb for training transformer models. For gradual unfreezing with the learning scheduler and finder experiment, we selected the best and worst performing models, i.e., QARiB (accuracy: 0.958) and AraBERT V02 (accuracy: 0.62), respectively and the AraGPT2 (accuracy: 0.91) due to the uniqueness of its architecture. Here, we applied the learning rate scheduler and finder without gradual unfreezing. Eventually, the results were confusing because the best performing QARiB resulted from using the first model (accuracy: 0.958), and the best performing AraBERT V02



(a)

(b)

(c)

(d)

Figure 2: Comparison of precision, recall, accuracy and F1 scores for models applied on the validation dataset (ArCOV19-Rumors, ANS, and AraNews).

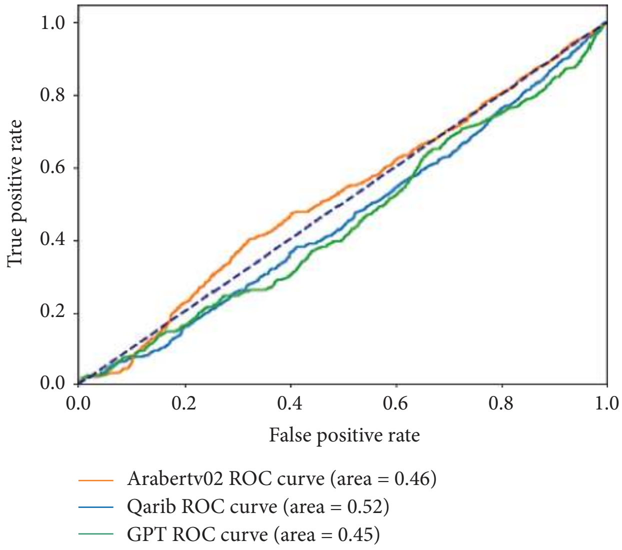


Figure 3: ROC curve for the models showing that QARiB has the best performance compared to AraBERTv02, AraGPT2.

Table 3: FND model performance on COVID-19-FAKES dataset.

Model	F1 score	Precision	Recall	Accuracy
AraBERT v02	**0.6731**	**0.9682**	**0.5426**	**0.5379**
AraGPT2	0.5790	0.9570	0.4405	0.4397
QARiB	0.6061	0.9649	0.4641	0.4683

(accuracy: 0.953) was obtained using the learning scheduler and learning scheduler finder without gradual unfreezing, which is the same as the best AraGPT2 (accuracy: 0.92). However, we conclude that gradual unfreezing may impact the performance of the models, e.g., in the QARiB case.

We trained Arabertv02, QARiB, and AraGPT2 on the AraNews dataset and ANS corpus. Here, two of the three models performed well with the learning rate scheduler with a special learning rate and without gradual freezing; thus, we decided to apply the same configurations in these experiments. Table 2 shows that both AraBERT V02 and QARiB obtained a similar accuracy value of approximately 0.8. In contrast, in the ANS experiment, QARiB obtained an accuracy of 0.68. Our interpretation of these results is due to the small size of the ANS dataset. In both experiments, AraGPT2 obtained the lowest accuracy.

It is important to point out that there is duplication in tweets in the rumors dataset, where two tweets have nearly the same content with different IDs. This led to a shortcoming in our experiment, which is the probability of having a validation point that the model was already trained on. However, this does not impede our discussion of the models because all models were trained in the same environment. Duplicates must be removed to solve this problem; however, this will lead to another problem, i.e., reduction of the dataset size. Another solution could be finding another dataset that is annotated by humans or uses a machine generated dataset, which would be less reliable but more abundant.

To evaluate the generalizability of the models, we evaluated the models on the Covid-19-Fakes dataset, and the results are shown in Table 3. An important advantage of this dataset is that it is related to the same topic as the rumors dataset.

As shown in Table 3, AraBERT v02 outperformed all other models in terms of generalization, and the models achieved a moderate F1 score. We consider that these results could be improved by training the models on larger datasets in the same domain as the test dataset. Note that we could not train the models using the Covid-19-Fakes dataset due to its huge class imbalance. This may provide an indication of why the models obtained very high precision scores. It is also difficult to compare models in terms of the generalization results because, in the training configuration, we set the training and validation points randomly for each training experiment. Therefore, we cannot precisely identify the impact of this configuration on model performance.

5. Conclusion and Future Work

In this paper, we have discussed a number of experiments conducted to empirically analyze whether current advances in deep learning models and large-scale language models for the Arabic language can benefit the Arabic FND task. Our experimental results demonstrate that transformer-based models outperform neural network-based solutions. In addition, we found that AraBERT v02 outperformed all compared models in terms of generalization. Although this work provides contributions toward realizing Arabic FND, we observed several limitations and challenges. First, regarding the data, we used a small dataset, and the repetition of tweets and unavailable tweets was problematic. In addition, the data suffered from noise and tweets that did not belong to any class. In the future, we can use a gold-standard dataset annotated by humans or use a machine generated dataset, which may be less reliable but would be more abundant. In addition, we could employ model ensemble and stacking techniques to experiment with new model architectures, or we could use deep neural models for feature extraction and then use traditional machine learning for the classification task.

Data Availability

The data used in the experiments are available upon request.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

This research project was supported by a grant from the "Research Center of the Female Scientific and Medical Colleges," Deanship of Scientific Research, King Saud University, Riyadh, Saudi Arabia.

References

[1] P. Patwa, M. Bhardwaj, V. Guptha et al., "Overview of CONSTRAINT 2021 shared tasks: detecting English COVID-19 fake news and hindi hostile posts," in *Proceedings of the CONSTRAINT 2021*, Delhi, India, March 2021.

[2] N. K. Conroy, V. L. Rubin, and Y. Chen, "Automatic deception detection: methods for finding fake news," *Proceedings of the Association for Information Science and Technology*, vol. 52, no. 1, pp. 1–4, 2015.

[3] G. Gorrell, K. Bontcheva, L. Derczynski, E. Kochkina, M. Liakata, and A. Zubiaga, "RumourEval 2019: Determining Rumour Veracity and Support for Rumours," February 2021, http://arxiv.org/abs/1809.06683.

[4] X. Zhou and R. Zafarani, "A survey of fake news," *ACM Computing Surveys*, vol. 53, no. 5, pp. 1–40, 2020.

[5] L. Derczynski, K. Bontcheva, M. Liakata, R. Procter, G. Wong Sak Hoi, and A. Zubiaga, "SemEval-2017 task 8: RumourEval: determining rumour veracity and support for rumours," in *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pp. 69–76, Vancouver, Canada, August 2017.

[6] G. Gorrell et al., "SemEval-2019 task 7: RumourEval, determining rumour veracity and support for rumours," in *Proceedings of the 13th International Workshop on Semantic Evaluation*, pp. 845–854, Minneapolis, Minnesota, USA, June 2019.

[7] F. Haouari, M. Hasanain, R. Suwaileh, and T. Elsayed, "ArCOV19-rumors: Arabic COVID-19 twitter dataset for misinformation detection," October 2020, http://arxiv.org/abs/2010.08768.

[8] M. K. Elhadad, K. F. Li, and F. Gebali, "COVID-19-FAKES: a twitter (Arabic/English) dataset for detecting misleading information on COVID-19," in *Advances in Intelligent Networking and Collaborative Systems*, pp. 256–268, Springer, Cham, Switzerland, 2021.

[9] E. M. B. Nagoudi, A. Elmadany, M. Abdul-Mageed, T. Alhindi, and H. Cavusoglu, "Machine generation and detection of arabic manipulated and fake news," February 2021, http://arxiv.org/abs/2011.03092.

[10] J. Khouja, "Stance prediction and claim verification: an Arabic perspective," in *Proceedings of the Third Workshop on Fact Extraction and VERification (FEVER)*, pp. 8–17, Seattle, WA, USA, July 2020.

[11] A. P. B. Veyseh, M. T. Thai, T. H. Nguyen, and D. Dou, "Rumor detection in social networks via deep contextual modeling," in *Proceedings of the 2019 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM)*, pp. 113–120, Vancouver, Canada, August 2019.

[12] E. Shushkevich and J. Cardiff, "TUDublin team at constraint@AAAI2021—COVID19 fake news detection," February 2021, http://arxiv.org/abs/2101.05701.

[13] S. M. S.-U.-R. Shifath, M. F. Khan, and M. S. Islam, "A transformer based approach for fighting COVID-19 fake news," February 2021, http://arxiv.org/abs/2101.12027.

[14] A. Gautam and S. Masud, “Fake news detection system using XLNet model with topic distributions: CONSTRAINT@ AAAI2021 shared task,” February 2021, http://arxiv.org/abs/2101.11425.

[15] A. Wani, I. Joshi, S. Khandve, V. Wagh, and R. Joshi, “Evaluating deep learning approaches for Covid19 fake news detection,” February 2021, http://arxiv.org/abs/2101.04012.

[16] B. Koloski, T. S. Perdih, S. Pollak, and B. Škrlj, “Identification of COVID-19 related fake news via neural stacking,” February 2021, http://arxiv.org/abs/2101.03988.

[17] J. Ma, W. Gao, and K.-F. Wong, “Detect rumor and stance jointly by neural multi-task learning,” in *Proceedings of the Companion Proceedings of the the Web Conference 2018*, pp. 585–593, Republic and Canton of Geneva, Switzerland, April 2018.

[18] Y. Ren, B. Wang, J. Zhang, and Y. Chang, “Adversarial active learning based heterogeneous graph neural network for fake news detection,” February 2021, http://arxiv.org/abs/2101.11206.

[19] B. Ghanem, S. P. Ponzetto, P. Rosso, and F. Rangel, “Fake-Flow: fake news detection by modeling the flow of affective information,” February 2021, http://arxiv.org/abs/2101.09810.

[20] I. Ahmad, M. Yousaf, S. Yousaf, and M. O. Ahmad, “Fake news detection using machine learning ensemble methods,” *Complexity*, vol. 2020, Article ID 8885861, 11 pages, 2020.

[21] M. Fajcik, P. Smrz, and L. Burget, “BUT-FIT at SemEval-2019 task 7: determining the rumour stance with pre-trained deep bidirectional transformers,” in *Proceedings of the 13th International Workshop on Semantic Evaluation*, pp. 1097–1104, Minneapolis, MN, USA, June 2019.

[22] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, “BERT: pre-training of deep bidirectional transformers for language understanding,” February 2021, http://arxiv.org/abs/1810.04805.

[23] A. Khandelwal, “Fine-Tune Longformer for Jointly Predicting Rumor Stance and Veracity,” in *Proceedings of the 8th ACM IKDD CODS and 26th COMAD*, pp. 10–19, New York, NY, USA, January 2021.

[24] I. Beltagy, M. E. Peters, and A. Cohan, *Longformer: the long-document transformer*, http://arxiv.org/abs/2004.05150, February 2021.

[25] B. Chen et al., “Transformer-based language model fine-tuning methods for COVID-19 fake news detection,” February 2021, http://arxiv.org/abs/2101.05509.

[26] P. Patwa, S. Sharma, S. Pykl et al., “Fighting an infodemic: COVID-19 fake news dataset,” 2020, https://arxiv.org/abs/2011.03327.

[27] A. A. Irissappane, H. Yu, Y. Shen, A. Agrawal, and G. Stanton, “Leveraging GPT-2 for classifying spam reviews with limited labeled data via adversarial training,” January 2021, http://arxiv.org/abs/2012.13400.

[28] H. Saadany, C. Orasan, and E. Mohamed, “Fake or real? A study of Arabic satirical fake news,” in *Proceedings of the 3rd International Workshop on Rumours and Deception in Social Media (RDSM)*, pp. 70–80, December 2020, https://www.aclweb.org/anthology/2020.rdsm-1.7.

[29] G. Jardaneh, H. Abdelhaq, M. Buzz, and D. Johnson, “Classifying Arabic tweets based on credibility using content and user features,,” in *Proceedings of the 2019 IEEE Jordan International Joint Conference on Electrical Engineering and Information Technology (JEEIT)*, pp. 596–601, Amman, Jordan, April 2019.

[30] S. Sabbeh and S. Baatwah, “Arabic news credibility on twitter: an enhanced model using hybrid features,” *Journal of Theoretical and Applied Information Technology*, vol. 96, pp. 2327–2338, 2018.

[31] A. Vaswani et al., “Attention is all you need,” in *Proceedings of the 31st International Conference on Neural Information Processing Systems*, pp. 6000–6010, Red Hook, NY, USA, December 2017.

[32] F. Harrag, M. Dabbah, K. Darwish, and A. Abdelali, “Bert transformer model for detecting Arabic GPT2 auto-generated tweets,” in *Proceedings of the Fifth Arabic Natural Language Processing Workshop*, pp. 207–214, December 2020, https://www.aclweb.org/anthology/2020.wanlp-1.19.

[33] M. Alkhair, K. Meftouh, K. Smaïli, and N. Othman, “An Arabic corpus of fake news: collection, analysis and classification,” in *Arabic Language Processing: From Theory to Practice*, pp. 292–302, Springer, Cham, Switzerland, 2019.

[34] W. Antoun, F. Baly, and H. Hajj, “AraBERT: Transformer-Based Model for Arabic Language Understanding,” in *Proceedings of the Twelfth International Conference on Language Resources and Evaluation (LREC 2020)*, Marseille, France, May 2020.

[35] W. Antoun, F. Baly, and H. Hajj, “AraELECTRA: pre-training text discriminators for Arabic language understanding,” 2020, https://arxiv.org/abs/2012.15516.

[36] W. Antoun, F. Baly, and H. Hajj, “AraGPT2: pre-trained transformer for Arabic language generation,” February 2021, http://arxiv.org/abs/2012.15520.

[37] A. Ahmed, H. Sabit, M. Hamdy, D. Kareem, and S. Younes, “QARiB: QCRI Arabic and dialectal BERT,” 2020, https://github.com/qcri/QARIB.

[38] M. Abdul-Mageed, A. Elmadany, and E. M. B. Nagoudi, “ARBERT & MARBERT: deep bidirectional transformers for Arabic,” 2020, https://arxiv.org/abs/2101.01785.

[39] K. Clark, M.-T. Luong, Q. V. Le, and C. D. Manning, “ELECTRA: pre-training text encoders as discriminators rather than generators,” February 2021, http://arxiv.org/abs/2003.10555.

[40] A. Abdelali, K. Darwish, N. Durrani, and H. Mubarak, “Farasa: a fast and furious segmenter for Arabic,” in *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Demonstrations*, San Diego, CA, USA, 2016.

[41] T. Mikolov, K. Chen, G. Corrado, and J. Dean, “Efficient estimation of word representations in vector space,” in *Proceedings of the 1st International Conference on Learning Representations, ICLR 2013*, Scottsdale, AZ, USA, May 2013.

[42] P. Bojanowski, E. Grave, A. Joulin, and T. Mikolov, “Enriching word vectors with subword information,” 2016, https://arxiv.org/abs/1607.04606.

[43] Q. Le and T. Mikolov, “Distributed representations of sentences and documents,” in *Proceedings of the 31st International Conference on Machine Learning*, vol. 32, no. 2, pp. 1188–1196, Bejing, China, June 2014.

[44] J. Pennington, R. Socher, and C. Manning, “GloVe: global vectors for word representation,” in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 1532–1543, Doha, Qatar, October 2014.

[45] D. P. Kingma and J. Ba, “Adam: A Method for Stochastic Optimization,” in *Proceedings of the Paper Presented at the the 3rd International Conference for Learning Representations*, San Diego, CA, USA, 2015, http://arxiv.org/abs/1412.6980.

[46] F. Pedregosa, G. Varoquaux, A. Gramfort et al., “Scikit-learn: machine learning in Python,” *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.