Arabic Quran Verses Authentication Using Deep Learning and Word Embeddings

Zineb Touati-Hamad
Laboratory of Mathematics, Informatics and Systems, University Larbi Tebessi, Algeria
zineb.touatihamad@univ-tebessa.dz

Mohamed Ridda Laouar
Laboratory of Mathematics, Informatics and Systems, University Larbi Tebessi, Algeria
ridda.laouar@univ-tebessa.dz

Issam Bendib
Laboratory of Mathematics, Informatics and Systems, University Larbi Tebessi, Algeria
issam.bendib@univ-tebessa.dz

Saqib Hakak
Faculty of Computer Science, University of New Brunswick, Canada
saqib.hakak@unb.ca

Abstract: *Nowadays, with the developments witnessed by the Internet, algorithms have come to control all aspects of digital content. Due to its Arabic roots, it is ironic to find that Arabic Quranic content is still thirsty to benefit from computer linguistics, especially with the advent of artificial intelligence algorithms. The massive spread of Islamic-typed websites and applications has led to a widespread of digital Quranic content. Unfortunately, such content lacks censorship and can rarely match resourcefulness. It is quite difficult, especially for a non-native speaker of the Arabic language, to distinguish and authenticate the provided Quranic verses from the non-Quranic Arabic texts. Text processing techniques classified outside the field of Natural Language Processing (NLP) give less qualified results, especially with Arabic texts. To address this problem, we propose to explore Word Embeddings (WE) with Deep Learning (DL) techniques to identify Quranic verses in Arabic textual content. The proposed work is evaluated using twelve different word embeddings models with two popular classifiers for binary classification, namely: Convolutional Neural Network (CNN) and Long Short Term Memory (LSTM). The experimental results showed the superiority of the proposed approach over traditional methods in distinguishing between the Quranic verses and the Arabic text with an accuracy of 98.33%.*

Keywords: *Arabic text, Quranic verse, Authentication, NLP, Word Embeddings, Word2vec, DL, CNN, LSTM.*

Received February 3, 2021; accepted October 10, 2021
https://doi.org/10.34028/iajit/19/4/13

1. Introduction

The digital Holy Quran is one of the holiest books among 1.3 billion Muslims around the world. The Holy Quran consists of 114 chapters (Surahs) of varying lengths and is available in different writing styles such as Uthmani, non-Uthmani, etc., [34]. The holy Quran was revealed in the Arabic language and was preserved from all distortion and corruption, unlike the rest of the heavenly books. Therefore, the role of every Muslim is to work to preserve the authenticity and integrity of this noble Book [17, 18].

Identification of the Quranic verses/surahs can be defined as distinguishing the words of the text corresponding to the content of the Holy Quran, while authentication is ensuring that they are written exactly as it is written in all copies of the Holy Quran [29].

With the development of internet applications, Quranic verses are increasingly quoted and cited on blogs, forums, and social media sites. Verses are copied from unreliable websites, so it is difficult for the reader to distinguish between authentic Quran verses from the regular text [12, 29]. The problem is more challenging to non-native speakers of the Arabic language who are not familiar with the language and might get confused easily. For native speakers, although it is easier to distinguish verses written in uthmani script, diacritic verses, and verses written between two brackets, the problem lies in distinguishing non-diacritic verses where it becomes difficult to determine the beginning and the end of each verse or surah incorporated in an Arabic text. Text processing techniques classified outside the field of Natural Language Processing (NLP) give less qualified results, especially with Arabic Quranic texts, as they require time and in-depth analysis into the original version of the Quran text [6]. Therefore, it is very important to develop intelligent systems that can help analyze this type of online resource.

Deep Learning (DL) is a branch of artificial intelligence, one of its most important studies is the application of deep neural networks in solving problems in the field of computational linguistics and NLP [27]. DL relies on incremental training of layers on more complex representations of data at the same time. The power of DL lies in eliminating the feature engineering phase, which is one of the major bottlenecks in machine learning pipelines [27]. In addition, DL classifiers outperform traditional classifiers in achieving more gains when dealing with sequence-sensitive texts. Therefore, it is considered an excellent solution for dealing with Quranic texts.

At present, there is no automated application that can recognize Quranic verses using DL techniques.

Hence, the main contribution of this research study is to propose a new approach based on DL and Word Embeddings (WE) techniques to build models capable of automatically classifying the content of Quranic and Arabic texts. The remainder of this paper is organized as follows: section 2 presents the related works. In section 3, we present an overview of the charges and motivations. Section 4 describes our methodology of classifying the Quranic text. The experiments and results are discussed in section 5. Finally, the work is concluded in section 6.

2. Related Work

There is not much work available in the area of Digital Quran authentication. The works in [18, 20] present a comprehensive review of state-of-the-art Digital Quran and Hadith authentication approaches. Similarly. [17, 36] presents recent advances in authenticating digital content with the main focus on Arabic content which is available in both text and image formats respectively. Few of the related works include the works of [9, 11, 12, 14, 15, 16, 19] where the authors have proposed exact string-matching based approaches [13] to detect uthmani and non-uthmani Quranic verses. [23] has proposed a watermarking-based approach to protect and detect Quranic verses. In one of the latest works, [5] have proposed a cryptographic hash function to preserve the integrity of the digital Quran. All of the above-mentioned approaches have focused on authentication and preserving the integrity of the Digital Quran content. None of these works has addressed the problem of classifying the Quranic content from the regular text using DL techniques. Although, few works have focused on classification problems concerning the Arabic content such as the work of [1, 8, 22].

However, the scope of these works is limited and has focused on different aspects of the Arabic language such as abuse detection, mood detection, and opinion mining aspects.

3. Challenges and Motivations

Social networking sites are the most popular platforms for sharing Quranic verses. Quranic verses can be identified by relying on some beginning words such as "قال الله تعالى" or ending words such as "صدق الله العظيم", or by determining the percentage of diacritics, and so on. These techniques are considered traditional and require a great deal of time and effort, as they always need a reference for approval. It is not difficult to document the Quranic verses, but it requires manual identification in advance, which makes the process tedious and not instantaneous. Unfortunately, these forms are considered more search engines than authentication systems [30].

The power of DL in automatic text processing appears in understanding and classifying texts in an automated and real-time manner. Comprehensive training is a key feature of DL that makes it a powerful tool for NLP. There are several tasks in which DL models are significantly superior to the rest of the previous models. First, DL can extract features automatically as it can handle even an unlimited number of features. Also lies the ability to work with insufficient knowledge, in our case, we train the models on only a certain percentage of the Quran and the rest is used for verification. Finally, some neural networks are dedicated to dealing with texts in general, and other deep neural networks are dedicated to dealing with sequential texts in particular [33].

Many classic classification algorithms can do a pretty good job. If we look at other non-neural network classification techniques, they are trained on multiple words as separate inputs that are nothing but a word with no real meaning as a sentence, and while predicting the class, it will give the output according to the statistics and not according to the meaning. One of the good reasons to use DL models is that it is effective in memorizing important information. We can use a multi-word string to find out which category it belongs to. This is very useful while working with NLP. If we use appropriate layers of embedding and coding in DL models, the model will know the actual meaning in the input string and give the most accurate output class.

4. Proposed Methodology

Figure 1 illustrates the overall proposed layout for identifying the Arabic Quranic verses consisting of several steps. The data were first collected from the two primary sources of tanzil.net[1] and Arabic Learner Corpus (ALC[2]). Then, the collected data were passed through the preprocessing step. The text representation step receives datasets in order to represent words in dimensional vectors. In order to preserve the relationships between words in the text, we choose the WE technique to extract numerical features from the text. We trained the Convolution Neural Network (CNN) and Convolution Neural Network-Long Short Term Memory (CNN-LSTM) models using the training data. Finally, we test the models by using the testing dataset vectors.

[1]Tanzil.net is a Quranic initiative founded in 2007 to generate a thoroughly vetted Unicode Quran text to be text utilized in Quranic websites and apps.
[2]Arabiclearnercorpus.com (ALC) is a project featuring a collection of written and spoken materials from learners of Arabic in Saudi Arabia.

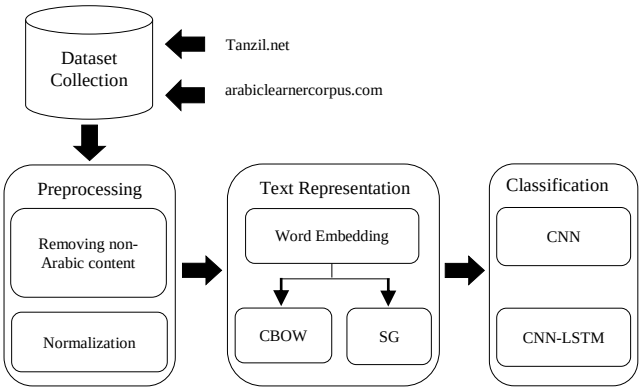

Figure 1. Structure of the proposed approach.

4.1. Dataset Collection

To train the algorithm, we explored two popular datasets. Concerning the Holy Quran, the verified Unicode Quranic text, which serves as a reliable source provided by tanzil.net [35], was used and named the "Verse-Text" class. In tanzil.net, Quranic texts are usually represented in two general formats: Uthmani Script and Simple Script. The other available formats include Text (TEXT) format, Structured Query Language (SQL) format, and Extensible Markup Language (XML) format. Figure 2 shows the "Al Fatiha" verse in XML format, structured as tree nodes, where the "aya" node is the child of the "surah" node, which is the child of the Quran root node.

```
<quran>
<sura index="1" name="الفاتحة">
<aya index="1" text="بسم الله الرحمن الرحيم" />
<aya index="2" text="الحمد لله رب العالمين" />
<aya index="3" text="الرحمن الرحيم" />
<aya index="4" text="مالك يوم الدين" />
<aya index="5" text="إياك نعبد وإياك نستعين" />
<aya index="6" text="اهدنا الصراط المستقيم" />
<aya index="7" text="صراط الذين أنعمت عليهم غير المغضوب عليهم
ولا الضالين" />
</sura>
</quran>
```

Figure 2. Simple Quranic text as in tanzil.net XML format.

```
<ALC>
  <doc ID="S001_T1_M_Pre_NNAS_W_C">
    <header>
      …
                                  </header>
    <text>
      <title> رحلة إلى أبها </title>
      <text_body>

قمنا برحلة إلى أبها، كنا تسعة أشخاص، كان موعد تجمعنا في ملعلب إسكان الطلاب
الساعة 9 15، وبعد ذلك انطلقنا إلى المطار الملك خالد بالرياض، عند وصولنا في
المطار، ركبنا بطائر إلى أبها، لما وصلنا في أبها جاء رجلان للاستقبالنا، ثم بعد السلام
نقلوانا بحافلة في طريق إلى الفندق أبها فيها مناظر جميلة، شهادة القرون في طريق
والجبال، وذهبنا لذيارت جامعة الملك خالد الجامعة مشاء الله جميل جداً، أشكر عمادة بقيام
هذه رحلة جزاهم الله خيراً

      </text_body>
    </text>
```

Figure 3. Simple arabic text as in ALC XML format.

For regular Arabic texts, we explored the Arabic Learner Corpus website, which provides a collection of written and spoken materials produced by learners of Arabic in Saudi Arabia [4]. This data was used in our approach as the "Arabic-Text" class. Figure 3 shows an example of this data in XML format, where "title" and "text body" nodes are both children of the text root node used in our data.

4.2. Preprocessing

The Quranic dataset comprises 6236 documents labeled as "Verse-text", and the Arabic dataset consists of 1585 Arabic documents labeled as "Arabic-text". For the normalization of datasets, the following preprocessing steps were taken:

- Organizing the second dataset: the paragraphs of the second dataset were very long, in this step we split them into proper sentences based on punctuation marks. In the end, we got 6375 valuable sentences.
- Removing punctuation and dialectics: punctuation marks and Harakah, such as Fathah / ـَ /, Dammah / ـُ /, Kasrah / ـِ /, Sukoon / ـْ /, Shaddah / ـّ / and Tanween / ـٌ ـٍ ـً / were filtered.
- Removing non-Arabic characters: filtering non-Arabic content is an important step when dealing with digital data from the web because there is another language whose letters overlap with the Arabic alphabet, such as Urdu and Persian languages [32].
- Removing special symbols and links: in this step, all the emojis, emoticons, dates, times, and URLs were removed.
- Removing Kasheeda: kasheeda is a type of extended character, for example, "Bismi Allah" with and without kasheeda may look like the following:

Word	Normal	Kasheeda
Bism	بسم	بســــــــم
Allah	الله	اللـــــــــه

- Normalizing of Alefs, Alef Maksura, and Tah Marbutah: Replacing Hamza / أ and إ / and Maddah / آ / with simple Alif / ا /, while replacing the letter Dagger Alif / ى / with Yā / ي /, and replacing Tā' marbūṭah / ة / with hāʾ / ه /. This is because Internet users often abuse these characters due to their similarity [2] and the difficulty of finding some of them on the keyboard.

4.3. Text Representation

WE [7] is a distributed representation of document vocabulary in a vector space. One of the common techniques to construct such an embedding is word2vec [10, 25, 26]. It accepts individual words of the corpus as input and produces a real-valued vector in a high-dimensional space. This technique captures the context, the semantic as well as the structure of words, this helps to map the similar words in geometrically close vectors spaces. The Word2Vec tool is considered as a collection of two different neural-network architectures: Continuous Bag of

Words (CBOW) and Skip-Gram (SG). As shown in Figure 4, in common, the two architectures are neural networks with one hidden layer intended to represent word vectors. While the difference is that CBOW takes the context as an input (W_{t-2}, W_{t-1}, W_{t+1}, W_{t+2}) and tries to output the target word (W_t) and has high grammatical accuracy, SG does the opposite: it takes the target word as an input (W_t) and tries to produce a suitable context for this word (W_{t-2}, W_{t-1}, W_{t+1}, W_{t+2}) and has higher semantic accuracy [25].

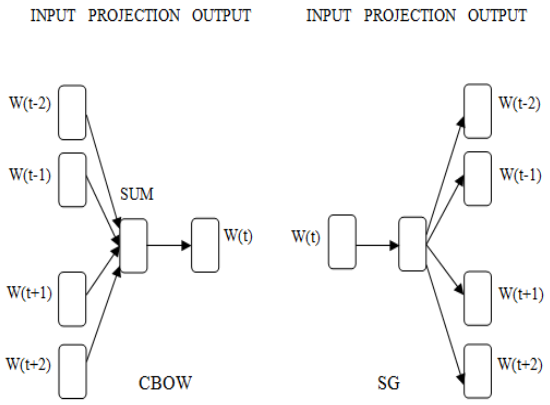

Figure 4. CBOW and Skip-Gram architectures [25].

The quality of WE can be affected by the dimension of the vector space, which ranges from one hundred to one thousand. The larger the vector, the greater the accuracy and complexity of the model in terms of calculation time. In our study, we used several models of dimensions 100 and 300 provided in the second version of AraVec by Soliman *et al*. [32]. AraVec models take three different Arabic content domains: Wikipedia, Twitter, and World Wide Web, with the two WE architectures: CBOW and SG. These models are summarized in Table 1.

Table 1. Description of different AraVec models [32].

Model	Documents	Vocabularies	Dimension
Twitter-CBOW	66,900,000	331,679	300
Twitter-SG	66,900,000	331,679	300
Twitter-CBOW	66,900,000	331,679	100
Twitter-SG	66,900,000	331,679	100
Wikipedia-CBOW	1,800,000	162,516	300
Wikipedia-SG	1,800,000	162,516	300
Wikipedia-CBOW	1,800,000	162,516	100
Wikipedia-SG	1,800,000	162,516	100
Www-CBOW	132,750,000	234,961	300
Www-SG	132,750,000	234,961	300
Www-CBOW	132,750,000	234,961	100
Www-SG	132,750,000	234,961	100

4.4. Representation of Word2vec

In NLP, handling text features requires encoding each feature as a unique dimension. In word-based attributes such as word2vec, each word is included in the dimension space and represented as a vector in that space. Suppose an Arabic text A consists of N words, $A=(W_1,W_2,...W_N)$. First, each word W_i from the text is represented by the d-dimensional feature vector V_i, and thus all text words are represented by the $d \times N$ feature matrix, where each element of the matrix corresponds to the corresponding word in the text. In order to represent the text with a one-dimensional attribute vector to serve as a single input to DL algorithms, we apply the following equation:

$$V(A) = \frac{\sum_{i=1}^{N} V_i}{N} \quad (1)$$

4.5. Classification

Neural Networks (NN) have been used in language processing in a variety of research. The simple NN consists of a connected set of neurons. Each one produces a series of real-value activations [31]. DL is a large NN composed of multiple processing layers to learn data representations with multiple levels of abstraction [31]. In this study, two DL classifiers have been considered for experiments: Convolution Neural Network (CNN) and Recurrent Neural Network (RNN). CNN [24] are Feedforward Neural Networks (FNN), this architecture makes efficient use of layers with convolving filters that are applied to local features [24]. While RNNs differ from FFNs in that they have memory [21]. A special kind of Recurrent Neural Networks (RNNs) is LSTM [21], composed of a memory cell, an input gate, an output gate and a forget gate. The cell stores a value for either long or short periods. This is achieved by using the activation function for the memory cell. In recent research, both networks have been devoted to a variety of machine learning problems. The current research uses one-dimensional CNN (1D CNN), where the kernel moves in one direction and LSTM to distingue the Quranic verse in Arabic text. The following is a complete description of the network architectures for both sub-models in our system:

4.5.1. CNN Model

For the CNN model, each input must be represented in the form of a matrix (*embedding dimension*× *vocabulary size*). We represent the embedding dimension with the dimension of the used AraVec model (300 or 100), and we represent the vocabulary size with the length of the longest verse in the Holy Quran "Ayat al-Din: 129 words". Therefore, the dimensions of the matrix are either 300×129 or 100×129. Padding with 0 is used for sentences containing less than 129 words. After that, it goes into the Conv1D layer, which consists of 128 filters to get the maximum depth concerning the available calculation possibilities, followed by a max-pooling layer with a pool size of 5 and 0.5 dropouts to avoid overfitting. Then, we take the output from the max-

pooling, flatten it, and feed it into the fully connected neural network with one dense hidden layer of 128 units to reduce intraclass classification errors. The activation function is ReLU [3]. The output layer consists of 2 softmax units to predict the class of the text. For optimization, we use Stochastic Gradient Descent (SGD) optimizer [28]. We have saved the output prediction weights to predict the testing datasets. The fit function uses number of epochs=50, batch size=10, validation split=20.

4.5.2. CNN-LSTM Model

CNN-LSTM architecture combines both convolutional layers to learn local features and LSTM layers to learn global features. Like the CNN model, our CNN-LSTM model uses an embedding layer, a 1D-Conv layer of 128 filters, and a kernel size of 5 followed by max-pooling with a pool size of 5, plus an LSTM layer with 300 or 100 units (depending on the dimension of the WE model used), followed by one dense layer of 128 units, and finally, a sigmoid activation function applied to the output of the LSTM.

5. Experiments

5.1. Evaluation Measures

To evaluate the proposed neural network models, we rely on computing each model's accuracy, precision, recall, and F1 score on test datasets. These measures range from 0 % to 100 % and are calculated as follows for each of the positive and negative categories:

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

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

$$F1-score = \frac{2\times(Precision\times Recall)}{(Precision+Recall)} \quad (4)$$

$$Accuracy = \frac{TP+TN}{P+N}\times 100 \quad (5)$$

Where means Positive (*P*); means Negative (*N*); means True Positive *(TP)*; means False Positive *(FP)*; means True Negative *(TN)* and means False Negative *(FN)*.

5.2. Experimental Results

The dataset is divided into 3 sets: 60% for training, 20% for testing and 20% for validation. Table 2 summarizes the number of elements in each set.

Table 2. Dataset division.

Class	Train	Validation	Test	Total
Verse-text	3991	998	1247	6236
Arabic-text	4080	1020	1275	6375
Total	8071	2018	2522	12611

The training set of the two models achieved an accuracy of 100% in most cases. Table 3 presents the results of the testing set in terms of precision, recall, and F1 score, and Figure 5 shows the results of the testing set in terms of accuracy considered in this study, where 0 indicates the class “Arabic-Text” and 1 indicates the class “Verse-Text”.

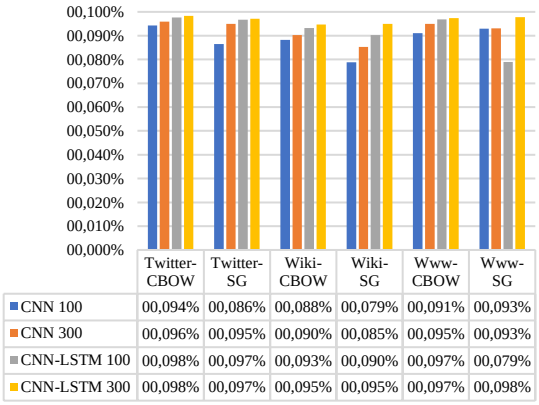

Figure 5. Test accuracy results.

Table 3. Results of the proposed models.

Model	Technique	Dim	Class	Precision	Recall	F1-score
CNN	Twitter-CBOW	300	0	93.87	95.07	94.47
			1	94.94	93.71	94.32
		100	0	90.13	96.25	93.09
			1	95.92	89.32	92.50
	Twitter-SG	300	0	93.28	97.04	95.12
			1	96.87	92.91	94.85
		100	0	76.88	99.50	86.74
			1	99.28	69.66	81.87
	Wiki-CBOW	300	0	91.43	86.12	88.69
			1	86.71	91.81	89.19
		100	0	89.23	84.05	86.56
			1	84.73	89.72	87.15
	Wiki-SG	300	0	99.56	68.11	80.88
			1	75.51	99.70	85.93
		100	0	99.82	56.00	71.75
			1	69.12	99.90	81.71
	Www-CBOW	300	0	93.91	94.19	94.05
			1	94.09	93.81	93.95
		100	0	93.67	88.87	91.21
			1	89.27	93.91	91.53
	Www-SG	300	0	86.88	98.42	92.29
			1	98.15	84.93	91.06
		100	0	86.56	98.32	92.07
			1	98.03	84.53	90.78
CNN-LSTM	Twitter-CBOW	300	0	98.07	97.58	97.83
			1	97.66	98.14	97.90
		100	0	98.22	95.76	96.98
			1	95.61	98.15	96.87
	Twitter-SG	300	0	97.29	98.67	97.97
			1	98.52	96.98	97.74
		100	0	95.79	99.24	97.48
			1	99.13	95.21	97.13
	Wiki-CBOW	300	0	99.08	92.04	95.43
			1	91.89	99.06	95.53
		100	0	99.26	89.01	93.85
			1	89.16	99.27	93.94
	Wiki-SG	300	0	98.02	94.03	95.98
			1	93.73	97.92	95.78
		100	0	99.65	81.53	89.68
			1	83.10	99.68	90.64
	Www-CBOW	300	0	99.70	96.21	97.92
			1	95.99	99.68	97.80
		100	0	99.60	95.17	97.33
			1	94.94	99.58	97.20
	Www-SG	300	0	99.21	96.21	97.69
			1	95.97	99.16	97.54
		100	0	99.19	58.42	73.53
			1	68.55	99.48	81.17

For the first model, The CNN architecture using the

CBOW technique learned using a corpus collected from Twitter with the dimension of 300 achieves the highest test accuracy of 95.87%. It also achieves the best precision without bias (93.87% for the class '0' and 94.94 for the class '1') and the best F1- score (94.47% for the class '0' and 94.32% for the class '1'). For recall, the same architecture achieved the best results (95.07% for the class '0' and 93.71 for the class '1').

For the second model, the hybrid CNN-LSTM architecture using the CBOW technique learned using a corpus collected from Twitter with the dimension of 300 achieves the highest accuracy of 98.33%. It also achieves the best precision without bias (98.07 % for the class '0' and 97.66 for the class '1') and the best F1- score (97.83% for the class '0' and 97.90% for the class '1'). For recall, the same architecture achieved the best results (97.58% for the class '0' and 98.14 for the class '1').

Table 4 presents the average accuracy, precision, recall, and f1 for the best result in CNN and CNN-LSTM architectures. Combining the CNN with the LSTM improves the accuracy of the CNN alone by 2.46%, the precision by 3.46%, the recall, and the F1 score by 3.47%. This means that this combination can make distingue verse and Arabic text 3.46% more than the CNN alone. Therefore, the CNN-LSTM hybrid model outperforms the CNN model. The results of the previous comparison revolve around two points. First, using pre-trained WE models in a fixed dimension performs better than models with classic vector representation. Besides, the combination of CNN and LSTM layers results in more feature capture, improving all metrics indicating better recognition of verses and Arabic texts.

Table 4. comparison results.

Models	Accuracy	Precision	Recall	F1 Score
CNN	95.87%	94.40%	94.39%	94.39%
CNN-LSTM	98.33%	97.86%	97.86%	97.86%

6. Conclusions and Future Work

This paper highlights the implications of using DL models to identify Quranic verses in Arabic textual content. Our models are based on DL classifiers, CNN and LSTM, and rely only on pre-trained WE based on three completely different data sources. Despite the difficulty of the Arabic language and the lack of repetition of Quranic verses, our models have achieved a satisfactory result.

From this work, we conclude that:

- DL and WE techniques are able to detect Arabic verses with an accuracy of 98.33% and 97.86% for precisions, recall and F1.
- The CNN-LSTM classifier greatly outperforms the CNN classifier. This explains the power of the CNN model to extract deep features and pass them to the LSTM model, which performs classification based on the extracted feature.
- The models built with the textual content of Twitter excel the other data sources used to learn WE models, due to its frequent use by the Arab internet pioneers and the diversity of their data contents.
- The models built with CBOW were able to represent the repeated words in the Quran better, and it achieved a better training speed than SK.
- The larger pre-trained WE vectors store more information because there are many possible cases, and the quality of word representation deteriorates whenever the vector size is less than 300.

This work will help protect Quranic content, as well as enhance the confidence of Internet users towards Quranic quotes.

In some cases, these models fail to determine the word order of the verses. In future work, we will focus on this point and try different and more complex deep learning models to improve the results. Also, we will pay attention to the problematics of diacritical verses.

Acknowledgment

This work is supported by the Algerian General Direction of Scientific Research and Technological Development (DGRSDT) and the LAMIS Laboratory.

References

[1] Abdellatif M. and Elgammal A., "Offensive Language Detection in Arabic Using Ulmfit," *in Proceedings of the 4th Workshop on Open-Source Arabic Corpora and Processing Tools, with a Shared Task on Offensive Language Detection*, Marseille, pp. 82-85, 2020.

[2] Abozinadah E., Mbaziira A., and Jr J., "Detection of Abusive Accounts with Arabic Tweets," *International Journal of Knowledge Engineering*, vol. 1, no. 2, pp. 113-119, 2015.

[3] Agarap A., "Deep Learning Using Rectified Linear Units (Relu)," *arXiv preprint arXiv:1803.08375*, 2018.

[4] Alfaifi A., Atwell E., and Hedaya I., "Arabic Learner Corpus (ALC) V2: A New Written and Spoken Corpus of Arabic Learners," *in Proceedings of Learner Corpus Studies in Asia and the World)*, Kobe, pp. 77-89, 2014.

[5] Almazrooie M., Samsudin A., Gutub A., Salleh M., Omar M., and Hassan S., "Integrity Verification for Digital Holy Quran Verses Using Cryptographic Hash Function and Compression," *Journal of King Saud University-Computer and Information Sciences*, vol. 32, no. 1, pp. 24-34, 2020.

[6] Arkok B. and Zeki A., "Classification of Quranic Topics Using Ensemble Learning," *in Proceedings of the 8th International*

Conference on Computer and Communication Engineering, Kuala Lumpur, pp. 244-248, 2021.

[7] Bengio Y., Ducharme R., Vincent P., and Jauvin C., "A Neural Probabilistic Language Model," *Journal of Machine Learning Research*, vol. 3, pp. 1137-1155, 2003.

[8] Elnagar A., Al-Debsi R., and Einea O., "Arabic Text Classification Using Deep Learning Models," *Information Processing and Management*, vol. 57, no. 1, pp. 102121, 2020.

[9] Gilkar G., Hakak S., Kamsin A., Rahman M., and Rahman M., "An Exact Matching Approach to Enhance Retrieval Process for Quranic Texts," *in Proceedings of the 4th ACM International Conference of Computing for Engineering and Sciences*, Kuala Lumpur, pp. 1-4, 2018.

[10] Goldberg Y. and Levy O., "Word2vec Explained: Deriving Mikolov Et Al.'S Negative-Sampling Word-Embedding Method," *arXiv preprint arXiv:1402.3722*, 2014.

[11] Hakak S., Kamsin A., Idris M., Gani A., Amin G., and Zerdoumi S., "Diacritical Digital Quran Authentication Model," *Pertanika Journal of Science and Technology*, vol. 25, pp. 133-142, 2017.

[12] Hakak S., Kamsin A., Palaiahnakote S., Tayan O., Idris M., and Abukhir K., "Residual-Based Approach for Authenticating Pattern of Multi-Style Diacritical Arabic Texts," *Plos One*, vol. 13, no. 6, 2018.

[13] Hakak S., Kamsin A., Shivakumara P., Gilkar G., Khan W., and Imran M., "Exact String Matching Algorithms: Survey, Issues, and Future Research Directions," *IEEE Access*, vol. 7, pp. 69614-69637, 2019.

[14] Hakak S., Kamsin A., Shivakumara P., Idris M., and Gilkar G., "A New Split Based Searching for Exact Pattern Matching for Natural Texts," *PloS One*, vol. 13, no. 7, 2018.

[15] Hakak S., Kamsin A., Shivakumara P., Tayan O., Idris M., and Gilkar G., "An Efficient Text Representation for Searching and Retrieving Classical Diacritical Arabic Text," *Procedia Computer Science*, vol. 142, pp. 150-157, 2018.

[16] Hakak S., Kamsin A., Shivakumara, P., and Idris M., "Partition-Based Pattern Matching Approach for Efficient Retrieval of Arabic Text," *Malaysian Journal of Computer Science*, vol. 31, no. 3, pp. 200-209, 2018.

[17] Hakak S., Kamsin A., Tayan O., Idris M., and Gilkar G., "Approaches for Preserving Content Integrity of Sensitive Online Arabic Content: A Survey and Research Challenges," *Information Processing and Management*, vol. 56, no. 2, pp. 367-380, 2019.

[18] Hakak S., Kamsin A., Tayan O., Idris M., Gani A., and Zerdoumi S., "Preserving Content Integrity of Digital Holy Quran: Survey and Open Challenges," *IEEE Access*, vol. 5, pp. 7305-7325, 2017.

[19] Hakak S., Kamsin A., Veri J., Ritonga R., and Herawan T., "A Framework for Authentication of Digital Quran," *in Proceedings of Information Systems Design and Intelligent Applications*, India, pp. 752-764, 2018.

[20] Hakak S., Kamsin A., Khan W., Zakari A., Imran M., Bin-Ahmad K., and Gilkar G., "Digital Hadith Authentication: Recent Advances, Open Challenges, and Future Directions," *Transactions on Emerging Telecommunications Technologies*, pp. e3977, 2020.

[21] Hochreiter S. and Schmidhuber J., "Long Short-Term Memory," *Neural Computation*, vol. 9, no. 8, pp. 1735-1780, 1997.

[22] Hussein A., Al-Kafri M., Abonamah A., and Tariq M., "Mood Detection Based on Arabic Text Documents using Machine Learning Methods," *International Journal of Advanced Trends in Computer Science and Engineering*, vol. 9, no. 4, 2020.

[23] Kamaruddin N., Kamsin A., and Hakak S., "Associated Diacritical Watermarking Approach to Protect Sensitive Arabic Digital Texts," *in AIP Conference Proceedings*, vol. 1891, no. 1, pp. 020074, 2017.

[24] LeCun Y., Bottou L., Bengio Y., and Haffner P., "Gradient-Based Learning Applied to Document Recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278-2324, 1998.

[25] Mikolov T., Chen K., Corrado G., and Dean J., "Efficient Estimation of Word Representations in Vector Space," *arXiv preprint arXiv:1301.3781*, 2013.

[26] Mikolov T., Sutskever I., Chen K., Corrado G., and Dean J., "Distributed Representations of Words and Phrases and Their Compositionality," *Advances in Neural Information Processing Systems*, vol. 26, pp. 3111-3119, 2013.

[27] Pudaruth S., Soyjaudah S., and Gunputh R., "Classification of Legislations using Deep Learning," *The International Arab Journal of Information Technology*, vol. 18, no. 5, pp. 651-662, 2021.

[28] Ruder S., "An Overview of Gradient Descent Optimization Algorithms," *arXiv preprint arXiv:1609.04747*, 2016.

[29] Sabbah T. and Selamat A., "Support Vector Machine-Based Approach for Quranic Words Detection in Online Textual Content," *in Proceedings of the 8th IEEE Malaysian Software Engineering Conference*, Langkawi, pp. 325-330, 2014.

[30] Sabbah T. and Selamat A., "A Framework for Quranic Verses Authenticity Detection in Online Forum," *in Proceedings of Taibah University*

International Conference on Advances in Information Technology for the Holy Quran and Its Sciences, Madinah, pp. 6-11, 2013.

[31] Schmidhuber J., "Deep Learning in Neural Networks: An Overview," *Neural Networks*, vol. 61, pp. 85-117, 2015.

[32] Soliman A., Eissa K., and El-Beltagy S., "Aravec: A Set of Arabic Word Embedding Models for use in Arabic Nlp," *Procedia Computer Science*, vol. 117, pp. 256-265, 2017.

[33] Touati-Hamad Z., Laouar M., and Bendib I., "Authentication of Quran Verses Sequences Using Deep Learning," *in Proceedings of the International Conference on Recent Advances in Mathematics and Informatics*, Tebessa, pp. 1-4, 2021.

[34] Touati-Hamad Z., Laouar M. R., and Bendib I., "Quran Content Representation in NLP," *in Proceedings of the 10th International Conference on Information Systems and Technologies*, Lecce, pp. 1-6, 2020.

[35] Zarrabi-Zadeh H., Tanzil-Quran Navigator, http://tanzil.net/download/, Last Visited, 2020.

[36] Zerdoumi S., Sabri A., Kamsin A., Hashem I., Gani A., Hakak S., and Chang V., "Image Pattern Recognition in Big Data: Taxonomy and Open Challenges: Survey," *Multimedia Tools and Applications*, vol. 77, no. 8, pp. 10091-10121, 2018.

Zineb Touati-Hamad received her Master's degree from Tebessa University in 2019. Presently, she is a Ph.D. Student at LAMIS Laboratory. Her research interests include Machine Learning, Natural Language Processing and Information systems.

Mohamed Ridda Laouar received his Ph.D. degree from Valenciennes University in 2005. Presently, he is a professor at Tebessa University and LAMIS Laboratory. His current research interests are Decision Making, Artificial Intelligence and Information systems.

Issam Bendib received his Ph.D. degree from Annaba University in 2018. Presently, he is an associate professor at Tebessa University. His current research interests: Information Retrieval and Machine Learning.

Saqib Hakak received his Ph.D. degree from Kuala Lumpur University in 2018. Presently, he is an associate professor at the University of New Brunswick, Canada. His current research interests: Cybersecurity, Internet of Things, Natural, Language Processing, Cloud and Edge Computing