Siamese AraBERT-LSTM Model based Approach for Arabic Paraphrase Detection

Adnen Mahmoud[1,2] **Mounir Zrigui**[1]
[1]University of Monastir, Research Laboratory in Algebra, Numbers Theory and Intelligent Systems RLANTIS, Monastir 5000, Tunisia
[2]University of Sousse, Higher Institute of Computer Science and Communication Techniques ISITCom, Hammam Sousse 4011, Tunisia
mahmoud.adnen@gmail.com mounir.zrigui@fsm.rnu.tn

Abstract

Paraphrase detection allows identifying the degree of likelihood between source and suspect sentences. It is a critical machine learning problem in computational linguistics. This is due to the expression variability and ambiguities especially in Arabic language. Previous neural models have yielded promising results, but are computationally expensive. They cannot directly align long-form sentences expressing different meanings. To address this issue, Siamese neural network is proposed for Arabic paraphrase detection based on deep contextual semantic textual similarity. Despite that the pre-trained word embedding models have advanced NLP, they ignored the contextual information and meaning within the sentence. In this paper, the potential of deep contextualized word representations was firstly investigated using Arabic Bidirectional Encoder Representation from Transformers (AraBERT) as an embedding layer. Then, Long Short Term Memory (LSTM) modeled high-level semantic knowledge. Finally, cosine distance identified the degree of semantic textual similarity. Using our own generated corpus, experiments showed that the proposed model outperformed state-of-the-art methods, in terms of F1 score.

1 Introduction

The accumulation of textual data exchanged on the web over time has increased the potential source of paraphrase (Karaoglan et al., 2016). Its identification has become increasingly challenging especially in the case of Arabic language due to its richness of ambiguous specificities (Sghaier and Zrigui, 2020). The same sentence can be reformulated in different ways using semantically similar words (Mahmoud and Zrigui, 2021a). This task allows modelling and identifying the semantic interactions between sentence pairs. It represents a challenge in the area of information retrieval and Natural Language Processing (NLP). Recently, several neural models for word embedding have been introduced like word2vec (Mikolov et al., 2014) or GloVe (Pennington et al., 2014). The produced vectors representations of sentence pair are used as inputs to measure the similarity between them. However, these models provided fixed representation for each word and did not capture its context in different sentences (Mahmoud and Zrigui, 2019a).

To deal with this drawback, contextualized word representation methods such as ELMo (Peters et al., 2018) and BERT (Babi et al., 2020) have become prevalent and received a lot of attention for obtaining sentence representations. They learnt efficiently the contextualized word representations from deep bidirectional language model pre-trained on large text corpora or via utilizing the encoder of transformer (Babi et al., 2020). Better sentence representations are captured from each generated word embedding based on its surrounding context. Therefore, we are motivated to use them for sentence modelling and Arabic paraphrase detection.

In this paper, the main objectives are focused on studying how the applications of Bidirectional Encoder Representations from Transformers (BERT) and neural networks models are suitable for semantic equivalence assessment and Arabic paraphrase detection. Indeed, the contextual features are firstly extracted using Arabic BERT (AraBERT) model. Then, the resulted embedded vectors are trained by applying deep Siamese Long Short Term Memory (LSTM) model for sequential data modelling. Finally, semantic

similarity scores are identified. To conduct experiments, paraphrased corpus is proposed preserving semantic and syntactic properties of original sentences. This paper is organized as follows: First, state of the art on paraphrase detection is presented in section 2. Then, the proposed approach is detailed in section 3. Subsequently, the experiments are described in section 4. Finally, we end by a conclusion and future work in section 5.

2 State of the Art

Although processing language and comprehending the contextual meaning is an extremely complex task, paraphrase detection is a sensitive field of research for specific language (Mahmoud and Zrigui, 2019b). Following the literature, numerous neural models were proposed for modelling semantic similarity among sentence pair. They have gained promising results in major NLP tasks distinguish supervised and unsupervised approaches.

Unsupervised methods use pre-trained word/phrase embeddings directly for the similarity task without training a neural network model on them which supervised ones do (Aliane and Aliane, 2020).

Word2vec is one of the most popular unsupervised methods. It generates words embeddings according to their semantics in the sentence. Some researches were already used it to detect similarity between texts. Veisi et al. (2022) employed word2vec and cosine distance for Persian text similarity detection. The same approach was proposed by Gharavi et al. (2016). Indeed, the generated word vectors from word2vec model were averaged to produce sentence vector. Then, the Jaccard coefficient was used to report plagiarism cases. For Arabic language, Nagoudi et al. (2018) detected verbatim and complex reproductions using fingerprinting and word embedding. Next, different NLP techniques were combined like word alignments, Inverse Document Frequencies (IDF) and Part-Of-Speech (POS) weighting to identify the most descriptive words in each textual unit. Similarly, Yalcin et al. (2022) indexed each document according to the grammatical classes of their n-grams. The aim was to access rapidly to sentences that were possible plagiarism candidates. Then, word2vec and Longest Common Subsequence (LCS) were combined for measuring semantic similarity.

Unlike word2vec, GloVe model does not rely on local context information, but incorporates global statistics through word co-occurrences to obtain word vectors (Mahmoud and Zrigui, 2021b). For English text similarity identification, the superiority of GloVe compared to word2vec was demonstrated by Hindocha et al. (2019) and Mohammed et al. (2019).

Other works adopted FastText model. For example, Iqbal et al. (2021) investigated several word embedding techniques (word2vec, GloVe, and FastText) for Bengali semantic similarity. Experiments showed that FastText with cosine distance were the most suitable for this task.

Recently, context-depending embedding techniques have been introduced. They used transformers and long and short-term memory techniques to convert a word into n-dimensional vector. To enhance unsupervised sentence similarity methods, Ranashinghe et al. (2019) enhanced various context-based models: Embedding from Language Models (ELMO) (Peters et al., 2018), Bidirectional Encoder Representations from Transformers BERT (Babi et al., 2020), Flair (Akbik et al., 2019) and stacked embedding on different datasets: English, Spanish and Biomedical. The best experimental results were obtained with stacked embeddings of ELMO and BERT.

On the *supervised methods* side, deep pairwise fine-grained similarity network is based on Siamese architecture. Two identical neural networks sharing the same weights. The resulted output vectors are fed to a join function for paraphrase prediction (Mahmoud and Zrigui, 2021c).

Convolutional Neural Networks (CNN) were efficient to extract the most descriptive n-grams of different semantics through convolution and pooling layers. This model was applied for sentence modeling and semantic text similarity computation as shown by Shao (2017) and He and Lin (2015) for English, and Mahmoud and Zrigui (2017) for Arabic. Recurrent neural networks have recently shown promising results for analysing sequential data and modelling long-term dependencies within sentence (Haffar et al., 2021). For modelling context and structure of sentence, Tai et al. (2015) proposed tree-structured LSTM model while Liu et al. (2019) introduced a Multi-Layer Bidirectional LSTM (Bi-LSTM) and TreeLSTM model. As shown in (Hambi and Benabbou, 2019), LSTM model learnt from the output of doc2vec while CNN model learnt thereafter the most relevant features

from documents for plagiarism detection. As described in (Othman et al., 2021), LSTM model was augmented with an attention mechanism to extract the most representative words within questions. Then, CNN retrieved relevant questions. Hamza et al. (2020) extracted semantic and syntactic relations between words using ELMo model. Then, different deep neural models were analyzed such as simple models, CNN and RNN (Bi-LSTM, GRU) mergers models, and ensemble models. Then, a concatenation operation was used as a merge operation and Softmax function for Arabic similar questions classification. Recently, Meshram and Kumar (2021) demonstrated that deep contextualized word representations using BERT model became a better way for feature extraction from sentences. LSTM was thereafter applied for high level features knowledge. Then, Manhattan distance was used for similarity identification.

Following the literature, our approach combines the power of unsupervised learning through contextualized word embedding and supervised deep pairwise similarity network that outperformed state-of-the-art. Our motivation is to use them for sentence modelling and paraphrase identification in Arabic. This is due to the rich language of features that made its processing difficult than other languages (Meddeb et al. 2021). It is non-vocalized, non-concatenative, homographic, agglutinative and derivational, which needs deep understanding of textual components (Haffar et al., 2020)

3 Proposed Approach

In this section, deep contextual embedding based similarity approach is presented briefly for Arabic paraphrase detection. It is based on a Siamese neural architecture. It has proven its relevance for learning sentence semantic comparability through two identical sub-networks that are fit for handling sentence pair and likeness measure. The phases constituting the proposed architecture are described in Fig. 1:

1) Firstly, text features are extracted using the Arabic Bidirectional Encoder Representations from Transformers (AraBERT) process.
2) The embedded vectors are subsequently trained by using Long Short Term Memory (LSTM) recurrent neural network model.
3) Finally, similarity scores are determined for each sentence pair, and the semantic textual similarity is learned.

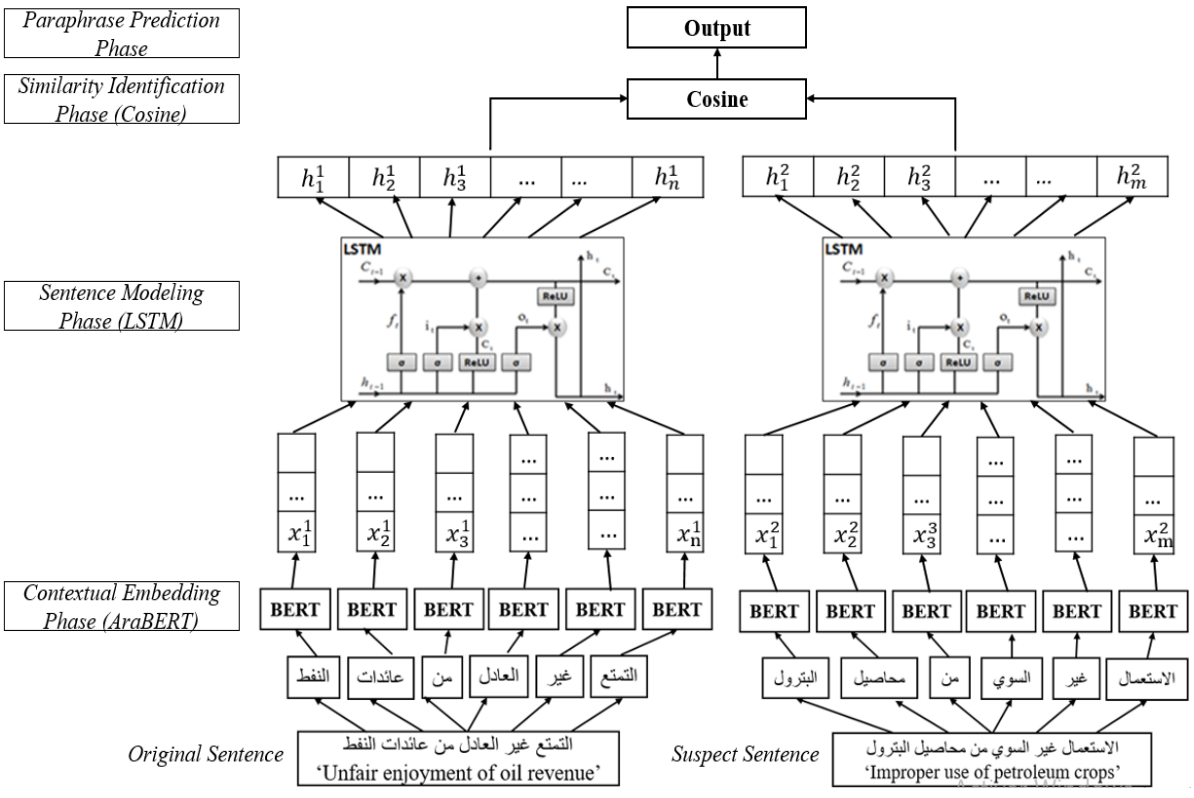

Fig. 1: Proposed architecture.

3.1 Arabic BERT (AraBERT) Embedding Phase

The use of word embeddings allows to effectively detect the syntactic and semantic similarities between words. In this work, we resorted to the AraBERT model which outperformed previous language models like word2vec, GloVe, etc. It addressed effectively ambiguity in which multiple vector representations could be extracted for the same word based on its context. More precisely, AraBERT is a bidirectional transformer for generating sentence representation by learning the context of each word on all of its surroundings in the sentence. Because AraBERT uses sub-words as a unit instead of words, the source S and target T sentences are tokenized into words. Then, the obtained tokens are thereafter encoded by the AraBERT model. For the input sequence of N tokens $\{w_1, \dots, w_n\}$, we obtain the final hidden states $\{h_1, \dots, h_n\}$ representing the output of the transformer as denoted in Eq. (1) :

$$h_i = AraBERT(w_1, \dots, w_n) \quad (1)$$

To generate the representations of source h_s and target h_t sentences, we use the mean_pooling process applied on the outputs of AraBERT model. It consists computing the means over each vector dimension as follows in Eq. (2) and Eq. (3):

$$h_s = mean\_pooling(h_1, \dots, h_m) \quad (2)$$

$$h_t = mean\_pooling(h'_1, \dots, h'_n) \quad (3)$$

Where: m and n are the lengths of source and target sentences.

3.2 Sentence Modeling Phase

The major reason for relying on Long-Short Term Memory (LSTM) recurrent neural network is its proven performance to capture short and long-term dependencies, model variable-length of sequential data and prevent the vanishing gradient problem of RNN. LSTM is characterized by a memory cell that is capable of maintaining its state over time, and internal mechanisms called gates to regulate the information flow.

Given the input vector x_t, hidden sate h_t and memory state c_t, the updates in LSTM are performed as denoted in Eqs. (4, 5, 6, 7, 8 and 9):

$$i_t = ReLU(W_i.x_t + U_i h_{t-1} + b_i) \quad (4)$$

$$f_t = ReLU(W_f.x_t + U_f h_{t-1} + b_f) \quad (5)$$

$$\check{c}_t = ReLU(W_c.x_t + U_c h_{t-1} + b_c) \quad (6)$$

$$c_t = i_t \odot \check{c}_t + f_t \odot c_{t-1} \quad (7)$$

$$o_t = ReLU(W_o.x_t + U_o h_{t-1} + b_o) \quad (8)$$

$$h_t = o_t \odot Tanh(c_t) \quad (9)$$

Where : i_t, f_t, c_t, o_t are input, forget, memory and output gates at time t ; $W_i, W_f, W_c, W_o, U_i, U_f, U_c, U_o$ are weight matrices ; b_i, b_f, b_c, b_o are the bias vectors ; ReLU (Rectified Linear Unit) is the activation function and $\odot$ denotes the Hadamart product of matrices.

3.3 Similarity Identification Phase

Once we have the vectors (A, B) that capture the underlying meaning of sentence pair, the semantic similarity is computed using Cosine similarity measure as defined in Eq. (10):

$$Sim(A,B) = \frac{A.B}{||A||.||B||} = \frac{\sum_{i=1}^{N} A_i B_i}{\sqrt{\sum_{i=1}^{N} A_i^2}\sqrt{\sum_{i=1}^{N} B_i^2}} \quad (10)$$

Where: A_i and B_i are the components of the vectors A and B, respectively.

For prediction, the obtained scores of $Sim(A,B)$ are converted into probabilities $P \in [0,1]$ as defined in Eq. (11). To decide whether or not A and B are paraphrased (i.e., semantically equivalent), the obtained P is compared to a threshold $\alpha = 0.25$:

$$P = \frac{Sim(A,B)}{10} = \frac{Sim(A,B) \times 5}{100} + \frac{50}{100} \quad (11)$$

4 Experiments and Discussion

4.1 Dataset

To tackle the lack of publicly available paraphrased corpora in Arabic, we intend to develop our own corpus. The main idea is to capture more semantic features from sentence pairs. This is done by combining word2vec model and POS weighting for better-paraphrased sentence generation. It consists of the following operations:

Data collection. The Open Source Arabic Corpora (OSAC) is used as a source corpus from which passages of texts are extracted and replaced semantically. To do this, vocabulary model is proposed from which original words are replaced. It is collected from various resources (i.e. Arabic Corpora Resource (AraCorpus), King Saud University Corpus of Classical Arabic (KSUCCA) and a set of Arabic papers from Wikipedia) including more than 2.3 billion words.

Data preprocessing. To remove worthless data and reduce thereafter the time required for

further processing, preprocessing operations are applied:

(1) Unnecessary data (e.g. extra white spaces, titles numeration, non-Arabic words) are removed.
(2) Some writing forms are normalized (e.g., *Hamza* "أ" and *Taa Marboutah* "ة" to "ا" and "ه".)).
(3) Sentences are tokenized into words to reduce lexical parsimony.

Paraphrased corpus generation. Semi-artificial approach is proposed as follows:

(1) Given a random variable $a \leq x \leq b$, the degree of paraphrase D is defined by applying a random uniform function, as denoted in Eq. (12):

$$f(x) = \begin{cases} \frac{1}{b-a}, a \leq x \leq b \text{ and } (b-a) \in [1.33, \dots, 2.22] \\ 0, \text{ otherwise} \end{cases} \quad (12)$$

(2) The number of words to replace P is defined from the OSAC source corpus of size N. It is defined according to $D \in [45\%, \dots, 75\%]$, as follows in Eq. (13):

$$P = N \times D \quad (13)$$

(3) To replace source words according to an index chosen randomly, random shuffle function is used.
(4) Since paraphrase allows replacing the meaning of original words semantically, synonyms are identified from the vocabulary using Skip gram model. It predicts the context of middle word according to the unique word representation in a surrounding window as input. The original word is replaced by its most similar one that has the same grammatical class from the created vocabulary. In this way, original and paraphrased sentences will have the same syntactic changes with similar words semantically. The combination of word2vec and POS is reported to be good in capturing syntactic and semantic features of words.
(5) Human judgments are used for final corpus validation.

Table 1 summarizes some obfuscations forms created through our proposed approach:

Original sentence	التمتع غير العادل من عائدات النفط. 'Unfair enjoyment of oil revenue.'	
w_i	Obfuscation forms	
	Synonym substitution	Add / remove of words
التمتع *'enjoyment'*	الاستعمال 'use'	الاستعمال 'use'
غير *'not'*	غير 'not'	-
العادل *'fair'*	السوي 'proper'	الخاطئ 'wrong'
من *'of'*	من 'of'	ل 'of'
عائدات *'revenue'*	محاصيل 'crops'	محاصيل 'crops'
النفط *'oil'*	البترول 'petroleum'	البترول 'petroleum'
Suspect sentences	الاستعمال غير السوي من محاصيل البترول 'Improper use of petroleum crops'	الاستعمال الخاطئ لمحاصيل البترول 'Misuse of petroleum crops'

Table 2: Examples of Paraphrased sentences in Arabic

Table 2 describes the data used for training and testing the proposed approach:

Corpora	Models	Total pairs	Paraphrased pairs	Original pairs
OSAC	Train	3,600	2,400	1,200
	Test	1,500	1,000	500
Semeval	Test	250	196	94

Table 2: Experimental dataset

4.2 Neural Models Configuration

Word2vec. Table 3 illustrates the parameters of word2vec algorithm that are useful to capture the right synonyms of the target word and increase the quality of Arabic paraphrased sentence generation:

Parameters	Values
Vocabulary size	More than 2.3 billion words
Vector dimension	300
Window size	3
Minimum count	≤ 5
Workers	8
Epochs	7

Table 3: Parameters of word2vec model

AraBERT[1]**.** This model is employed for sentence modelling. As shown in Table 4, the parameters (i.e., number of epochs, hidden layers, attention heads, and hidden size) are fixed according to the dataset and the memory reserved. The overall model is trained by Adam optimizer.

Parameters	Values
Hidden layers	4
Attention heads	4
Hidden size	256
Dropout	0.1
Optimizer	Adam
Epochs	50
Activation function	ReLU

Table 4: Parameters of AraBERT model

LSTM. For sentence modeling, the parameters that increased the performance of the proposed LSTM model are summarized in Table 5:

Parameters	Values
Hidden units number	256
Activation function	ReLU
Loss probability	0.2
Optimizer	Adam
Batch size	100

Table 5: Parameters of LSTM model

4.3 Evaluation Metrics

F1 score is a measure of the proposed model's accuracy on the generated dataset. It is defined as the harmonic mean of the precision and recall ranging in [0, 1], as defined in Eq. (14) (Mahmoud and Zrigui, 2021d):

$$2 \times \frac{P \times R}{P + R} \tag{14}$$

Where: precision P is the fraction of true positive examples among the ones that the model classified as positive; recall R is the fraction of examples classified as positive among the total number of positive examples.

4.4 Discussion

The performances of our approach using the proposed corpus and SemEval dataset are comparable to detect semantic meaning of words, which also depends on deep contextual embedding. Experimental results are shown in Table 6:

Corpora	Models	F1
OSAC	GloVe-Cosine	0.7750
	AraBERT-Cosine	0.8050
	GloVe-LSTM	0.8731
	AraBERT-LSTM	0.8975
SemEval	GloVe-Cosine	0.7650
	AraBERT-Cosine	0.7850
	GloVe-LSTM	0.8397
	AraBERT-LSTM	0.8650

Table 6: Experimental results

Experiments demonstrated that GloVe model with cosine similarity achieved the lowest F1 score (77.5% with OSAC and 76.5% with SemEval). This is due to the fact that GloVe worked on word level and cannot cope with Arabic language morphology. Indeed, different senses of the words are combined into one vector which can result a confused representation of the ambiguous Arabic language. As a solution, we proposed to integrate it topped with LSTM layer. It increased the number of learnable weights and paraphrase prediction layer. It captured efficiently sentence semantics better. As expected, adding a pairwise sentence similarity sub-networks improved the performance of our model achieving the best F1 scores: 87.31% using OSAC and 83.97% using SemEval.

Instead of using GloVe model, the impact of contextualized word embedding AraBERT is examined. It consistently improved the results in all the proposed models. While Arabic language is highly derivational language mutating many morphological variations of each word, AraBERT model was able to generate vector representation of any arbitrary words and was not limited to the vocabulary space. It obtained the highest F1 scores: 89.75% with OSAC and 86.50% with SemEval.

Furthermore, we can notice that low performances are obtained when using SemEval corpus compared to the proposed OSAC corpus. Based on the complex nature and structure of Arabic language, this limit is related to the nature of the corpus, its language and the proposed paraphrase detection approach: Compared to the OSAC corpus, SemEval corpus has a relatively small vocabulary and training examples. That's why, the performance of AraBERT based approach hasn't been affected. We can deduce

[1] https://github.com/aub-mind/arabert/tree/master/arabert

also the effectiveness of the model when facing small amount of data and more generally its suitability when dealing with low resources languages like Arabic.

As depicted in Table 7 and Fig. 2, final experimental results are competitive with those obtained with state-of-the-art methods achieving the best F1 score with AraBERT-LSTM model. It was efficient for text similarity prediction and paraphrase detection:

References	Corpora	Models	F1
Ayata et al. (2017)	SemEval English	Word2vec-LSTM	0.587
Peinelt et al. (2020)	SemEval English	tBERT-LDA	0.524
	MSRP English	tBERT-LDA	0.884
Meshram and Kumar (2021)	SICK, STS, clinical dataset	Word2vec-BERT	0.889
		GloVe-BERT	0.864
		BERT-BERT	0.896
Ours	OSAC Arabic	AraBERT-LSTM	0.897
	SemEval Arabic	AraBERT-LSTM	0.865

Table 7: Comparison with state-of-the-art methods

For sentiment analysis and Tweets similarity, word2vec algorithm was useful for features representation while LSTM model was thereafter efficient for avoiding the long-term dependency problem, as shown by Ayata et al. (2017). Recently, Meshram and Kumar (2021) and Peinelt et al. (2020) approved that the use of BERT model as a contextualized word embedding was even applying it on topics for English text similarity. Compared to the state-of-the-methods, the overall best performing method was obtained using AraBERT for contextualized word embedding in Arabic, LSTM for sentence pair modelling and cosine for similarity prediction as demonstrated in our proposed final approach. It achieved the highest F1 scores (89.75% using OSAC and 86.50% using SemEval).

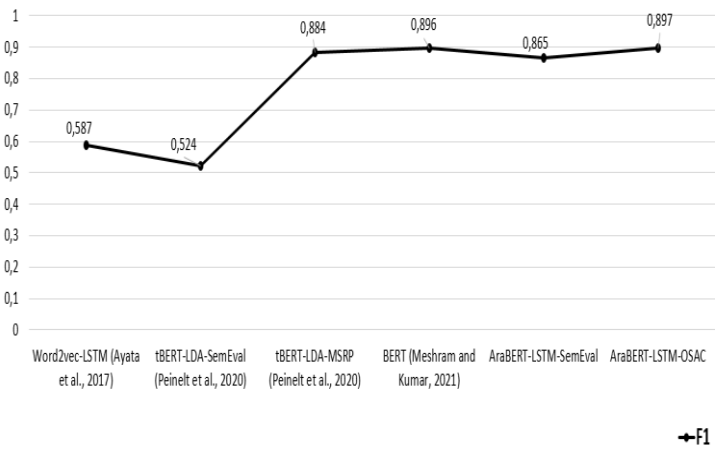

Fig. 2: Comparison with state-of-the-art methods

5 Conclusion and Future Work

In this paper, we introduced an Arabic paraphrased corpus preserving semantic and syntactic features of sentences. Original words are replaced by their most similar ones that had the same part-of-speech from a vocabulary. The created corpus included various forms of obfuscation like same polarity, add/deletion of words, etc. Then, we studied how this corpus could be useful efficiently in the evaluation of Arabic paraphrase detection using deep contextual word embeddings. AraBERT-LSTM based approach outperformed significantly state-of-the-art methods. It alleviated data sparsity and Arabic word semantics achieving the following F1 scores: 89,75% with OSAC and 86.50% SemEval. For future research, one possible way to further improve our system could be to feed AraBERT embedding to other recurrent neural networks like Bi-directional LSTM (Bi-LSTM), or Gated Recurrent Units (Bi-GRU).

References

Akbik A., Bergmann T., Blythe D., Rasul K., Schweter S., and Vollgraf R. 2019. An easy-to-use framework for state-of-the-art NLP, Conference of the North American Chapter of the Association for Computational Linguistics (demonstrations).

Aliane A. A., and Aliane H. 2020. Evaluating SIAMESE architecture neural models for Arabic textual similarity and plagiarism detection, 4th International Symposium on Informatics and its Applications (ISIA), M'sila, Algeria: 1-8.

Ayata D., Saraclar M., and Ozgur A. 2017. BUSEM at SemEval-2017 Task 4 sentiment analysis with word embedding and long short term memory

RNN approaches, 11th International Workshop on Semantic Evaluations (SemEval-2017), Vancouver, Canada: 777–783.

Babi K., Martincic-Ipsi S., and Meštrovic A. 2020. Survey of neural text representation models, Information, volume.11: 1-32.

Gharavi E., Bijari K., Zahirnia K., and Veisi H. 2016. A deep learning approach to Persian plagiarism detection, FIRE (Working Notes): 1-6.

Haffar N., Hkiri E., and Zrigui M. 2020. Enrichment of Arabic TimeML corpus, International Conference on Computational Collective Intelligence (ICCCI), Da Nang, Vietnam: 655–667.

Haffar N., Ayadi R., Hkiri E., and Zrigui M. 2021. Temporal ordering of events via deep neural networks, 16th International Conference on Document Analysis and Recognition (ICDAR), Lausanne, Switzerland: 762-777.

Hambi M., and Benabbou F. 2019. A new online plagiarism detection system based on deep learning, International Journal of Advanced Computer Science and Applications (IJACSA), 11(9): 470-478.

Hamza A., En-Nahnahi N., and Ouatik S. E. 2020. Contextual word representation and deep neural networks-based method for Arabic question classification, Advances in Science, Technology and Engineering Systems Journal, 5(5): 478-484.

He H., Gimpel K., and Lin J. 2015. Multi-perspective sentence similarity modeling with convolutional neural networks, Conference on Empirical Methods in Natural Language Processing (EMNLP): 1576–1586.

Hindocha E., Yazhiny V., Arunkumar A., and Boobalan P. 2019. Short-text semantic similarity using GloVe word embedding, International Research Journal of Engineering and Technology (IRJET), volume. 6: 553-558.

Iqbal A., Sharif O., Hoque M. M., and Sarker I. H. 2021. Word embedding based textual semantic similarity measure in Bengali, Procedia Computer Science, volume.193: 92–101.

Karaoglan D., Kisla T., and Metin S. K. 2016. Description of Turkish paraphrase corpus structure and generation method, International Conference on Computational Linguistics and Intelligent Text Processing (CICLing), Turkey: 208-217.

Liu L., Yang W., Rao J., Tang R., and Lin J. 2019. Incorporating contextual and syntactic structures improves semantic similarity modeling, Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP): 1204–1209.

Mahmoud A., and Zrigui M. 2017. Semantic similarity analysis for paraphrase identification in Arabic texts, 31st Pacific Asia Conference on Language, Information and Computation (PACLIC), Philippine: 274-281

Mahmoud A., and Zrigui M. 2019a. Sentence embedding and convolutional neural network for semantic textual similarity detection in Arabic language, Arabian for Engineering and Science Journal, volume. 44: 9263-9274.

Mahmoud A., and Zrigui M. 2019b. Similar meaning analysis for original documents identification in Arabic language, International Conference on Computational Collective Intelligence (ICCCI), Hendaye, France: 193–206.

Mahmoud A., and Zrigui M. 2021a. Arabic semantic textual similarity identification based on convolutional gated recurrent units, International Symposium on INnovations in Intelligent Systems and Applications (INISTA), Kocaeli, Turkey: 1-7.

Mahmoud A., and Zrigui M. 2021b. Hybrid attention-based approach for Arabic paraphrase detection, Applied Artificial Intelligence: 1-16.

Mahmoud A., and Zrigui M. 2021c. Semantic similarity analysis for corpus development and paraphrase detection in Arabic, International Arab Journal of Information Technology (IAJIT), volume. 18: 1-7.

Mahmoud A., and Zrigui M. 2021d. BLSTM-API: Bi-LSTM recurrent neural-based approach for Arabic paraphrase identification, Arabian for Science and Engineering, volume. 46: 4163-4174.

Meddeb O., Maraoui M., and Zrigui M. 2021. Arabic text documents recommendation using joint deep representations learning, Procedia Computer Science, 192(1): 812-821.

Meshram S., and Kumar M. A. 2021. Long short-term memory network for learning sentences similarity using deep contextual embeddings, International Journal of Information Technology, 13(4): 1633–1641.

Mikolov T., Sutskever I., Corrado G., and Dean J. 2014. Distributed representations of words and phrases and their compositionality, arXiv:1405.4053.

Mohammed S. M., Jacksi K., and Zeebaree S. R. M. 2019. A state-of-the-art survey on semantic similarity for document clustering using GloVe and density-based algorithms, Indonesian Journal of

Electrical Engineering and Computer Science, 22(1): 552-562.

Nagoudi E. B. A., Khorsi H., Cherroun H., and Schwab D. 2018. A two-level plagiarism detection system for Arabic documents, Cybernetics and Information Technologies, In press, 18(1): 1-17.

Othman N., Faiz R., and Smaïli K. 2021. Learning English and Arabic question similarity with Siamese neural networks in community question answering services, Data and Knowledge Engineering (In press): 1-26.

Peinelt N., Nguyen D., and Liakata M. 2020. tBERT: Topic models and BERT joining forces for semantic similarity detection, 58th Annual Meeting of the Association for Computational Linguistics: 7047–7055.

Pennington J., Socher R., and Manning C. 2014. GloVE: Global vectors for word representation, Conference on Empirical Methods in Natural Language Processing (EMNLP), Doha, Qatar: 1532-1543.

Peters M. E., Neumann M., Iyyer M., Gardner M., Clark C., Lee K., and Zettlemoyer L. 2018. Deep contextualized word representations, Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, New Orleans, Louisiana: 2227-2237.

Ranashinghe T., Orasan O., and Mitkov R. 2019. Enhancing unsupervised sentence similarity methods with deep contextualized word representations, International Conference on Recent Advances in Natural Language Procesing (RANLP), Varna, Bulgaria: 994-1003.

Sghaier M. A., and Zrigui M. 2020. Rule-based machine translation from Tunisian dialect to modern Arabic standard, Procedia Computer Science, volume. 196: 310-319.

Shao Y. 2017. Hcti at semeval-2017 task 1: use convolutional neural network to evaluate semantic textual similarity, 11th International Workshop on Semantic Evaluation (SemEval-2017): 130– 133.

Tai K. S., Socher R., and Manning C. D. 2015. Improved semantic representations from tree-structured long short-term memory networks, arXiv preprint arXiv:1503.00075.

Veisi H., Golchinpour M., Salehi M., and Gharavi E. 2022. Multi-level text document similarity estimation and its application for plagiarism detection, Iran Journal of Computer Science: 1-13.

Yalcin K., Cicekli I., and Ercan G. 2022. An external plagiarism detection system based on part-of-speech (POS) tag n-grams and word embedding, Expert Systems with Applications, volume. 197.