International Journal of Electrical and Computer Engineering (IJECE)
Vol. 11, No. 4, August 2021, pp. 3519~3528
ISSN: 2088-8708, DOI: 10.11591/ijece.v11i4.pp3519-3528

Using deep learning models for learning semantic text similarity of Arabic questions

Mahmoud Hammad, Mohammed Al-Smadi, Qanita Bani Baker, Sa'ad A. Al-Zboon
College of Computer and Information Technology, Jordan University of Science and Technology, Irbid, Jordan

Article Info

Article history:

Received Jul 22, 2020
Revised Dec 9, 2020
Accepted Jan 13, 2021

Keywords:

Arabic dataset
BERT
Deep learning
Machine learning
Semantic text similarity

ABSTRACT

Question-answering platforms serve millions of users seeking knowledge and solutions for their daily life problems. However, many knowledge seekers are facing the challenge to find the right answer among similar answered questions and writer's responding to asked questions feel like they need to repeat answers many times for similar questions. This research aims at tackling the problem of learning the semantic text similarity among different asked questions by using deep learning. Three models are implemented to address the aforementioned problem: i) a supervised-machine learning model using XGBoost trained with pre-defined features, ii) an adapted Siamese-based deep learning recurrent architecture trained with pre-defined features, and iii) a pre-trained deep bidirectional transformer based on BERT model. Proposed models were evaluated using a reference Arabic dataset from the mawdoo3.com company. Evaluation results show that the BERT-based model outperforms the other two models with an F1=92.99%, whereas the Siamese-based model comes in the second place with F1=89.048%, and finally, the XGBoost as a baseline model achieved the lowest result of F1=86.086%.

Corresponding Author:

Mahmoud Hammad
College of Computer and Information Technology
Jordan University of Science and Technology
Irbid, Jordan 22110
Email: m-hammad@just.edu.jo

1. INTRODUCTION

Semantic text similarity (STS) is a measurement used to determine how linguistic terms are equivalent to each other. Linguistic terms that usually studied are documents, sentences, words, and questions [1]. The STS improves the understanding of the semantic similarity between linguistic terms and increases the accuracy of several knowledge-based applications. This understanding gives the STS a great impact on many applications in artificial intelligence and the computational linguistics such as information retrieval, word sense disambiguation, knowledge acquisition, and natural language processing (NLP) [2, 3].

Regarding to the NLP field, STS applications can be extend from paraphrase identification and question similarity to machine translation [4]. The research in the STS has been greatly increased during the past few years, most of them driven by the annual SemEval competitions [5]. SemEval is an international workshop for semantic evaluation driven by the SIGLEX [1].

With the advent of Web 2.0 and social computing advancements, platforms for question answering has become widely used. According to the Quora, a well-known platform for question answering and knowledge sharing, over 10 million users visit Quora every month. With such a large number of visitors, similar questions definitely would be asked and answered by several users, hence confusing other users in

finding the right answer and causing the writers to feel like they have to write the same answer several times responding to similar questions.

Two questions asking about the same thing can be formed using a different set of vocabulary and syn-tactic structure. This makes detecting the semantic similarity between the questions is a challenging task. This research proposes three models for analyzing the semantic similarity of Arabic question pairs; i) a supervised-machine learning model using XGBoost [6] trained with pre-defined features, ii) an adapted Siamese deep learning recurrent architecture based on the work of [7], and iii) a pre-trained deep bidirectional transformer based on BERT model [8].

This paper presents several new non-trivial extensions to our preliminary work described in [9]:

- Our preliminary work [9] contains only traditional machine learning models such as XGBoost, SVM, and decision tree. In this manuscript, we have designed and implemented various deep learning models using transfer learning technique.
- We enlarged the dataset used for training and testing. In [9], we trained our models on 9,568 pairs of questions whereas in this paper, we trained our models on 15,712 pairs of questions, i.e., 31,424 distinct questions.
- Similar to our preliminary work in [9], two of our models trained using pre-engineered features including character-level features, word-level features, morphological features, semantic features, and word embedding features. Unlike our preliminary work, our best-achieved model, the BERT-based model, was able to learn the semantic similarity among pair of Arabic questions without the need for pre-engineered features. Hence, increasing the generality and the applicability of our approach.
- On top of the previous technical contributions, we discussed our work in light of other related research effort in the area of Arabic text similarity detection using deep learning. Moreover, the paper provides detailed description of the models along with the used parameters for training our models to give the best results.

The rest of this paper is organized as follows: Section 2. presents a brief survey of the literature for STS Then, section 3. describes our method for detecting similar Arabic questions. Section 4. presents the results of our intensive experiments. In section 5, results are analyzed and discussed. Finally, section 6. concludes the paper with avenue of future work.

2. RELATED WORK

Many researchers from various fields utilized semantic text similarity (STS) on different applications. This section compares and contrasts our research contribution in light of other research work in the field. Our work is related to the research body that applied machine learning and deep learning techniques to solve STS problems including [10-26]. However, all of the previously mentioned approaches designed their STS models for English language texts. Even though some of their models can be applied to Arabic texts, they will not produce high accuracy since their models are not designed nor trained on Arabic text. Therefore, these approaches cannot solve the problem we are trying to solve, that is accurately and efficiently detecting similar Arabic questions.

Although the majority of the researchers in the STS field developed techniques for the English languages, few of them developed STS approaches for the Arabic language. Next we discuss the main research efforts for detecting the semantic similarity of Arabic texts. Mohammad *et al.* [27] proposed an enhanced approach for paraphrase identification (PI) and STS in Arabic tweets. Sagher *et al.* [28] proposed a CNN deep learning model to classify Arabic sentences into three categories. [29] used and compared different STS methods to measure the cross-language semantic similarity for short sentences and phrases. Various approaches used STS to detect plagiarism in Arabic texts such as [30-32]. Ferrero *et al.* [33] proposed two different approaches to measure the STS of cross-language sentences for Arabic-English text. Moreover, [34] proposed a query-based Arabic text summarization approach that accepts Arabic document as well as user queries. Finally, [35] adopted morphological word2vec method for Neural machine translation for low resource settings

3. RESEARCH METHOD

This section describes our method in developing a machine-learning approach for accurately and efficiently detecting if two Arabic questions are similar or not.

3.1. Arabic questions pair's dataset

In order to evaluate our models, the Arabic questions pairs dataset provided by *mawdoo3.com* company is used. The dataset was manually annotated by mawdoo3's data annotation team. As shown in Table 1, the dataset consists of around 15k pairs of Arabic questions annotated as "similar" or "not". The data was divided into two files, training data with 11.997 pairs of questions and testing file with 3.715 pairs of questions.

Table 2 shows an example of two pairs of questions selected from the dataset and representing the two main categories: similar shown as "Yes" or not similar shown as "No". In the first row of the Table 2, Question1 asks about the birth city of the comprehensive thinker, Al-Razi, and Question2 asks about the city of Al-Razi museum. Clearly, those two questions are not similar since they are asking about two different things. On the other hand, Question1 and Question2 in the second row of the Table 2 are asking about the first country where communism political ideology was started. Those two questions were written in two different ways but they still have a similar meaning.

The utilized dataset is nearly balanced with 55.01% labeled as "not similar" and the rest, 44.99%, labeled as "similar" In order to further analyze the dataset, we computed the common words among each question pair. Figure 1 shows that the number of overlapping words between pair of "similar" and "not similar" questions is around 2 words. Therefore, relying on the overlapped words between pair of questions to know if they are similar will not give good results. Thus, Figure 1 shows that our problem is very challenging to solve.

Table 1. Number of instances in the dataset

Train dataset	Test dataset	Total
11,997	3,715	15,712

Table 2. An example of two instances, two pairs of questions, from the two classes

Question1	Question2	Label
في أي مدينة يقع متحف الرازي؟	أين ولد الرازي	No
في أي دولة ظهرت الشيوعية؟	أين بدأت الشيوعية؟	Yes

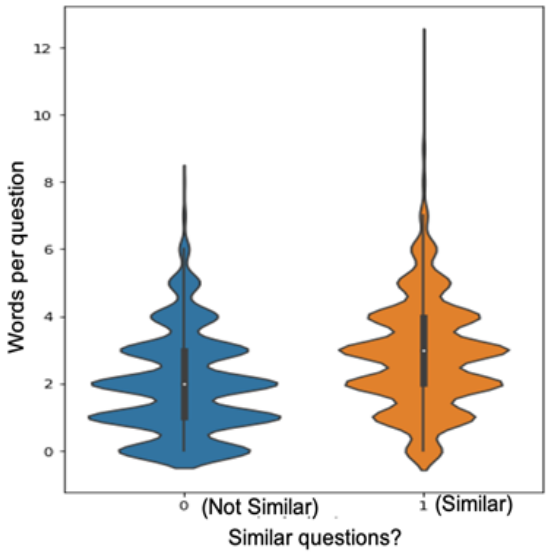

Figure 1. The distribution of common words in similar and non-similar questions for the training dataset

3.2. Data pre-processing

In order to prepare the dataset for further processing and to enhance the accuracy and reduce the noise in the data, various Arabic pre-processing steps were applied such as:

- Removal of non-Arabic words.
- Removal of hyperlinks and hashtags in all posts.
- Removal of Arabic diacritics such as
- Removal of punctuation and symbols such as "?, (,), , ' ! @ $ # —".
- Normalization, which is used to remove "HAMZA/ " from the "ALEF/ " (i.e., the replaced with the abstract version of the letter (((( ))).
- Removal of Arabic stop words.

The NLTK library [36] written in python was used to implement the data pre-processing and data cleaning phase.

3.3. Features extraction

As mentioned earlier two out of three models developed in this research were trained with pre-engineered features. After the data was cleaned, the following features were extracted:

- Character level features: This set of features includes: the total number of characters for the pair of questions, the number of different characters among the question pairs, the ratio of the different characters, the number of similar characters among the question pairs, and the ratio of the similar characters.
- Word level features: This set includes: the total number of words for the pair of questions, the number of different words among the question pairs, and the ratio of the different words, the number of similar words among the question pairs, and the ratio of the similar words. Moreover, the type of similarity of question pairs is used as another feature. This feature is computed as a binary feature and depending on the question interrogative particles (i.e. similarity of the first word in each question). Also, the text overlap features were computed on the word level and based on our previous research [27]. The text overlap features include the number of overlapping words divided by the number of words in question1, the number of overlapping words divided by the number of words in question2, and the harmonic mean of the previous two features.
- Morphological features: Stemming was used to represent each question. The Arabic language is morphologically rich [37], therefore, representing the pair of questions using their stem words increases the chance of word similarity on the surface level.
- Semantic level features: This feature set includes: TF-IDF, Jaccard, and Cosine similarity measures. These features were computed for each pair of questions. The lexical similarity measures are computed on both the original questions and their stemmed features.
- Word Embedding features: The pre-trained model for Arabic content AraVec 3.0 [38] is used to compute the embeddings features for the input questions. The Twitter-CBOW with embedding size of 100 is used out of the AraVec available pre-trained models.

3.4. The developed models

This research implements three models for analyzing the semantic similarity of Arabic questions' pairs: i) a supervised-machine learning model using XGBoost [6], ii) an adapted Siamese deep learning recurrent architecture based on the work of [7], and iii) a pre-trained deep bidirectional transformer based on BERT model [8].

We have extracted features from the dataset to train the first two models, the XGBoost and the Siamese neural network. However, the BERT model is adapted directly without any feature extraction step. We have carefully selected these three models among many other models for many reasons. The XGBoost was the best performing model in [9], the Siamese deep learning model works well in semantic text similarity [7, 39], and the Google BERT model is the state-of-the-art model used for several natural-language processing (NLP) applications.

3.4.1. Supervised-machine learning model using XGBoost

XGBoost [6] is a short standing for eXtreme gradient boosting. XGBoost is a scalable machine learn- ing system for tree boosting and it is available as an open-source package. In the machine learning competition published by Kaggle in 2015, among the 29 winning solutions, 17 solutions adapted XGBoost. Among these 17 solutions, 8 solutions used XGBoost to train the model, while the rest 9 combined XGBoost with the artificial neural network as ensembles.

XGBoost approach provides a parallel tree boosting known as gradient boosted regression tree (GBRT) or gradient boosting machine (GBM) which is a scalable and efficient implementation of gradient boosting framework proposed by [40, 41]. XGBoost algorithm combines weak base learning models into a stronger learner in an iterative manner. It is available in several languages such as Python, R, and Julia. XGBoost can be integrated with several language data science pipelines as scikitlearn. The XGBoost model is trained in an additive manner. As shown in (1), ft needs to be added to minimize the objective $t^{(t)}$. Where y^{it-1} is the prediction of the $i-th$ instance at the $t-th$ iteration.

$$t^{(t)} = \overset{n}{1}|(y_i\hat{y}_i)^{t-1} + f_t(X_i)) + \Omega(f_t) \quad (1)$$

In this work, we used the XGBoost Python package introduced in [6] to train the model with the pre-defined features in order to enhanced approach for learning semantic similarities in Arabic questions. The XGBoost classifier was trained using the extracted features as explained in 3.3. The XGBoost model was trained with a maximum tree depth of 6 and a learning rate (eta)=0.06, 0.04, and 0.02 for 6.000 epochs on each.

3.4.2. A Siamese deep learning recurrent architecture

Siamese neural network implements two symmetric neural networks with shared wights to learn semantic similarities among inputs. Siamese neural networks is used in many semantic similarity applications such as: face verification using symmetric convolutional networks [42], speech understanding and speaker-specific information extraction [43], and in semantic text similarity [7, 39].

In this research, we utilized the Siamese neural network architecture to develop an enhanced approach for learning semantic similarities in Arabic questions. The model consists of two symmetric layers each contains an embedding layer followed by a bi-directional long short term memory (LSTM) layer and then an LSTM layer. Being Siamese-based the weights used to train both bi-LSTEM and LSTM layers for both questions are shared see Figure 2. The output of the symmetric layers are then concatenated with the output of the features layer and fed to fully connect dense layers with batch normalization [44] and Dropout [45] layers. The final layer is dense classification layer with activation of 'sigmoid' to get binary classification value whether questions are similar or not.

The batch normalization [44] and dropout [45] layers are used to regularize the output of the Siamese layers and to avoid common problems in deep learning such as: i) Internal covariate shift (i.e. "the change in the distribution of network activations due to the change in network parameters during training." [44]) and ii) Overfitting during neural network training. The term "dropout" refers to "randomly dropping out units (hidden and visible) in a neural network. By dropping a unit out, we mean temporarily removing it from the network, along with all its incoming and outgoing connections" [45]. The L2 regularization with value=0.001 was used with the fully connect dense layers.

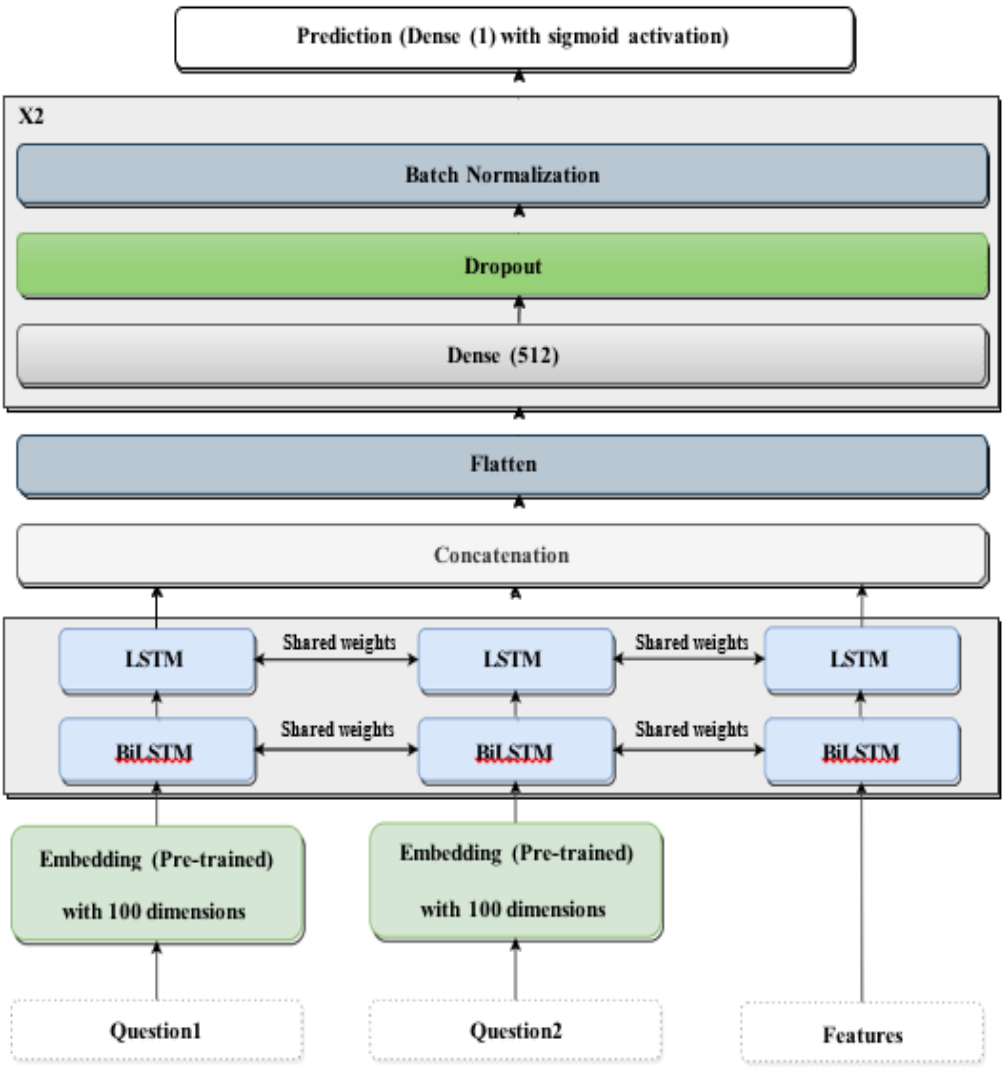

Figure 2. The Siamese-based model architecture

3.4.3. A Pre-trained deep bidirectional transformer based on BERT model

In our work, we also used bidirectional encoder representations from transformers (BERT) model [8]. BERT is a state-of-the-art model used for several natural-language processing (NLP) applications. BERT is a language representation released by Google in October 2018 utilizing the encoder-decoder transformer architecture to train the model representations using unannotated data. The BERT model representations are built over contextual representations like Semi-supervised Learning [46], ULMFit [47], and ELMo [48].

The BERT model has many versions, see Table 3 for more details. The Uncased versions of BERT model means that the text has been lower-cased before tokenization and the Cased versions means that the case of the text is preserved. For the sake of this research, the BERT-Base, Uncased is used. The BERT-Base, Uncased is built out of 12 layers with 768 hidden layers, with 12 heads, and 110 M trained parameters.

Table 3. The BERT model versions

BERT version	# Layers	# Hidden	# Heads	# Parameters
BERT-Large, Uncased	24	1024	16	340 M
BERT-Large, Cased	24	1024	16	340 M
BERT-Base, Uncased	12	768	12	110 M
BERT-Base, Cased	12	768	12	110 M

As shown in Figure 3, in the BERT model, the input represents a pair of sentences, which is here a pair of questions, in one token sequence. As shown, two questions packed together to the input token sequence. The first token of the sequence is a special classification token called "[CLS]". To differentiate the two questions in the token sequence, a special token called "[SEP]" is used to separate them. Then, a learned embedding is added to every token in order to indicate whether it belongs to Question1 or Question2. The input embedding is denoted as "E". The final hidden vector for the special "CLS" token is denoted as "C" and the final hidden vector for the i^{th} input token is denoted as Ti. The input embeddings are represented as the summation of the token embeddings, the segmentation embeddings, and the position embeddings.

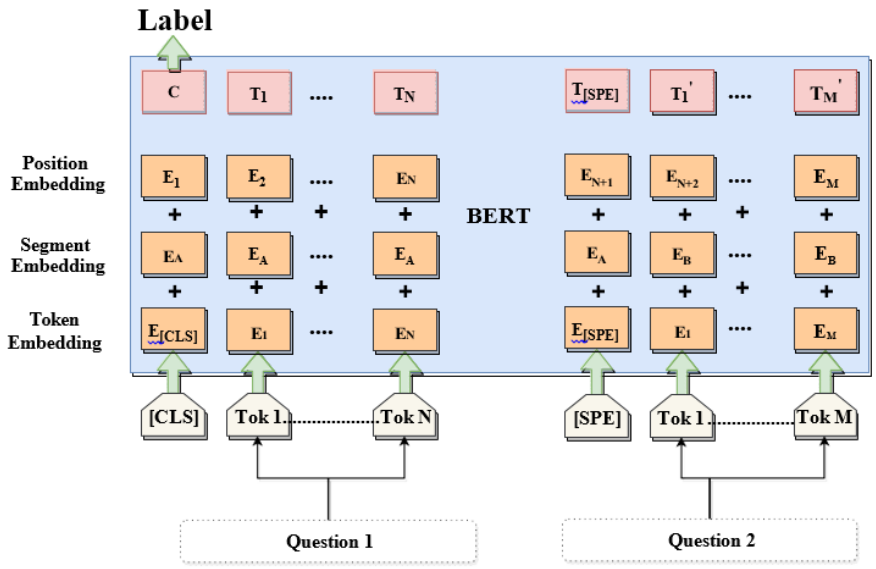

Figure 3. The BERT model embedding mechanism

As presented in Figure 4, the BERT-based model utilizes the encoder-decoder transformer architecture to learn the semantic similarity of the input questions. Transformers [49] implements different layers of multi- head self-attention with feed-forward and skipping mechanism. In contrast to traditional attention mechanism [50], the multi-head self-attention attends only to the input sequence of text and the multi-head functionality enables each layer to attend to different words within the input sequence of text. The positional encoding mechanism represents the input sequence order, words position within the sequence, and the distance between words as a vector which is then added to the embedding layer. These vectors help in

capturing the contextual information within the input sequence. Each self-attention layer is followed by residual connection represented by a normalization layer that adds the input vector of the self-attention layer to the output vector from the same self-attention layer helping to carry forgotten information to the next layer. For more information the reader is redirected to [49].

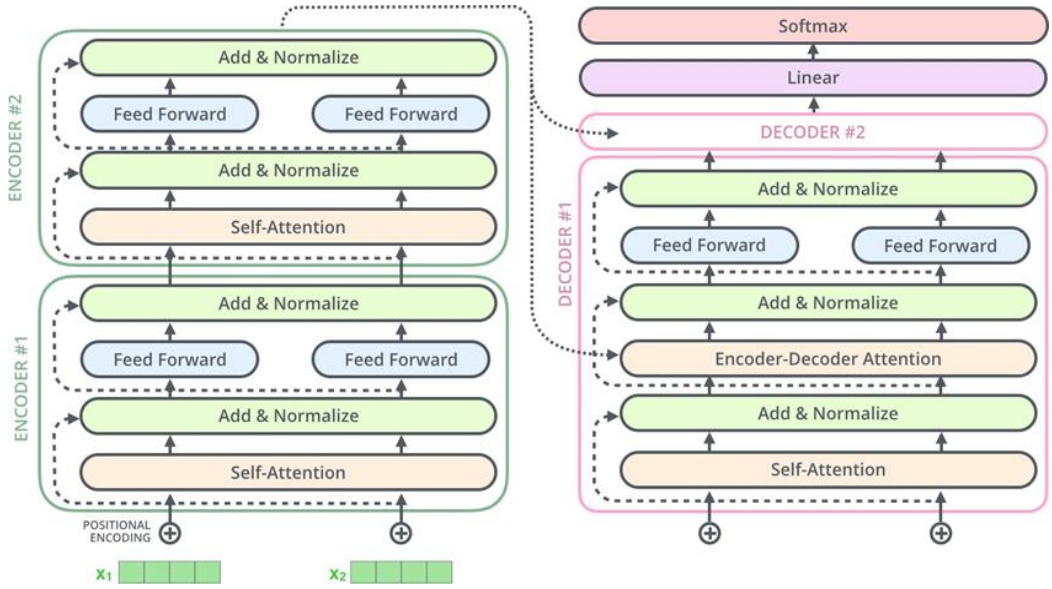

Figure 4. A transformer architecture of 2 stacked encoders and decoders

4. EXPERIMENTATION AND RESULTS

In this study, the three developed models are evaluated using the Arabic questions dataset provided by *mawdoo3.com*. These three models are XGBoost [6], Siamese neural network [7], and BERT model [8]. The F1 measure is used to evaluate the performance of the models. The F1 measure is the harmonic mean of precision and recall.

The XGBoost classifier was trained using the pre-engineered features computed on the training dataset with max tree depth of 6, learning rate (eta) of 0.06, 0.04, and 0.02 for 6.000 epochs. On the other hand, the Siamese-based model was trained using the pre-engineered features. The shared Bi-LSTM and LSTM layers had 100 hidden layers and an input size of 100. The model was trained for 100 epochs with early stopping on the 98 epoch. The early stopping is used to avoid training overfitting is based on monitoring the validation loss value. Only the model with the best weights was saved and then used for evaluating the test dataset. We have trained the model with the following hyper-parameters: hidden=100, embedding size=100, batch size=512, learning rate=0.001, and number of epochs=98.

Finally, the BERT-based model was trained for 20 epochs with data embedding size of 100, batch size (BS)=16, a learning rate (LR)=(2e-5-1e-5), a warm-up proportion (WP)=0.1, and number of iterations per loop (IPL)=(1000-250000). The model was trained on the stemmed version of the questions' pairs without using the other pre-engineered features, as shown in Table 4.

Table 4. The hyperparameters used to train the BERT-based model and their results on the test data

BS	# Epochs	LR	WP	IPL	F1-Score
16	2	2e-5	0.1	1000	88.77%
16	5	2e-5	0.1	10000	90.12%
16	10	1e-5	0.1	100000	91.56%
16	15	1e-5	0.1	150000	91.20%
16	20	1e-5	0.1	250000	92.99%

Table 5 shows that the BERT-based model outperforms the other two models with an F1=92.99%, whereas the Siamese-based model comes in the second place with F1=89.048%. Finally, the XGBoost, as a baseline model, achieved the lowest result of F1=86.086%. It is worth mentioning that the results we

obtained in this research is the best results on this dataset including our preliminary work in [9] in which the best model in [9] achieved F-measure of 82.61%.

Figure 5 depicts the model accuracy on both the training and validation datasets during the training phase. Figure 6 shows the loss value for both the training and the validation during the model training phase. As depicted in both figures no model overfitting can be seen during the training phase.

Table 5. The results obtained out of the three developed models

Model	XGBoost	Siamese-based	BERT-based
F1	86.086%	89.048%	**92.99%**

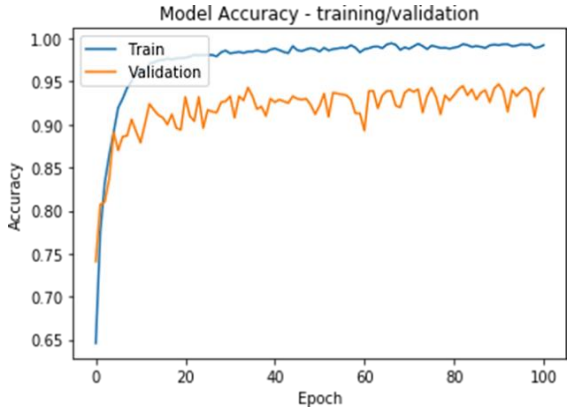

Figure 5. The computed accuracy of the Siamese-based model on the training and validation datasets

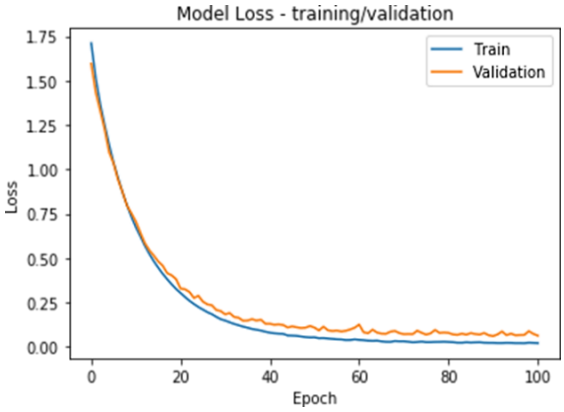

Figure 6. The loss values on the training and validation on each epoch during the Siamese-based training

5. DISCUSSION

Having a closer look to the experimentation results, it can be seen that the BERT-based model is outperforming the other two models in terms of F1 results with 3% higher than the Siamese-based model and 6% higher than the XGBoost one. The results go in line with literature as the Transformers based models such as BERT [8], ULMFit [47], and ELMO [48] are revolutionizing the NLP research. These techniques are leading the developed models in many NLP tasks such as text classification and sequence-to-sequence labeling.

In contrast to the other two developed models, the BERT-based model was able to learn the semantic similarity among input questions' pairs without the need for pre-engineered features. This explains the power of transformers in handling NLP tasks more efficiently when compared to CNN and RNN-based models. Computing features can reduce the applicability of the developed model for production services. Users may get negative experience waiting for the model to compute the features and then classify the input text.

Focusing on the pre-engineered features, selected features boasted the results of the developed models. The Siamese-based model achieved only an F1 value of 78.186% without the pre-engineered features (relying only on the embedding features). This indicates how powerful the features selected to train the models with a margin of results' enhancement reaches 10% for the Siamese-based model. This also emphasizes how powerful our BERT-based model when compared to the Siamese-based model without features with a difference of around 15% in terms of achieved results without features.

6. CONCLUSION AND FUTURE WORK

This research proposes three different approaches to analyze the semantic similarity between a pair of Arabic questions. The first model is a supervised-machine learning model using XGBoost trained using a set of pre-engineered features, the second is an adapted Siamese-based deep learning recurrent architecture also trained using a set of pre-engineered features, and finally, a pre-trained deep bidirectional transformer based on BERT model. The proposed approaches were evaluated using a dataset collected by mawdoo3.com (see section 3.1.). The evaluation results show that the BERT-based model outperforms the other two proposed models with 6% of enhancement in the F1-score (see section 5).

In this research, we have only considered detecting if two questions are similar or not. Detecting similar questions to a given question using our approach is an interesting avenue of future work. Besides that, we plan to enhance the BERT-based model architecture by combining the pre-engineered features to it, and investigate their impact on the model results. Moreover, we are planning to extract features from the BERT model and feed them to other machine learning approaches utilizing the flexible architecture of an encoder-decoder architecture in a transfer learning mechanism.

ACKNOWLEDGEMENTS

This research is partially funded by Jordan University of Science and Technology, Research Grant Number: 20170107 and 2019301.

REFERENCES

[1] J. Ramaprabha, S. Das, and P. Mukerjee, "Survey on sentence similarity evaluation using deep learning," *Journal of Physics: Conference Series*, vol. 1000, 2018, Art. no. 012070.
[2] S. Zhang, X. Zheng, and C. Hu, "A survey of semantic similarity and its application to social network analysis," *2015 IEEE International Conference on Big Data (Big Data)*, Santa Clara, CA, USA, 2015, pp. 2362-2367.
[3] D. Sa´nchez, M. Batet, D. Isern, and A. Valls, "Ontology-based semantic similarity: A new feature-based approach," *Expert systems with applications*, vol. 39, no. 9, pp. 7718-7728, 2012.
[4] E. Agirre, M. Diab, D. Cer, and A. Gonzalez-Agirre, "Semeval-2012 task 6: A pilot on semantic textual similarity," *Proceedings of the First Joint Conference on Lexical and Computational Semantics-Volume 1: Proceedings of the main conference and the shared task, and Volume 2: Proceedings of the Sixth International Workshop on Semantic Evaluation*, Association for Computational Linguistics, 2012, pp. 385-393.
[5] SemEval2019, "Semantic evaluation 2019," 2019. [Online]. Available: http://alt.qcri.org/semeval2019/.
[6] T. Chen, and C. Guestrin, "Xgboost: A scalable tree boosting system," *KDD '16: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2016, pp. 785-794.
[7] J. Mueller, and A. Thyagarajan, "Siamese recurrent architectures for learning sentence similarity," *AAAI'16: Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence*, 2016, pp. 2786-2792.
[8] J. Devlin, M. -W. Chang, K. Lee, K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," *Computation and Language*, 2019.
[9] M. Hammad, M. AL-Smadi, Q. Bani Baker, M. Al-asa'd, N. Al-khdour, M. B. Younes, E. Khwaileh, "Question to question similarity analysis using morphological, syntactic, semantic, and lexical features," *Journal of Universal Computer Science*, vol. 26, no. 6, pp. 671-697, 2020.
[10] F. Saric, G. Glavas, M. Karan, J. Snajder, B. D. Basic, "Takelab: Systems for measuring semantic text similarity," *First Joint Conference on Lexical and Computational Semantics (*SEM)*, 2012, pp. 441-448.
[11] T. Zhu, M. Lan, "ECNU: Leveraging on ensemble of heterogeneous features and information enrichment for cross level semantic similarity estimation," *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014)*, 2014, pp. 265-270.
[12] N. P. A. Vo, O. Popescu, and T. Caselli, "FBK-TR: SVM for semantic relatedness and corpus patterns for RTE," *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014)*, 2014, pp. 289-293.
[13] J. Zhao, and M. Lan, "ECNU: Leveraging word embeddings to boost performance for paraphrase in twitter," *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015)*, 2015, pp. 34-39.
[14] A. Sanborn, and J. Skryzalin, "Deep learning for semantic similarity, CS224d: Deep Learning for Natural Language Processing Stanford," CA, USA: Stanford University, 2015.
[15] S. Arora, Y. Liang, and T. Ma, "A simple but tough-to-beat baseline for sentence embeddings," *Published as a conference paper at ICLR 2017*, pp. 1-16, 2017.
[16] S. Wan, Y. Lan, J. Guo, J. Xu, L. Pang, X. Cheng, "A deep architecture for semantic matching with multiple positional sentence representations," *Thirtieth AAAI Conference on Artificial Intelligence*, 2016.
[17] J. Tian, Z. Zhou, M. Lan, Y. Wu, "ECNU at semeval-2017 task 1: Leverage kernel-based traditional NLP features and neural networks to build a universal model for multilingual and cross-lingual semantic textual similarity," *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, 2017, pp. 191-197.
[18] D. Prijatelj, J. Kalita, J. Ventura, "Neural networks for semantic textual similarity," *Proceedings of the 14th International Conference on Natural Language Processing (ICON-2017)*, 2017, pp. 456-465.
[19] H. He, and J. Lin, "Pairwise word interaction modeling with deep neural networks for semantic similarity measurement," *Proceedings of NAACL-HLT 2016*, 2016, pp. 937-948.
[20] H. He, K. Gimpel, J. Lin, "Multi-perspective sentence similarity modeling with convolutional neural networks," *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, 2015, pp. 1576-1586.
[21] J. Mueller, and A. Thyagarajan, "Siamese recurrent architectures for learning sentence similarity," *Thirtieth AAAI Conference on Artificial Intelligence*, 2016, pp. 2766-2792.
[22] B. Agarwal, H. Ramampiaro, H. Langseth, M. Ruocco, "A deep network model for paraphrase detection in short text messages," *Information Processing & Management*, vol. 54, no. 6, pp. 922-937, 2018.
[23] Y. Le, Z. J. Wang, Z. Quan, J. He, B. Yao, "ACV-TREE: A new method for sentence similarity modeling," *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence (IJCAI)*, 2018, pp. 4137-4143.

[24] J. Tian, M. Lan, Y. Wu, J. Wang, L. Qiu, S. Li, L. Jun, L. Si, "An adversarial joint learning model for low-resource language semantic textual similarity," *European Conference on Information Retrieval (ECIR 2018)*, vol. 10771, 2018, pp. 89-101.
[25] X. Tang, S. Chen, L. Do, Z. Min, F. Ji, H. Yu *et al.*, "Improving multilingual semantic textual similarity with shared sentence encoder for low-resource languages," *Ground Al*, 2018.
[26] Y. Yang, S. Yuan, D. Cer, S.-y. Kong, N. Constant, P. Pilar *et al.*, "Learning semantic textual similarity from conversations," *Proceedings of The Third Workshop on Representation Learning for NLP*, pp. 164-174, 2018.
[27] A. -S. Mohammad, Z. Jaradat, A. -A. Mahmoud, Y. Jararweh, "Paraphrase identification and semantic text similarity analysis in Arabic news tweets using lexical, syntactic, and semantic features," *Information Processing & Management*, vol. 53, no. 3, pp. 640-652. 2017.
[28] D. Sagheer, and F. Sukkar, "Arabic sentences classification via deep learning," *International Journal of Computer Applications*, vol. 182, no. 5, pp. 40-46, 2018.
[29] S. Alzahrani, "Cross-language semantic similarity of Arabic-English short phrases and sentences," *Journal of Computer Science (JCS)*, vol. 12, no. 1, pp. 1-18, 2016.
[30] D. Suleiman, A. Awajan, N. Al-Madi, "Deep learning based technique for plagiarism detection in arabic texts," *2017 International Conference on New Trends in Computing Sciences (ICTCS)*, Amman, Jordan, 2017, pp. 216-222.
[31] M. Al-Suhaiqi, M. A. Hazaa, M. Albared, "Arabic English cross-lingual plagiarism detection based on keyphrases extraction, monolingual and machine learning approach," *Asian Journal of Research in Computer Science*, vol. 2, no. 3, pp. 1-12, 2018.
[32] H. Cherroun, A. Alshehri *et al.*, "Disguised plagiarism detection in Arabic text documents," *2018 2nd International Conference on Natural Language and Speech Processing (ICNLSP)*, Algiers, Algeria, 2018, pp. 1-6.
[33] J. Ferrero, D. Schwab, H. Cherroun *et al.*, "Word embedding-based approaches for measuring semantic similarity of Arabic-English sentences," *International Conference on Arabic Language Processing (ICALP 2017)*, vol. 782, 2017, pp. 19-33.
[34] R. M. Badry, and I. F. Moawad, "A semantic text summarization model for Arabic topic-oriented," *International Conference on Advanced Machine Learning Technologies and Applications (AMLTA 2019)*, vol. 921, 2019, pp. 518-528.
[35] P. Shapiro, and K. Duh, "Morphological word embeddings for Arabic neural machine translation in low-resource settings," *Proceedings of the Second Workshop on Subword/Character LEvel Models*, 2018, pp. 1-11.
[36] E. Loper, and S. Bird, "NLTK: the natural language toolkit," *Proceedings of the ACL Interactive Poster and Demonstration Sessions*, 2004, pp. 214-217.
[37] M. Abdul-Mageed, M. T. Diab, M. Korayem, "Subjectivity and sentiment analysis of modern standard Arabic," *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, 2011, pp. 587-591.
[38] A. B. Soliman, K. Eissa, S. R. El-Beltagy, "ARAVEC: A set of Arabic word embedding models for use in Arabic NLP," *Procedia Computer Science*, vol. 117, pp. 256-265, 2017.
[39] W. T. Yih, K. Toutanova, J. C. Platt, C. Meek, "Learning discriminative projections for text similarity measures," *Proceedings of the fifteenth conference on computational natural language learning*, Portland, USA, 2011, pp. 247-256.
[40] J. Friedman, T. Hastie, R. Tibshirani *et al.*, "Additive logistic regression: a statistical view of boosting (with discussion and a rejoinder by the authors)," *The annals of statistics*, vol. 28, no. 2, pp. 337-407, 2000.
[41] J. H. Friedman, "Greedy function approximation: a gradient boosting machine," *Annals of statistics*, vol. 29, no. 5, pp. 1189-1232, 2001.
[42] S. Chopra, R. Hadsell, Y. LeCun *et al.*, "Learning a similarity metric discriminatively, with application to face verification," *CVPR*, vol. 1, pp. 539-546, 2005.
[43] K. Chen, and A. Salman, "Extracting speaker-specific information with a regularized Siamese deep network," *Proceedings of the 24th International Conference on Neural Information Processing Systems*, 2011, pp. 298-306.
[44] S. Ioffe, and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal co-variate shift," *Proceedings of the 32nd International Conference on International Conference on Machine Learning*, vol. 37, 2015, pp. 448-456.
[45] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, R. Salakhutdinov, "Dropout: a simple way to prevent neural networks from overfitting," *The journal of machine learning research*, vol. 15, no. 1, pp. 1929-1958, 2014.
[46] A. M. Dai, and Q. V. Le, "Semi-supervised sequence learning," *Advances in neural information processing systems*, 2015, pp. 3079-3087.
[47] J. Howard, and S. Ruder, "Universal language model fine-tuning for text classification," *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics*, Melboume, Australia, vol. 1, 2018, pp. 328-339.
[48] M. E. Peters, M. Neumann, M. Iyyer, M. Gardner, C. Clark, K. Lee *et al.*, "Deep contextualized word representations," *Proc. of NAACL*, 2018.
[49] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Kaiser, I. Polosukhin, "Attention is all you need," *Advances in neural information processing systems*, 2017, pp. 5998-6008.
[50] D. Bahdanau, K. Cho, Y. Bengio, "Neural machine translation by jointly learning to align and translate," *conference paper at ICLR*, 2015, pp. 1-15.