

The Inception Team at NSURL-2019 Task 8: Semantic Question Similarity in Arabic

Hana Al-Theiabat and Aisha Al-Sadi *

Jordan University of Science and Technology, Irbid, Jordan
haaltheiabat13@cit.just.edu.jo, asalsadi16@cit.just.edu.jo

Abstract

This paper describes our method for the task of Semantic Question Similarity in Arabic in the workshop on NLP Solutions for Under Resourced Languages (NSURL). The aim is to build a model that is able to detect similar semantic questions in Arabic language for the provided dataset. Different methods of determining questions similarity are explored in this work. The proposed models achieved high F1-scores, which range from (88% to 96%). Our official best result is produced from the ensemble model of using pre-trained multilingual BERT model with different random seeds with 95.924% F1-Score, which ranks the first among nine participants teams.

1 Introduction

Semantic matching or semantic similarity is a significant part of natural language processing (NLP) field for its variety of tasks. It used to measure the similarity and the relationship between different textual elements, such as words, sentences, or documents. Semantic matching has been involved in many NLP applications; including question answering, where it is used to assess question answering and retrieval tasks by employing it to estimate the similarity of query answer among all candidate answers (Wang et al., 2016). In addition, it has played a significant role in top-k re-ranking in machine translation (Brown et al., 1993), information extraction (Grishman, 1997) and automatic text summarization (Ponzanelli et al., 2015).

Natural language has complicated structures either from sequential or hierarchical perspectives, capturing the relationship between two questions is becoming a challenging task. For example, questions that have the same meaning while their words have a different order. An effective semantic match-

ing algorithm, therefore, needs to consider an appropriate semantic representation to capture the similarity without being affected with words order.

This paper focuses on detecting semantic question similarity, which is a common challenge in Question-and-answer (Q&A) websites, such as Quora and Stack Overflow. This work targets Arabic questions dataset published by Mawdoo3 AI¹. Most of these questions are related to information provided by Mawdoo3.com which is the largest comprehensive Arabic content website. For these websites, the benefit of detecting duplicate questions is to improve the efficiency of search engines by being aware of the different paraphrases of the same question.

The rest of paper is organized as follows. Section 2 presents related works. While Section 3 presents some details about the dataset. Section 4, presents the proposed models for solving the semantic similarity in Arabic language task. Results for all proposed models and the final results are presented in Section 5. Finally, the paper conclusion is presented in Section 6.

2 Related Work

Semantic matching has been a long-established problem in NLP. Many approaches were proposed to solve this problem. The conventional approaches were mainly based on representing text as a vector of word features. The bag-of-words (BoW) method (Wu et al., 2008) employed the word occurrence and Term Frequency-Inverse Document Frequency (TF-IDF) (Paltoglou and Thelwall, 2010) as the word feature. However, these types of models disregard word meaning, orders, and even grammar. In contrast, word embedding models such as word2vec (Mikolov et al., 2013) and Glove (Pennington et al., 2014) have been widely used instead

* These authors contributed equally to the work

¹ <https://ai.mawdoo3.com/nsurl-2019-task8>

of BoW as they can learn distributional semantic representation for words. So based on word embeddings, the Word Movers Distance (WMD) (Kusner et al., 2015) was proposed to measure the dissimilarity between two texts assuming that similar words should have similar vectors. Although WMD can estimate semantic similarity between texts, the order, and interactions between words are excluded.

Recently many deep learning models have been proposed for text matching. A common framework has been adopted is the Siamese architecture (Mueller and Thyagarajan, 2016; Pang et al., 2016; Severyn and Moschitti, 2015; Wang et al., 2017) where the encoder, which can be either Convolutional Neural Network (CNN) or Recurrent Neural Network (RNN), is applied individually on the two input texts, so both texts are encoded into intermediate contextual representations. Then, the matching result is generated by performing a scoring mechanism over contextual representations. Although this framework supports parameter sharing in its network, it purely learns complicated relationships among texts.

Another framework is based on matching aggregation (Wang and Jiang, 2017) which first matches the small units (such as words) of two texts to produce comparison vectors, then these vectors are aggregated and fed into a CNN or RNN for the final classification. This framework improves capturing the interactive features between two texts, but still it limits exploring the matching in only word-word manner.

As the main focus of this paper is to detect semantically equivalent questions, the following is the review of related approaches that were adopted to detect duplicate questions on Quora dataset. As Quora recently published a dataset of 400K labeled questions, massive researches have been proposed on this dataset for question paraphrase identification challenge (qou). One Relevant approach that was proposed for this challenge is the Bilateral Multi-Perspective Matching model (BIMPM) model (Wang et al., 2017) which encodes two questions with a Bidirectional Long Short-Term Memory Network (BiLSTM). Then, a multi-perspective matching in the two directions is applied to both questions, and for each time step, questions are matched using different types of extensive matching. On Quora dataset, the result of this model reached 88.17%. In (Mirakyan et al., 2018), a novel

architecture can obtain a high-level understanding of the question pairs through extracting the semantic features using dense interaction tensors (attention) network which called Densely Interactive Inference Network (DIIN). DIIN outperforms BiLSTM on Quora to achieve accuracy of 89.06%. Moreover, Multi-Task Deep Neural Network (MT-DNN) (Liu et al., 2019) achieved competitive performance on several tasks including question paraphrase on Quora with an accuracy of 89.6%. Specifically, MT-DNN Combined multi-task learning and pre-trained bidirectional transformer model for language representation learning.

3 Dataset Description

The dataset used in this task is provided by Mawdoo3 (Seelawi et al., 2019). It is a dataset for questions in Arabic language, it consists of 11,997 labeled question pairs as training data, and 3,715 question pairs as testing data. Label '1' means the question pairs are similar in semantic where label '0' means the opposite. 55% of the training question pairs are with label '0', and 45% are with label '1'. The max length of question 1 is 14 words with an average of 5.7 words per question, while the max length of question 2 is 28 words with an average of 5.3 words per question. Table 1 shows samples from the training dataset.

question1	question2	is_duplicate
ما هي الطرق الصحيحة للاعتناء بالحامل؟	كيف اهتم بطفلي؟	0
ما هي وسائل الاتصالات الحديثة؟	ماذا تعني بوسائل الاتصال الحديثة؟	1
ما طريقة تحضير محشي الكوسا؟	من طرق تحضير محشي الكوسا؟	1
ما طريقة تحضير حلى الطليقات؟	من طرق تحضير طليقات الكيك؟	0
من الآيات القرآنية عن الرعي والرعية؟	ما هو تعريف الرعي والرعية؟	0
أين تقع قارة أوروبا؟	ما هو موقع اليمن؟	0

Table 1: Question samples from Mawdoo3 dataset

The only processing step that was applied to the dataset is to unify countries names, some examples are shown in Table 2.

الأردن	المملكة الأردنية الهاشمية
سوريا	الجمهورية السورية
سنغافورة	سنغافورا

Table 2: Unify countries names example

4 Methodology

In this work, four different deep learning approaches are presented to solve the semantic similarity task, which are RNN based model, CNN based model, multi-head attention based model,

and finally BERT model. In this section, each model is discussed.

4.1 Convolutional Neural Network Model

In NLP field, CNN has shown the ability to extract most informative n-gram features from the input sequence, and then apply the activation on these features (Kim, 2014). Although CNN is known for the applications in the image processing field, it is used here for text classification application.

The proposed model architecture is shown in Figure 1. Firstly, the words are mapped in the dictionary to get a representation for each word. Then each question is fed to three consecutive layers. In each layer, the convolutional layer is applied, followed by activation and then max pooling. Hence, each question's output is a feature representation which is used to get the similarity label by computing the cosine similarity between the two questions features.

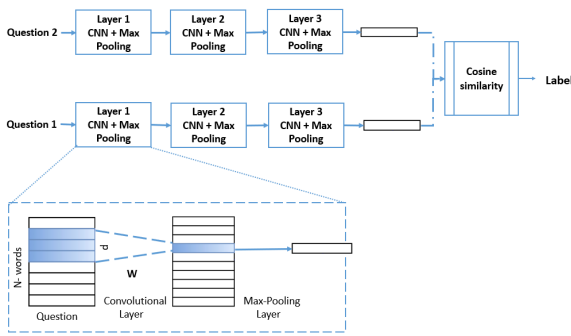


Figure 1: CNN model architecture used for detecting semantic questions similarity

4.2 Recurrent Neural Network Model

The significant advantage of RNNs is the computation of the same task over each element of the sequence, so the output for each block depends on the previous computations. Hence, RNN has been increasingly prevalent in NLP field specifically for RNN types that have a memory to remember the information through the sequence.

In this model, the input is a sequence of question-pairs that are concatenated to represent a single sequence. Then, the sequence is encoded by the dictionary to be fed into a bi-directional Gated Recurrent Units (GRUs) network with 128 hidden units to generate the similarity label as output.

4.3 Multi-head Attention Network Model

Multi-head attention model (Vaswani et al., 2017) allows to learn on various locations of the encoded

words. Our network consists of a stacked encoder-decoder structure with eight heads.

For each question-pairs of sequence length n , at each layer l , the encoder maps a sequence of words $Q_l = w_1^l, \dots, w_n^l$ into hidden representation $h^l = h_1^l, \dots, h_n^l$. After computing the attention on all positions jointly, the transformer stacks all hidden representation h^l at the current layer l together into matrix H^l . Given h , the decoder then generates output sequence $y^l = y_1^l, \dots, y_n^l$, and after that apply softmax to estimate the output label. The transformer also contains two sub-layers, a multi-head attention layer, and a position-encoding layer.

The position-encoding layer benefits the network to keep track of relative positions for each word in the sequence since the context and the meaning of a sequence depend on the order of its words.

In the multi-head attention layer, instead of computing single attention on the overall sequence, it jointly gets attention from different representations at different positions. As a result, each head looks differently on encoder output, and the decoder easily learns to retrieve valuable information from the encoder.

4.4 BERT Model

Recently, pre-training language models have shown a significant role to improve many NLP tasks including question-pairs paraphrasing (Dolan and Brockett, 2005). There are two approaches to apply these pre-trained language representations on NLP tasks; either feature-based or fine-tuning. For the feature-based approach (Peters et al., 2018), researchers use the output of pre-trained model as additional features in their models, based on the task they target. On the other hand, the fine-tuning approach (Radford et al., 2018) permits the model to be trained on another task by learning task-specific parameters. The two strategies were mentioned previously have limitations to learning general language representations since they adopt the left-to-right unidirectional architectures. On the other hand, Bidirectional Encoder Representations from Transformers (BERT) (Devlin et al., 2018), has strongly outperformed previous cutting-edge unidirectional models.

BERT model relies on the multi-head self-attention mechanism, which enables it to achieve the state-of-the-art accuracy on a wide range of tasks such as, natural language inference, question answering, and sentence classification. The

architecture of BERT model is built upon the transformer layer, which is called the self-attention layer. For each layer, the representations of words are exchanged from previous layers regardless of their positions, in contrast to traditional unidirectional models. For each input word, the model learns bidirectional encoder representations by using the masked language model, which randomly masks some of the words from the input to predict the masked word contextually.

As BERT offers pre-trained models for English language and multilingual model for 104 languages (ber) including the Arabic language, we applied the sentence pairs classification task on Arabic questions through fine-tuning the multilingual model as illustrated in Figure 2.

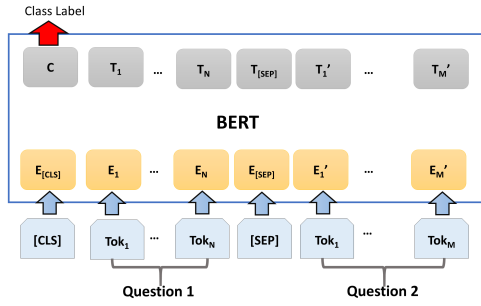


Figure 2: BERT model used for question pair similarity classification task

5 Experiments and Results

For each of the four models explained in the methodology section, different hyper-parameters are used, such as learning rate, number of hidden nodes, and number of epochs. Table 3 shows the main parameters values that give the best results for each model.

The evaluation metric that was used for this task is F1-Score that measures the precision p and recall r together as illustrated in the equations [1]-[3]:

$$F1 = 2p.r/(p + r) \quad (1)$$

$$p = tp/(tp + fp) \quad (2)$$

$$r = tp/(tp + fn) \quad (3)$$

where:

- tp: true positive examples
- fp: false positive examples

Model	Main parameters
RNN Model	hidden size = 128 cell type = GRU bidirectional = true number of train epochs = 10 train batch size = 512 learning rate = 0.001
CNN Model	number of filters = 50, 50, 50 filter sizes = 2, 3, 4 number of blocks = 2 number of train epochs = 10 train batch size = 512 learning rate = 0.001
Multi-Head Attention Model	number of heads = 8 use residual = false layer normalization = false number of train epochs = 10 train batch size = 512 learning rate = 0.001
BERT Model	max seq length = 50 train batch size = 8 learning rate = 2e-5 number of train epochs=50

Table 3: Main parameters for each proposed model

- fn: false negative examples

Test data evaluation is automatically done online on Kaggle website by submitting the test predictions file. The evaluation system is as the following:

- Public score: calculated with approximately 30% of the data
- Private score: calculated with approximately 70% of the data

During the competition, the public score for each submitted file was shown directly. Then after the competition ended, the submitted file with the highest public score was chosen to calculate its private score and compete other teams based on it.

Table 4 shows the highest F1-Score for each of the four models for the public score and the private score of the test data. As illustrated, BERT model with pre-trained multilingual outperforms the remaining models with F1-score of 96.050% on the public score, and 95.617% on the private score.

Note that the previous results are based on the best public score for every single model of the four models. Since BERT model gives the best results, we conducted other experiments with different random seeds in order to ensemble BERT model. Hard voting is used as ensemble method in which the predictions for each BERT experiments are involved in voting to get the final prediction.

Model	Public Score (%)	Private Score (%)
RNN Model	88.061	88.312
CNN Model	88.330	88.773
Multi-Head Attention Model	86.804	87.889
BERT Model	96.050	95.617

Table 4: Results of 30% of the test data

Model	Public Score (%)	Private Score (%)
Ensemble of best 3 seeds	95.960	96.155
Ensemble of best 4 seeds	96.499	95.924
Ensemble of best 5 seeds	95.691	96.232
Ensemble of best 6 seeds	95.332	96.001

Table 5: BERT Ensemble Results

Table 5 shows the results of the ensemble models of BERT with different number of experiments each with different random seed.

In the ensemble of four and six seeds when the number of votes is equal, high priority was given to the experiments with the best public score.

The result of the ensemble of four seeds has the best public score, so it was chosen for the final evaluation and got the first place. Although other seeds results had lower public scores, they had higher private scores than the official private score. So actually, our best result is 96.232% while the official best result is 95.924%.

6 Conclusion

This paper describes our participation in NSURL Task 8; Semantic Question Similarity in Arabic. Different models were proposed for the task; RNN model, CNN model, Multi-head model, BERT model, and ensemble model of BERT. The ensemble model clearly outperforms all other models in this task by achieving 95.924% F1-Score. This performance ranks first place among nine participating teams.

References

[Bert - google research github.](#)

[Quora dataset question pairs.](#)

Peter F. Brown, Stephen Della Pietra, Vincent J. Della Pietra, and Robert L. Mercer. 1993. The mathematics of statistical machine translation: Parameter estimation. *Computational Linguistics*, 19(2):263–311.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. [BERT: pre-training of deep bidirectional transformers for language understanding.](#) *CoRR*, abs/1810.04805.

William B. Dolan and Chris Brockett. 2005. [Automatically constructing a corpus of sentential paraphrases.](#) In *Proceedings of the Third International Workshop on Paraphrasing, IWP@IJCNLP 2005, Jeju Island, Korea, October 2005, 2005.*

Ralph Grishman. 1997. [Information extraction: Techniques and challenges.](#) In *Information Extraction: A Multidisciplinary Approach to an Emerging Information Technology, International Summer School, SCIE-97, Frascati, Italy, 14-18, 1997*, pages 10–27.

Yoon Kim. 2014. [Convolutional neural networks for sentence classification.](#) In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014, October 25-29, 2014, Doha, Qatar; A meeting of SIGDAT, a Special Interest Group of the ACL*, pages 1746–1751.

Matt Kusner, Yu Sun, Nicholas Kolkin, and Kilian Weinberger. 2015. From word embeddings to document distances. In *International conference on machine learning*, pages 957–966.

Xiaodong Liu, Pengcheng He, Weizhu Chen, and Jianfeng Gao. 2019. [Improving multi-task deep neural networks via knowledge distillation for natural language understanding.](#) *CoRR*, abs/1904.09482.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. [Efficient estimation of word representations in vector space.](#) In *1st International Conference on Learning Representations, ICLR 2013, Scottsdale, Arizona, USA, May 2-4, 2013, Workshop Track Proceedings.*

Martin Mirakyan, Karen Hambardzumyan, and Hrant Khachatryan. 2018. [Natural language inference over interaction space: ICLR 2018 reproducibility report.](#) *CoRR*, abs/1802.03198.

- Jonas Mueller and Aditya Thyagarajan. 2016. [Siamese recurrent architectures for learning sentence similarity](#). In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence, February 12-17, 2016, Phoenix, Arizona, USA.*, pages 2786–2792.
- Georgios Paltoglou and Mike Thelwall. 2010. [A study of information retrieval weighting schemes for sentiment analysis](#). In *ACL 2010, Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics, July 11-16, 2010, Uppsala, Sweden*, pages 1386–1395.
- Liang Pang, Yanyan Lan, Jiafeng Guo, Jun Xu, Shengxian Wan, and Xueqi Cheng. 2016. [Text matching as image recognition](#). In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence, February 12-17, 2016, Phoenix, Arizona, USA.*, pages 2793–2799.
- Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. [Glove: Global vectors for word representation](#). In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014, October 25-29, 2014, Doha, Qatar, A meeting of SIGDAT, a Special Interest Group of the ACL*, pages 1532–1543.
- Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. [Deep contextualized word representations](#). In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2018, New Orleans, Louisiana, USA, June 1-6, 2018, Volume 1 (Long Papers)*, pages 2227–2237.
- Luca Ponzanelli, Andrea Mocci, and Michele Lanza. 2015. [Summarizing complex development artifacts by mining heterogeneous data](#). In *12th IEEE/ACM Working Conference on Mining Software Repositories, MSR 2015, Florence, Italy, May 16-17, 2015*, pages 401–405.
- Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. 2018. Improving language understanding with unsupervised learning. Technical report, Technical report, OpenAI.
- Haitham Seelawi, Ahmad Mustafa, Al-Bataineh Hesham, Wael Farhan, and Hussein T. Al-Natsheh. 2019. NSURL-2019 task 8: Semantic question similarity in arabic. In *Proceedings of the first International Workshop on NLP Solutions for Under Resourced Languages, NSURL '19, Trento, Italy*.
- Aliaksei Severyn and Alessandro Moschitti. 2015. [Learning to rank short text pairs with convolutional deep neural networks](#). In *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval, Santiago, Chile, August 9-13, 2015*, pages 373–382.
- Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. [Attention is all you need](#). In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, 4-9 December 2017, Long Beach, CA, USA*, pages 5998–6008.
- Shuohang Wang and Jing Jiang. 2017. [A compare-aggregate model for matching text sequences](#). In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*.
- Zhiguo Wang, Wael Hamza, and Radu Florian. 2017. Bilateral multi-perspective matching for natural language sentences. *arXiv preprint arXiv:1702.03814*.
- Zhiguo Wang, Haitao Mi, and Abraham Ittycheriah. 2016. [Sentence similarity learning by lexical decomposition and composition](#). *CoRR*, abs/1602.07019.
- Ho Chung Wu, Robert Wing Pong Luk, Kam-Fai Wong, and Kui-Lam Kwok. 2008. [Interpreting TF-IDF term weights as making relevance decisions](#). *ACM Trans. Inf. Syst.*, 26(3):13:1–13:37.