NSURL-2019 Shared Task 8: Semantic Question Similarity in Arabic

Haitham Seelawi
haitham.seelawi@gmail.com

Ahmad Mustafa
Jordan University of Science
and Technology, Jordan
ammustafa@just.edu.jo

Hesham Al-Bataineh
AI Department
Mawdoo3 Ltd
Amman, Jordan
hisham.bataineh@mawdoo3.com

Wael Farhan
AI Department
Mawdoo3 Ltd
Amman, Jordan
wael.farhan@mawdoo3.com

Hussein T. Al-Natsheh
AI Department
Mawdoo3 Ltd
Amman, Jordan
h.natsheh@mawdoo3.com

Abstract

Question semantic similarity (Q2Q) is a challenging task that is very useful in many NLP applications, such as detecting duplicate questions and question answering systems. In this paper, we present the results and findings of the shared task (Semantic Question Similarity in Arabic). The task was organized as part of the first workshop on NLP Solutions for Under Resourced Languages (NSURL 2019) The goal of the task is to predict whether two questions are semantically similar or not, even if they are phrased differently. A total of 9 teams participated in the task. The datasets created for this task are made publicly available to support further research on Arabic Q2Q.

1 Introduction

Semantic Textual Similarity (STS) is a core task in Natural Language Processing and Understanding (NLP/NLU). Simply put, STS is concerned with inferring the similarity in meaning between a pair of sentences. It should be mentioned that there are other levels of granularity for STS such as: Lexical (i.e. single words), full paragraphs or whole documents.

In this paper, we focus on the STS of a question pair (or *Q2Q* Similarity). We assume that if two questions have the same answers, then they are semantically similar. Otherwise, if the answers are different or partially different, then the pair is considered non-equivalent.

STS provides the basis for Question Answering systems (QA). As the name suggests, QA systems are computer systems which can answer questions posed in a natural language form. These questions can be of either factoid or non-factoid nature. Factoid questions can be defined as questions for which a complete answer can be given in 50 bytes or less (a few words) (Soricut and Brill, 2004). These are typically questions that start with who, what, when or where, and have definitive answers. Non-factoid questions, on the other hand, require longer answers. They are mainly instructional or explanatory in nature.

One possible way to build QA systems using STS is having predefined questions along with their answers. When a user asks a question, a ranked list of these questions can be obtained, and based on that list, the best answer can be returned to the user. This method can be used, both, for factoid and non-factoid questions.

One important application to Q2Q is identifying duplicate questions in community question answering platforms (e.g., quora.com). Users may ask questions that might be already asked and answered by the community. Finding these duplicate questions saves the effort and time spent in answering already answered questions. However, detecting duplicate questions is challenging because these questions, although are semantically similar, they might be phrased differently. Moreover, dealing with the Arabic language in Q2Q similarity is challenging due to several factors. Arabic Q2Q datasets are scarce and limited in size. Moreover, the Arabic language is one of the most morphologically rich languages.

In this paper, we present the results and findings of the shared task (Semantic Question Similarity in Arabic). The task was organized as part of the first workshop on NLP Solutions for Under Resourced Languages (NSURL 2019)[1] The goal of the task is to predict whether two questions are similar or not. A total of 9 teams participated in the task. Among them, 4 teams have provided description papers, which are included in the NSURL workshop proceedings.

The rest of this paper is organized as the following. In Section 2, we discuss previously published

[1] http://nsurl.org/

work relating to Q2Q in Arabic. Section 3 provides an overview of the datasets used in the task. Next, in Section 4 we briefly describe the participants and the approaches they propose. Then we discuss the experiments and analyze the results of the competition in Section 5. Finally, we conclude in Section 6.

2 Related Work

Despite its importance and utility in NLP applications, research on STS at the level of sentences and higher, has only picked up steam in the past ten years (Cer et al., 2017). Nonetheless a lot has been accomplished since, but mainly in the English language. In the case of Arabic, there is plenty of room for new research to advance the current state of the art in this regard (Alian and Awajan, 2018) (Nakov et al., 2016). Therefore, most of our review below will focus on methods developed and used in English mainly, which might not be directly applicable to Arabic.

Some of the earliest methods used in the field made extensive use of so-called knowledge-based semantic similarities between words (Majumder et al., 2016). These can be thought of as lexical databases that model the semantic relationships of different words, taking into consideration their different senses. At the center of these databases is the concept of "synsets", which are groups of words that refer to a specific concept. The most popular such database is WordNet (Miller and Fellbaum, 2007). With the assistance of word alignment methods, various meaningful numerical features pertaining to the lexical units comprising a pair of sentences can be obtained from WordNet. Combined with other textual features, such as Part of Speech (POS), and Term Frequency - Inverse Document Frequency (TF-IDF), and fed into strong classifiers, such methods obtain very good results, albeit on closed domains of assessment (Saric et al., 2012; Sogancioglu et al., 2017; Pilehvar et al., 2013). Nonetheless, it can be easily seen that the construction of such databases, is very expensive in terms of human effort.

Semantic relationships can be modeled using another class of methods named Word Vector Representations (WVR). One of the biggest advantages of such methods is that they are typically trained in an unsupervised manner, making their construction very cheap in terms of human annotation. Some of these methods include Word2Vec (Mikolov et al., 2013), Glove (Pennington et al., 2014), ELMo (Peters et al., 2018) and BERT (Devlin et al., 2018). These word representations significantly boost the performance of machine learning algorithms (Mikolov et al., 2013), especially deep learning-based approaches.

One of the earlier and more basic methods of using WVR in STS, consisted in pooling the corresponding dimensions of tokens in a given sentence, using a specific pooling method, such as the average, or the maximum, to obtain a sentence level representation from WVR. The representation of each sentence in the pair would then serve as the input into a classifier or a predefined measure of similarity. One of the obvious advantages of such a method is its simplicity, and that it can be readily used in many classes of machine learning algorithms. However, it is apparent that by using pooling, we are losing all the information about the order of tokens in the original sentences, which definitely matters in defining the meaning of a sentence. Additionally, by using pooling methods, we are assuming that words and sentences can be represented using the same space size, which is a limitation of such a method (Wieting and Kiela, 2019).

One relatively recent advancement in STS, which accounts for the shortcomings of the pooling methods is the Siamese Recurrent Architecture (Mueller and Thyagarajan, 2016). By using two Recursive Neural Networks (RNNs), with shared weights, the pair of sentences are encoded into a higher dimensional space than the WVR used for the constituent tokens. Given the sequential nature of RNNs, this encoding takes into account the order of tokens in each sentence. The encoding is then fed into a feedforward dense neural network, with a value between 0 and 5 to predict the semantic similarity of the pair. One of the advantages of this method when it comes to inference, is that it can be used to produce a sentence level representation, which, with the use of a simple distance matrices, can be used to measure the similarity between two sentences without the need for the feedforward step (Neculoiu et al., 2016). This translates to much higher scalability in industrial applications. Another advantage is that it can be modified to account for errors in spelling (Neculoiu et al., 2016). Nonetheless, a major drawback of this method is that it requires a substantial amount of annotated data for training.

One method which overcome this limitation is Skip-thought Vectors (SV) (Kiros et al., 2015), which learn to embed text at the level of sentences, by training on continuous text (e.g. books and articles) in an unsupervised fashion. The representations can then used as feature inputs with the method of choice to predict the STS score. However, training SV requires very long period of time (it took about one month back in 2015 (Wieting and Kiela, 2019)).

One problem that most sequential deep learning methods suffer from is that the longer the sequence of text to encode is, the less efficient the representation becomes (Olah and Carter, 2016). This problem has been recently tackled by exploiting the attention mechanism in deep learning architectures. With the use of multi-head attention mechanism in constructing sentence embeddings, the state of the art of NLP in many STS dependent tasks has been significantly increased (Lin et al., 2017).

Another recent and novel development pertaining to STS, makes use of conversational data (Yang et al., 2018). The premise here is that sentences that are semantically related, will elicit similar responses in a conversation. However, an obvious shortcoming of such a method is that it is by design geared toward conversational tasks, as opposed to tasks that are factual by nature.

In a new research, (Al-Bataineh et al., 2019) tackles the issue of handling multiple dialects of the same language. The novel approach makes use of deep contextualized word embeddings (Peters et al., 2018) in addition to focus layer (He and Lin, 2016) to overcome out-of-vocabulary introduced by dialectical words.

As it stands now, the state of the art in STS are Universal Sentence Encoders (USE) (Cer et al., 2018). These encoders are trained on a wide variety of data types and tasks (i.e. using different signals such as entailment and SV like signals), with the idea of transfer learning at their heart. Under the hood, USEs can be powered by one of two deep learning architectures; the first is a transformer network, while the other is a deep averaging network. The main difference between these two versions, is that with the former, higher accuracies can be achieved, but with longer training times, whereas for the latter, training is less computationally intensive, at the expense of some accuracy in the final outcome.

Table 1: Mawdoo3 Q2Q dataset statistics.

Set	Similar	Not Similar	total
Train	5,397	6,600	11,997
Test	1,718	1,997	3,715
Total	7,115	8,597	15,712

3 Dataset

Despite the fact that there is a number of public datasets for QA in English language (such as SQuAD (Rajpurkar et al., 2016) and CoQA (Reddy et al., 2018) to name a few, there is a dearth of such datasets in Arabic. Therefore, we have developed a dataset[2] of 15, 712 pairs of questions, that were annotated and verified by an internal team of qualified natural language annotators. Each pair has a ground truth of either "0" (no semantic similarity), or "1" (denoting semantically similar pairs). We have randomly selected 11,997 pairs for training and used the remaining 3,715 for testing. We made sure that the collected data is balanced, where the number of similar question pairs is comparable with the not similar ones. Table 1 shows a detailed statistics of Mawdoo3 Q2Q dataset.

These questions were designed specifically to contain a balanced number of factoid and non-factoid questions. Additionally, great care was taken in assuring that the pairs of questions have varying STS and LS similarity, in a way that mimics the population of questions asked on the internet by Arabic language users. For example:

من هو رئيس الولَايَات المتحدة الأمريكية؟

which translates to "Who is the president of the United States of America?".

Table 1 lists a small sample of the dataset. The dataset consists of 3 fields, i.e. *question1* containing the text for one of the question pairs, *question2* containing the text of the second question, and *label* which is either 1 if question1 and question2 have a similar answer, or 0 if their answers are different. Figure 1 shows a histogram for a number of words per question against frequency. It can be seen that the maximum question length is 15 words and that the distribution of both *question1* and *question2* is almost the same.

[2] https://ai.mawdoo3.com/nsurl-2019-task8

Table 2: Sample of the Mawdoo3 Q2Q dataset. The dataset is composed of three columns. The first two are text fields containing question1 and question2 while the third column shows the label.

question1	question2	label
مَا هي الطرق الصحيحة لِّاعتنَاء بَالحَامل؟	كيف اهتم بطفلي؟	0
مَا طريقة تحضير محشي الكوسَا ؟	من طرق تحضير محشي الكوسَا؟	1
في أي عَام ولد توفيق الحكيم ؟	أين ولد توفيق الحكيم ؟	0
مَا طريقة تحضير اَالمهلبيخّة بجوز الهند؟	كيف احضر المهلبيخّة بجوز الهند؟	1
مَا طريقة تحضير الكيك المحشي بَالكريمة ؟	من طرق تحضير الكريمة؟	0
مَا هي حصوَات المرَارة؟	مَا هي حصَى المرَارة؟	1
كيف احضرَالمصَابيب مع المكشن؟	من طرق تحضير المصَابيب المحشي؟	0
مَا هو الموت؟	مَا أجمل مَا قيل بَالموت؟	0
في أي عَام بُني برج خليفة؟	أين يوجد برج خليفة؟	0
مَا طريقة تحضير عجينة البيتزَا بَالحليب ؟	من طرق تحضير عجينة البيتزَا ؟	0
مَا معنَى الجهَاد؟	مَا أنوَاع الجهَاد؟	0
لمَاذَا ميدَان بيكَاديلي يجذب الكثير من السيَاح؟	مَا اسم أهم معلم سيَاحي في بريطَانيَا؟	0
كم يبلغ طول تمثَال المسيح الفَادي؟	مَا هو طول التمثَال الفَادي؟	1
إِلَى كم يصل ارتفَاع أبو الهول الموجود في مصر؟	كم يبلغ عدّ سكَان مصر؟	0
من هو المدير العَام ؟	مَا هو تعريف المدير العَام ؟	1
مَا هي ادَارة الَاعمَال ؟	مَا هي مجَالَات ادَارة الَاعمَال ؟	0
مَاهو الكوليسترول؟	مَا تعريف الكوليسترول؟	1
مَا هي أهمية الَاستثمَار ؟	الَى مَاذَا يهدف الَاستثمَار ؟	1

4 Participants and Systems

The shared task was managed using a Kaggle competition platform[3] for registration and results submissions. We have published a baseline[4] that the participants can reproduce on the same dataset.

A total of 9 teams participated in this task, with total submissions of 547, and an average of more than 60 submissions per team. In this section, we report the methodologies used for four different teams.

4.1 The Inception

The Inception team members applied different deep learning approaches, including BERT model (Devlin et al., 2018). They fine-tuned the multilingual BERT model (Devlin et al., 2018) on the sentence similarity task.

They tried various combinations of hyperparameters. For the set of parameters that made the best predictions, they repeated the experiment with different random seeds, then created an ensemble model by voting between the prediction results of these experiments. The ensemble that is composed of 3 models performed better on the public dataset while 4, 5, and 6 models have better scores on the private dataset.

4.2 Tha3aroon

Tha3aroon team did heavy work on the dataset level before building the model. First, they made sure that punctuation marks are separated from the words by making sure that characters surrounding the punctuation marks are spaces. Next, they augmented the dataset 4 different methods:

- **Positive Transitive:** If A is similar to B, and B is similar to C, then A is similar to C.

[3] https://www.kaggle.com/c/nsurl-2019-task8

[4] https://github.com/mawdoo3/q2q_workshop

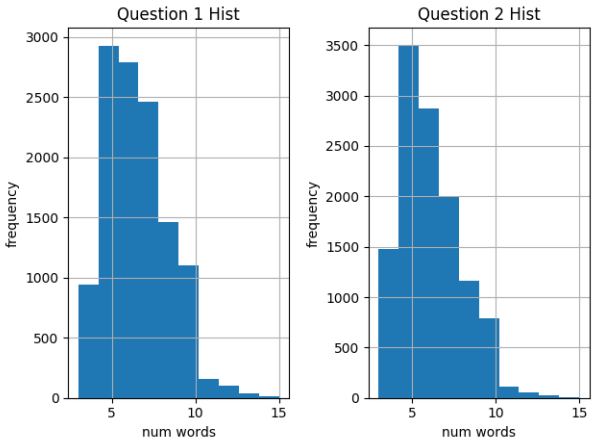

Figure 1: Distribution of question lengths (word count) in Mawdoo3 Q2Q dataset. The figure on the left shows Question 1 histogram, and Question 2 on the right.

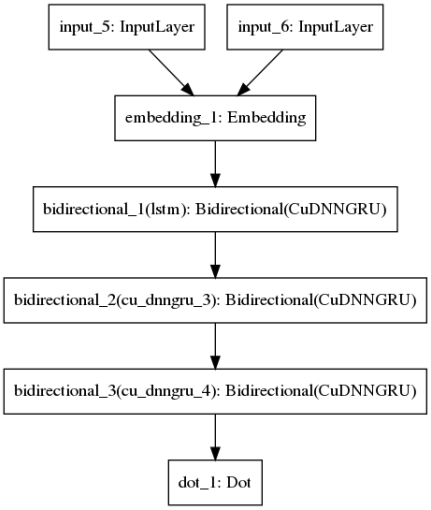

Figure 2: onekaggler model

- **Negative Transitive:** If A is similar to B, and B is NOT similar to C, then A is NOT similar to C. This rule combined with the previous one generates 5,490 extra examples (17,487 total).
- **Symmetric:** If A is similar to B then B is similar to A, and if A is not similar to B then B is not similar to A. This rule doubles the number of examples to 34,974 in total.
- **Reflexive:** By definition, a question A is similar to itself. This rule generates 10,540 extra positive examples (45,514 total) which help balance the positive and negative examples.

After the augmentation process, the training data contains 45,514 examples (23,082 positive examples and 22,432 negative ones).

To build meaningful representations for the input sequences, they used Arabic ELMo (Peters et al., 2018) pre-trained model [5] to extract contextual words embeddings and feed them as an input to the model. The model then consists of three components:

1. **Sequence representation extractor:** which takes the ELMo embeddings related to each word in the question as an input and feeds them to two special kinds of LSTM layers called Ordered Neurons LSTM (ON-LSTM) (Shen et al., 2018) and applies sequence weighted attention (Felbo et al., 2017) on the outputs of the second ON-LSTM layer to get the final question representation, this component uses the same weights to compute representations for pair questions.
2. **Merging layer:** After extracting the representations related to each question, they merged the representations using a pairwise squared distance function applied on the pair questions representation vectors.
3. **Deep neural network:** Consisting of four fully-connected layers that take the merged representation vector as an input and predicts the label using a sigmoid function as an output.

4.3 onekaggler

The onekaggler team has built a neural network model illustrated in Figure 2. The model consists of two input layers for question1 and question2, a shared trainable word embedding layer, using Word2Vec model (Mikolov et al., 2013), initialized with Aravec tweets_cbow_300 embedding model (Soliman et al., 2017), and a stack of 3 bidirectional GRU layers with 256, 128, 64 hidden nodes, respectively. The output layer is the dot product (which calculates cosine similarity) between the outputs of the last layer of question1 and question2. The team uses mean-squared-error as a loss function alongside with Nesterov Adam optimizer. They achieve 99% accuracy on the validation set and under 94% on the test set.

4.4 Speech Translation

The Speech Translation team members have gathered feature set using sklearn's Vector-

[5] https://github.com/HIT-SCIR/ELMoForManyLangs

izer Analyzer with three setups; `word-level`, `char-level`, and `char_Wb-level`. They have examined the use of n-grams (1, 2, 3, 4, and 5) for the three setups. As a preprocessing step, they applied punctuation removal, stop words filter, and text normalization. These features, combined with word stemming and POS tagging, are used for model training and testing. The team has compared the performance of a set of classifiers: BNB, LogReg, LSVM, MNB, PassAgg, PRP and SGD as well as CNN. The best performance is achieved by LSVM classifier.

5 Results and Discussion

Table 3 shows a summary of results for the participating teams. The Inception team has topped the list by achieving an accuracy score of 0.9592 using BERT models. ELMo model built by Tha3aroon scored second with an accuracy of 0.9485. This model was trained using the augmented dataset of 45,514 data samples. onekaggler team has scored third among all participants with 0.9481 accuracy using a stack of three Bidirectional GRUs. Speech Translation team has used 1 to 5 n-grams of words and characters and has experimented with several classifiers to score 0.8270, achieving the 7^{th}.

Table 3: Results for Semantic Question Similarity in Arabic. The table shows the 9 teams who participated in the workshop sorted in descending accuracy score.

#	Team Name	Score
1	The Inception	0.95924
2	Tha3aroon	0.94848
3	onekaggler	0.94809
4	Ayat Abedalla	0.91311
5	Dan Ofer	0.89465
6	heza	0.85736
7	Speech Translation	0.82698
8	AtyNegar	0.82583
9	Eyad Sibai	0.71434

One of the main takeaways is that BERT model accuracy is higher than ELMo model even when it was fine-tuned on an augmented dataset. The BERT model learns the representation of subwords while ELMo is character based model that uses convolution layers to learn word embeddings that handle out of vocabulary words. The reported results show that BERT is able to strike a good balance between a character based and word based representations and capture the word semantics for the problem of Arabic Q2Q.

Both of ELMo and BERT were able to outperform the traditional Word2Vec embeddings that is not able to capture contextual semantics nor learns subword embeddings. This proves that Arabic language (a morphologically rich language) complicates the training phase for such models because it needs to learn a completely new embedding for each morphology and is unable to generalize learnings across word variations. A word root in the Arabic language can have up to 1000 variation, Word2Vec needs to learn a number of weights equal to the number of variations multiplied by the vector size, while BERT and ELMo will only need to learn the word prefixes, roots, and word prefixes.

An interesting experiment would be to train BERT on the augmented data developed by Tha3aroon.

6 Conclusion

In this paper, we described the Arabic question similarity (Q2Q) shared the task that was organized in the workshop on NLP Solutions for Under Resourced Languages (NSURL 2019). The dataset of the shared task was made publicly available as a benchmark of this NLP task. A total of 9 teams participated in the task in which we provided a brief description of 4 of them who submitted their system description. The use of recent approaches in text embedding, i.e., BERT and ElMo, was a big factor in obtaining the best performing results. Another approach was using data augmentation that boosted up the performance. Also, an approach of using a neural network with Adam optimizer and an input layer that is initialized with pre-trained word vectors of the question pair was a well-performing solution. The ample number of participants in this workshop is an indication of the importance and interest in the Arabic language and Arabic semantic textual similarity. As future work, we would like to consider extending the task to news headlines as well as article titles.

7 Acknowledgement

We would like to thank Mawdoo3 AI data annotation team members who contributed to build and release Mawdoo3 Q2Q Dataset: Riham Badawi, Lana AlZaatreh, Raed AlRfouh, and Dana Barouqa. We would also like to thank Maw-

doo3[6] for making the datasets created for this task publicly available to support further research on Arabic Q2Q.

References

Hesham Al-Bataineh, Wael Farhan, Ahmad Mustafa, Haitham Seelawi, and Hussein T Al-Natsheh. 2019. Deep contextualized pairwise semantic similarity for arabic language questions. *arXiv preprint arXiv:1909.09490*.

Marwah Alian and Arafat Awajan. 2018. Arabic semantic similarity approaches–review. In *The 19th International Arab Conference on Information Technology (ACIT' 2018)*.

Daniel Cer, Yinfei Yang, Sheng-yi Kong, Nan Hua, Nicole Limtiaco, Rhomni St. John, Noah Constant, Mario Guajardo-Cespedes, Steve Yuan, Chris Tar, Yun-Hsuan Sung, Brian Strope, and Ray Kurzweil. 2018. Universal sentence encoder. *CoRR*, abs/1803.11175.

Daniel M. Cer, Mona T. Diab, Eneko Agirre, Iñigo Lopez-Gazpio, and Lucia Specia. 2017. Semeval-2017 task 1: Semantic textual similarity - multilingual and cross-lingual focused evaluation. *CoRR*, abs/1708.00055.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Bjarke Felbo, Alan Mislove, Anders Søgaard, Iyad Rahwan, and Sune Lehmann. 2017. Using millions of emoji occurrences to learn any-domain representations for detecting sentiment, emotion and sarcasm. *arXiv preprint arXiv:1708.00524*.

Hua He and Jimmy Lin. 2016. Pairwise word interaction modeling with deep neural networks for semantic similarity measurement. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 937–948.

Ryan Kiros, Yukun Zhu, Ruslan Salakhutdinov, Richard S. Zemel, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. 2015. Skip-thought vectors. In *Advances in Neural Information Processing Systems 28: Annual Conference on Neural Information Processing Systems 2015, December 7-12, 2015, Montreal, Quebec, Canada*, pages 3294–3302.

Zhouhan Lin, Minwei Feng, Cícero Nogueira dos Santos, Mo Yu, Bing Xiang, Bowen Zhou, and Yoshua Bengio. 2017. A structured self-attentive sentence embedding. *CoRR*, abs/1703.03130.

Goutam Majumder, Partha Pakray, Alexander F. Gelbukh, and David Pinto. 2016. Semantic textual similarity methods, tools, and applications: A survey. *Computación y Sistemas*, 20(4).

Tomas Mikolov, Ilya Sutskever, Kai Chen, Gregory S. Corrado, and Jeffrey Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Advances in Neural Information Processing Systems 26: 27th Annual Conference on Neural Information Processing Systems 2013. Proceedings of a meeting held December 5-8, 2013, Lake Tahoe, Nevada, United States.*, pages 3111–3119.

George A. Miller and Christiane Fellbaum. 2007. Wordnet then and now. *Language Resources and Evaluation*, 41(2):209–214.

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.

Preslav Nakov, Lluís Màrquez, Alessandro Moschitti, Walid Magdy, Hamdy Mubarak, Abed Alhakim Freihat, Jim Glass, and Bilal Randeree. 2016. SemEval-2016 task 3: Community question answering. In *Proceedings of the 10th International Workshop on Semantic Evaluation*, SemEval '16, San Diego, California. Association for Computational Linguistics.

Paul Neculoiu, Maarten Versteegh, and Mihai Rotaru. 2016. Learning text similarity with siamese recurrent networks. In *Proceedings of the 1st Workshop on Representation Learning for NLP, Rep4NLP@ACL 2016, Berlin, Germany, August 11, 2016*, pages 148–157.

Chris Olah and Shan Carter. 2016. Attention and augmented recurrent neural networks. *Distill*.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543.

Matthew E Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. *arXiv preprint arXiv:1802.05365*.

Mohammad Taher Pilehvar, David Jurgens, and Roberto Navigli. 2013. Align, disambiguate and walk: A unified approach for measuring semantic similarity. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics, ACL 2013, 4-9 August 2013, Sofia, Bulgaria, Volume 1: Long Papers*, pages 1341–1351.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. Squad: 100, 000+ questions for machine comprehension of text. *CoRR*, abs/1606.05250.

[6] ai.mawdoo3.com

Siva Reddy, Danqi Chen, and Christopher D. Manning. 2018. Coqa: A conversational question answering challenge. *CoRR*, abs/1808.07042.

Frane Saric, Goran Glavas, Mladen Karan, Jan Snajder, and Bojana Dalbelo Basic. 2012. Takelab: Systems for measuring semantic text similarity. In *Proceedings of the 6th International Workshop on Semantic Evaluation, SemEval@NAACL-HLT 2012, Montréal, Canada, June 7-8, 2012*, pages 441–448.

Yikang Shen, Shawn Tan, Alessandro Sordoni, and Aaron Courville. 2018. Ordered neurons: Integrating tree structures into recurrent neural networks. *arXiv preprint arXiv:1810.09536*.

Gizem Soganciogl u, Hakime Öztürk, and Arzucan Özgür. 2017. BIOSSES: a semantic sentence similarity estimation system for the biomedical domain. *Bioinformatics*, 33(14):i49–i58.

Abu Bakr Soliman, Kareem Eissa, and Samhaa R. El-Beltagy. 2017. Aravec: A set of arabic word embedding models for use in arabic nlp. *Procedia Computer Science*, 117:256 – 265. Arabic Computational Linguistics.

Radu Soricut and Eric Brill. 2004. Automatic question answering: Beyond the factoid. In *Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics, HLT-NAACL 2004, Boston, Massachusetts, USA, May 2-7, 2004*, pages 57–64.

John Wieting and Douwe Kiela. 2019. No training required: Exploring random encoders for sentence classification. *CoRR*, abs/1901.10444.

Yinfei Yang, Steve Yuan, Daniel Cer, Sheng-yi Kong, Noah Constant, Petr Pilar, Heming Ge, Yun-Hsuan Sung, Brian Strope, and Ray Kurzweil. 2018. Learning semantic textual similarity from conversations. In *Proceedings of The Third Workshop on Representation Learning for NLP, Rep4NLP@ACL 2018, Melbourne, Australia, July 20, 2018*, pages 164–174.