HAL
archives-ouvertes.fr

Improving the Community Question Retrieval Performance Using Attention-based Siamese LSTM

Nouha Othman, Rim Faiz, Kamel Smaïli

▶ To cite this version:

HAL Id: hal-02867309
https://hal.archives-ouvertes.fr/hal-02867309

Submitted on 14 Jun 2020

Improving the Community Question Retrieval Performance Using Attention-based Siamese LSTM

Nouha Othman[1,3], Rim Faiz[2], and Kamel Smaïli[3]

[1] LARODEC, University of Tunis, Tunisia
[2] LARODEC, University of Carthage, Tunisia
[3] LORIA, University of Lorraine, France
nouha.othman@loria.fr, rim.faiz@ihec.rnu.tn, kamel.smaili@loria.fr

Abstract. In this paper, we focus on the problem of question retrieval in community Question Answering (cQA) which aims to retrieve from the community archives the previous questions that are semantically equivalent to the new queries. The major challenges in this crucial task are the shortness of the questions as well as the word mismatch problem as users can formulate the same query using different wording. While numerous attempts have been made to address this problem, most existing methods relied on supervised models which significantly depend on large training data sets and manual feature engineering. Such methods are mostly constrained by their specificities that put aside the word order and ignore syntactic and semantic relationships. In this work, we rely on Neural Networks (NNs) which can learn rich dense representations of text data and enable the prediction of the textual similarity between the community questions. We propose a deep learning approach based on a Siamese architecture with LSTM networks, augmented with an attention mechanism. We test different similarity measures to predict the semantic similarity between the community questions. Experiments conducted on real cQA data sets in English and Arabic show that the performance of question retrieval is improved as compared to other competitive methods.

Keywords: Community question answering · Question retrieval · Siamese LSTM · Attention mechanism

1 Introduction

Community Question Answering (cQA) sites such as Yahoo! Answers[4], Stack-overflow[5], Quora[6], WikiAnswers[7], and Google Ejabat[8] give people the ability to post their various questions and get them answered by other users. Interestingly,

---

[4] http://answers.yahoo.com/
[5] http://stackoverflow.com/
[6] https://fr.quora.com/
[7] https://wiki.answers.com/
[8] https://ejaaba.com/

users can directly obtain short and precise answers rather than a list of potentially relevant documents. Community sites are exponentially growing over time, building up very huge archives of previous questions and their answers. However, multiple questions with the same meaning can make information seekers spend more time searching for the best answer to their question. Therefore, retrieving similar questions could greatly improve the QA system and benefit the community. Detecting similar previous questions that best match a new user's query is a crucial and challenging task in cQA, known as Question Retrieval (QR). Using the existing answers to similar previous questions could dodge the lag time incurred by waiting for new answers, thus enhancing user satisfaction. Owing to its importance, the question retrieval task has received wide attention over the last decade [14, 18, 17]. One critical challenge for this task is the word mismatch between the new posted questions and the existing ones in the archives as similar questions can be formulated using different, but related words. For instance, the questions *How can we relieve stress naturally?* and *What are some home remedies to help reduce feelings of anxiety?* have nearly the same meaning but include different words and then may be regarded as dissimilar. This constitutes a barrier to traditional Information Retrieval (IR) and Natural Language Processing (NLP) models since users can phrase the same query using different wording. Furthermore, community questions are mostly short, have different lengths, and usually have sparse representations with little word overlap. Although numerous attempts have been made to tackle this problem, most existing methods rely on the bag of-words (BOWs) representations which are constrained by their specificities that put aside the word order and ignore semantic and syntactic relationships. Recent advances in question retrieval have been achieved using Neural Networks (NNs) [5, 8, 12, 6] which provide powerful tools for modeling language, processing sequential data and predict the text similarity.

In this paper, we propose an approach based on NNs to detect the semantic similarity between the questions. The deep learning approach is based on a Siamese architecture with LSTM networks, augmented with an attention mechanism. We tested different similarity measures to compare the final hidden states of the LSTM layers.

2   Related Work

The question retrieval task has been intensively studied over the past decade. Early works were based on the vector space model referred to as VSM to calculate the cosine similarity between a query and archived questions [2]. However, the major limitation of VSM is that it favors short questions, while cQA services can handle a wide variety of questions not limited to factoïd questions. Language Models (LM)s [3] have been also used to model queries as sequences of terms instead of sets of terms. LMs estimate the relative likelihood for each possible successor term taking into account relative positions of terms. Nevertheless, such models might not be effective when there are only few common words between the questions. Further methods exploited the available category information of

questions such as in [2]. Wang et al [15] used a parser to build syntactic trees of questions, and rank them based on the similarity between their syntactic trees. Nonetheless, such an approach requires large training data and existing parsers are still not well-trained to parse informally written questions. Recent works focused on the representation learning for questions, relying on the Word Embedding model for learning distributed representations of words in a low-dimensional vector space. Along with the popularization of word embeddings and its capacity to produce distributed representations of words, advanced NN architectures such as Convolutional Neural Networks (CNN), Recurrent Neural Networks (RNN) and LSTM have proven effectiveness in extracting higher-level features from constituting word embeddings. For instance, Dos Santos et al. [5] employed CNN and bag-of-words (BOW) representations of the questions to calculate the similarity scores. Within the same context, Mohtarami et al. [8] developed a bag-of-vectors approach and used CNN and attention-based LSTMs to capture the semantic similarity between the community questions and rank them accordingly. LSTM model was also used in [12], where the weights learned by the attention mechanism were exploited for selecting important segments and enhancing syntactic tree-kernel models. More recently, the question retrieval task was modeled as a binary classification problem in [6] using a combination of LSTM and a contrastive loss function to effectively memorize the long term dependencies. In our work, we use a Siamese adaptation of LSTM [9] for pairs of variable-length sentences named Siamese LSTM. It is worth noting that work on cQA has been mostly carried out for other languages than Arabic mainly due to a lack of resources. Recent works in Arabic mainly rely on word embeddings and parse trees to analyze the context and syntactic structure of the questions [8, 1, 7, 13].

3   Description of the proposed ASLSTM approach

In order to improve the QR task, we propose an attentive Siamese LSTM approach for question retrieval, referred to as ASLSTM to detect the semantically similar questions in cQA. The approach is composed of three main modules namely, question preprocessing, word embedding learning and attentive Siamese LSTM. The basic principle underlying the ASLSTM approach is to map every question word token into a fix-sized vector. The word vectors of the questions are therefore fed to the Siamese LSTM with the aim of representing them in the final hidden states encoding semantic meaning of the questions. An attention mechanism is integrated in the Siamese architecture to determine which words should give more attention on than other words over the question. Community questions are then ranked by means of the Manhattan similarity function based on the vector representation of each question. A previous posted question is considered to be semantically equivalent to a queried question if their corresponding LSTM representations lie close to each other according to the Manhattan similarity measure. The historical question with the highest Manhattan score will be

returned as the most similar question to the new posted one. The components of ASLSTM and the dataset used are described below.

3.1   Dataset

We used the dataset released in [19] for the QR evaluation. The questions of the community collection were harvested from all categories in the Yahoo! Answers platform, and were randomly splitted into the test and search sets while maintaining their distributions in all categories. The community questions in the collection are in various structures, different lengths and belonging to diverse categories e.g., Health, Sports, Computers and Internet, Diet and Fitness, Pets, Travel, Business and Finance, Entertainment and Music etc. Table 1 gives some statistics on the experimental data set.

Table 1: Description of the Data Set

number of questions in the search set	1,123,034
number of queries in the test set	252
number of relevant questions in the test set	1,624
number of questions in the dev set	83
number of relevant questions in the dev set	644
questions' lengths (number of words)	[1;20]

For our experiments in Arabic, we translated the same English collection using Google Translation with a careful manual verification, as there is no large Arabic dataset available for the question retrieval task. Note that the Arabic collection includes exactly the same number of questions as the English set.

3.2   Question preprocessing

Pre-processing is important to make the question collections cleaner and easier to process. The question preprocessing module aims to filter the community questions and extract the useful terms in order to represent them in a formal way. It comprises text cleaning, tokenization, stopwords removal and stemming. Punctuation marks, non letters, diacritics, and special characters are removed. English letters are lowercased while dates are normalized to the token *date* and numerical digits are normalized to the token *num*. For the Arabic question collection, in addition to the aforementioned tasks, orthographic normalization was applied, including Tachkil removal, Tatweel removal, and letter normalization.

3.3   Word Embedding Learning

Word embeddings are low-dimensional vector representations of words, learned by harnessing large amounts of text corpora using shallow neural networks. In the word embedding learning module, we map every word into a fix-sized vector

using Word2Vec pretrained on an external corpus. For English word embedding training, we resorted to the publicly available word2vec vectors[9], with dimensionality of 300, that were trained on 100 billion words from Google News.

For the experiments in Arabic, we used the Yahoo!Webscope dataset[10], translated into Arabic including 1,256,173 questions with 2,512,034 distinct words. The Continuous Bag-of-Words (CBOW) model was used, as it has proven through experiments to be more efficient and outperform Skip gram on our dataset [10]. The training parameters of the CBOW model on the Arabic collection were set after several tests as follows:

- Size=300: feature vector dimension. We tested different values in the range $[50, 500]$ but did not get significant difference in terms of precision.
- Sample=1e-4: down sampling ratio for the redundant words in the corpus.
- Negative samples=25: number of noise words
- min-count=1: we set the minimum number of words to 1 to make sure we do not throw away anything.
- Context window=5: fixed window size.

3.4   Attentive Siamese LSTM

3.5   Siamese LSTM

The overall aim of Siamese LSTM is to compare a pair of sentences to decide whether or not they are semantically equivalent. Siamese LSTM uses the Siamese network [9] architecture which is known to have identical sub-networks LSTMleft and LSTMright that are passed vector representations of two sentences and return a hidden state encoding semantic meaning of the sentences. These hidden states are then compared using a similarity metric to return a similarity score.

In our work, Siamese LSTM was adapted to the context of question retrieval, that is to say, the sentence pairs become pairs of questions. LSTM learns a mapping from the space of variable length sequences d_{in} and encode the input sequences into a fixed dimension hidden state representation d_{rep}. More concretely, each question is represented as a word vector sequence and fed into the LSTM, which updates, at each sequence-index, its hidden state. The final state of LSTM for each question is a vector of d dimensions, which holds the inherent context of the question. Unlike vanilla RNN language models which predict next words, the given network rather compares pairs of sequences. A major feature of the Siamese architecture is the shared weights across the sub-networks, which reduce not only the number of parameters but also the tendency of overfitting.

To measure the similarity between the two question vectors, we tested several similarity measures and finally adapted the Manhattan one with which we acquired the best outcome as will be seen later in the next section.

---

[9] https://code.google.com/p/word2vec/

[10] The Yahoo! Webscope dataset Yahoo answers comprehensive questions and answers version 1.0.2, available at "http://research.yahoo.com/Academic_Relations"

The Manhattan similarity between the last hidden states of a sequence pairs $h^{(left)}$ and $h^{(right)}$ is computed as follows:

$$y = exp(- \parallel h^{(left)} - h^{(right)} \parallel_1) \tag{1}$$

For Siamese LSTM training, we employed the publicly available Quora Question Pairs dataset [11]. The given collection encompasses 400,000 samples of question duplicate pairs, where each sample has a pair of questions along with ground truth about their corresponding similarity (1: similar, 0: dissimilar). During LSTM training, we applied the Adadelta method for weights optimization to automatically decrease the learning rate. Gradient clipping was also used with a threshold value of 1.25 to avoid the exploding gradient problem [11]. The LSTM layers' size was set to 50 and the embedding layer's size to 300. We employed the back propagation and small batches of size equals 64, to reduce the cross-entropy loss and we resorted to the Mean Square Error (MSE) as a common regression loss function for prediction. We trained the model for several epochs to observe how the results varied with the epochs. We found out that the accuracy changed with the variation of the number of epochs but stabilized after epoch 25. The given parameters were set based on several empirical tests; each parameter was tuned separately on a development set to pick out the best one. Note that we used the same LSTM configuration for both languages.

3.6   Attention Mechanism

Attention mechanism with neural networks have recently achieved tremendous success in several NLP tasks [4, 12]. We assume that every word in a question contributes to the meaning of the whole question but the words do not have equal influential information. Thus, we should assign a probability to every word to determine how influential it is to the entire question.

The general architecture of the Siamese LSTM model augmented with an attention layer is illustrated in Fig 1, where the different constituent layers are shown from the input (question words) to the output (similarity score). Siamese LSTM model employs only the last hidden states of a sequence pair e.g., $h_5^{(a)}$ and $h_4^{(b)}$, which may ignore some information. To remedy this problem, in the attention layer, we used all hidden states $H = \{h_1, h_2, ..., h_L\}$, where h_i is the hidden state of the LSTM at time step i summarizing all the information of the question up to x_i and L denotes the length of the question. Note that $\alpha^{(a)}$ and $\alpha^{(b)}$ denote the weights of $LSTM_a$ and $LSTM_b$, respectively. Basically, the attention mechanism measures the importance of a word through a context vector. It computes a weight α_i for each word annotation h_i according to its importance. The final question representation r is the weighted sum of all the word annotations using the attention weights, computed by equation 4.

In the attention layer, a context vector u_h is introduced, which is randomly initialized and can be viewed as a fixed query, that allows to identify the infor-
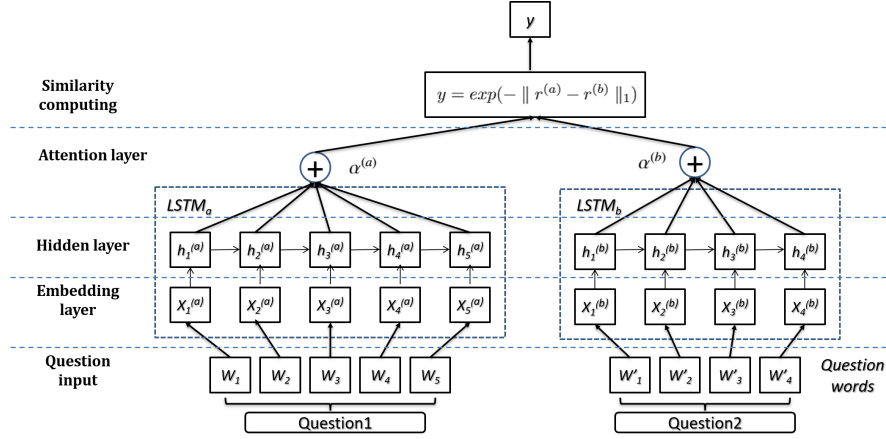
---
[11] www.kaggle.com/quora/question-pairs-dataset

Fig. 1: An illustration of Attentive Siamese LSTM model

mative words.

$$e_i = \tanh(W_h h_i + b_h), e_i \in [-1, 1] \tag{2}$$

$$\alpha_i = \frac{\exp(e_i^T u_h)}{\sum_{i=1}^{T} \exp(e_t^T u_h)}, \sum_{i=1}^{T} \alpha_i = 1 \tag{3}$$

$$r = \sum_{i=1}^{T} \alpha_i h_i, r \in R^{2L} \tag{4}$$

where W_h, b_h, and u_h are the learnable parameters, W_h is a weight matrix and b_h is a bias vector used to project each context vector into a common dimensional space and L is the size of each LSTM.

4   Experimental Evaluation

4.1   Evaluation Metrics

For the automatic evaluation, we used the following metrics: Mean Average Precision (MAP), Precision@n (P@n) and Recall as they are the most used ones for assessing the performance of the QR task. MAP assumes that the user is interested in finding many relevant questions for each query and then rewards methods that not only return relevant questions early, but also get good ranking of the results. Precision@n gives an idea about the classifier's ability of not labeling a positive sample as a negative one. It returns the proportion of the top-n retrieved questions that are equivalent. Recall is the measure by which we check how well the model is in finding all the positive samples of the dataset. It returns the proportion of relevant similar questions that have been retrieved over the total number of relevant questions. We also used accuracy, which returns the proportion of correctly classified questions as relevant or irrelevant.

4.2   Results and Discussion

We compare ASLSTM against our previous approach called WEKOS as well as the competitive state-of-the-art question retrieval methods tested in [19] on the same datasets. The methods being compared are briefly described below:

- **WEKOS** [10]: A word embedding based method which uses the cosine distance to measure the similarity between the weighted continuous valued vectors of the clustered questions.
- **TLM** [16]: A translation based language model which uses a query likelihood approach for the question and the answer parts, and integrates word-to-word translation probabilities learned through various information sources.
- **ETLM** [14]: An entity based translation language model, which is an extension of TLM where the word translation was replaced with entity translation to integrate semantic information within the entities.
- **PBTM** [20]: A phrase based translation model which uses machine translation probabilities assuming that QR should be performed at the phrase level.
- **WKM** [22]: A world knowledge based model which integrates the knowledge of Wikipedia into the questions by deriving the concept relationships that allow to identify related topics between the questions.
- **M-NET** [21]: A word embedding based model, which integrates the category information of the questions to get a category based word embedding.
- **ParaKCM** [19]: A key concept paraphrasing based approach which explores the translations of pivot languages and expands queries with paraphrases.

Table 2 gives a comparison of the performance of ASLSTM against the afore-mentioned models on the English Yahoo! Answers dataset.

As illustrated in Table 2, ASLSTM outperforms in English all the compared methods on all criteria by successfully returning a significant number of similar questions among the retrieved ones. This good performance indicates that the use of Siamese LSTM along with the attention mechanism is effective in the QR task. Word embeddings allow to obtain an efficient input representation for LSTM, capturing syntactic and semantic information in a word level.

Table 2:   Question retrieval performance comparison of different models in English.

|	TLM	ETLM	PBTM	WKM	M-NET	ParaKCM	WEKOS	ASLSTM
P@5	0.3238	0.3314	0.3318	0.3413	0.3686	0.3722	0.4338	**0.5033**
P@10	0.2548	0.2603	0.2603	0.2715	0.2848	0.2889	0.3647	**0.4198**
MAP	0.3957	0.4073	0.4095	0.4116	0.4507	0.4578	0.5036	**0.5799**

Interestingly, our approach does not require an extensive feature generation owing to the use of a pre-trained model. The results show that ASLSTM performs

better than translation and knowledge based methods, which provides evidence that the question representations made by the Siamese LSTM sub-networks can learn the semantic relatedness between pairs of questions and then are more adequate for representing questions in the question similarity task. The Siamese network was trained using backpropagation-through-time under the MSE loss function which compels the LSTM sub-networks to detect textual semantic difference during training. A key virtue of LSTM is that it can accept variable length sequences and map them into fixed length vector representations which can overcome the length and structure's problems in cQA.

Another significant finding is the effectiveness of the attention mechanism which was able to improve the performance of the approach. We assume that the attention mechanism managed to boost the similarity learning process by assigning a weight to each element of the question. The weights will then allow to compute which element in the sequence the neural network should more attend.

WEKOS averages the weighted embeddings, which is one of the most simple and widely used techniques to derive sequence embedding but it leads to losing the word order, while in ASLSTM, the LSTMs update their state to get the main context meaning of the text sequence in the order of words. The goal of the Siamese architecture is to learn a function which can map a question to an appropriate fixed length vector which is favor for similarity measurement. Interestingly, it offers vector representation for a very short text fragment that should grasp most of the semantic information in that fragment.

In order to properly assess the Siamese LSTM model performance on the similarity prediction problem, we plot training data vs validation data accuracy using the Matplotlib library.
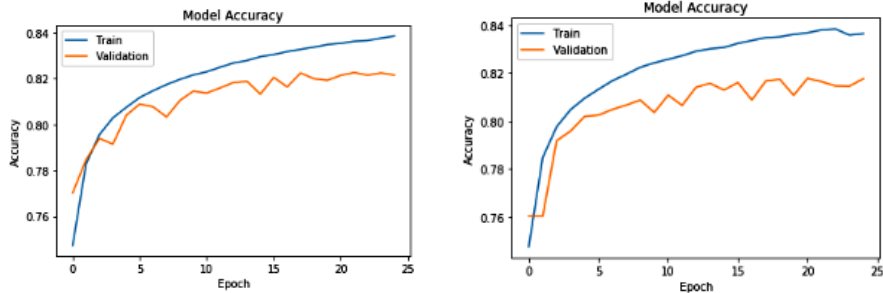
(a) Results on the English dataset

(b) Results on the Arabic dataset

Fig. 2: Epochs vs Accuracy of Siamese LSTM on the English and Arabic dataset

From the plots of accuracy given in Figures 2a and 2b, we observe that we get about 82% and 81% accuracy rate on the validation data for English and Arabic respectively. The model has comparable consistent accuracy on both train and validation sets. Both training and validation accuracy continue to

increase without a sudden decrease of the validation accuracy, indicating a good fit. Therefore, we can admit that, whilst the performance on the training set is slightly better than that of the validation set in term of accuracy, the model converged to a stable value without any typical overfitting signs.

It is worth mentioning that the accuracy used in the epochs-accuracy plots, is the binary accuracy calculated by Keras, and it implies that the threshold is set at 0.5 so, everything above 0.5 will be considered as correct.

Our results are fairly stable across different similarity functions, namely cosine and Euclidean distances. We found that the Manhattan distance outperformed them on both the English and Arabic datasets as depicted in Tables 3a and 3b which demonstrates that it is the most relevant measure for the case of high dimensional text data.

Table 3: Comparison between similarity measures

(a) Results on the English dataset

	P@5	Recall
Manhattan	0.5033	0.5477
Cosine	0.3893	0.4345
Euclidean	0.3393	0.3843

(b) Results on the Arabic dataset

	P@5	Recall
Manhattan	0.3702	0.4146
Cosine	0.2562	0.3006
Euclidean	0.2062	0.2506

Furthermore, we remarked that ASLSTM could find the context mapping between certain expressions mostly used in the same context such as *bug* and *error message* or also *need help* and *suggestions*. ASLSTM was also able to retrieve similar questions containing certain common misspelled terms like *recieve* instead of *receive*, but it failed to capture other less common spelling mistakes like *relyable* or *realible* instead of *reliable*. Such cases show that our approach can address some lexical disagreement problems. Moreover, there are few cases where ASLSTM fails to detect semantic equivalence, including queries having only one similar question and most words of this latter do not appear in a similar context with those of the query.

Table 4: Question retrieval performance of ASLSTM in Arabic

	WEKOS	ASLSTM
P@5	0.3444	0.3702
P@10	0.2412	0.2872
MAP	0.4144	0.4540
Recall	0.3828	0.4146

Table 4 shows that ASLSTM outperforms in Arabic the best compared system which proves that it can also perform well with complex languages.

Nevertheless, a major limitation of the proposed approach is that it ignores the morphological structure of Arabic words. Harnessing the word internal struc-

ture might help to capture semantically similar words. Therefore, endowing word embeddings with grammatical information such as, the person, gender, number and tense could help to obtain more meaningful embeddings that detect morphological and semantic similarity. In terms of recall, ASLSTM reaches 0.4136 for Arabic which implies that the number of omitted similar questions is not too big. Interestingly, unlike traditional RNNs, Siamese LSTM is able to effectively handle the long questions and learn long range dependencies thanks to its use of memory cell units that can store information across long input sequences.

4.3 Conclusion

In this paper, we presented an Attention-based Siamese LSTM approach, aiming at solving the question retrieval problem, which is of great importance in real-world cQA. For this purpose, we suggested using Siamese LSTM to capture the semantic similarity between the community questions. An attention mechanism was integrated to let the model give different attention to different words while modeling questions. Interestingly, we showed that Siamese LSTM is capable of modeling complex structures and covering the context information of question pairs. Experiments on large scale Yahoo! Answers datasets showed that the proposed approach can successfully improve the question retrieval task in English and Arabic and outperform some competitive methods evaluated on the same dataset. In the future, we plan to integrate morphological features into the embedding layer to improve the question representations.

References

1. Barrón-Cedeno, A., Da San Martino, G., Romeo, S., Moschitti, A.: Selecting sentences versus selecting tree constituents for automatic question ranking. In: Proc. of COLING, the 26th International Conference on Computational Linguistics. pp. 2515–2525 (2016)
2. Cao, X., Cong, G., Cui, B., Jensen, C.S.: A generalized framework of exploring category information for question retrieval in community question answer archives. In: Proc. of the 19th international conference on WWW. pp. 201–210. ACM (2010)
3. Cao, X., Cong, G., Cui, B., Jensen, C.S., Zhang, C.: The use of categorization information in language models for question retrieval. In: Proc. of the 18th conference on Information and knowledge management. pp. 265–274. ACM (2009)
4. Chorowski, J.K., Bahdanau, D., Serdyuk, D., Cho, K., Bengio, Y.: Attention-based models for speech recognition. In: Advances in neural information processing systems. pp. 577–585 (2015)
5. Dos Santos, C., Barbosa, L., Bogdanova, D., Zadrozny, B.: Learning hybrid representations to retrieve semantically equivalent questions. In: Proc. of ACL and the 7th International Joint Conference on NLP. vol. 2, pp. 694–699 (2015)
6. Kamineni, A., Shrivastava, M., Yenala, H., Chinnakotla, M.: Siamese lstm with convolutional similarity for similar question retrieval. In: 2018 International Joint Symposium on Artificial Intelligence and NLP. pp. 1–7. IEEE (2019)
7. Malhas, R., Torki, M., Elsayed, T.: Qu-ir at semeval 2016 task 3: Learning to rank on arabic community question answering forums with word embedding. In: Proc. of SemEval. pp. 866–871 (2016)

8. Mohtarami, M., Belinkov, Y., Hsu, W.N., Zhang, Y., Lei, T., Bar, K., Cyphers, S., Glass, J.: SLS at semeval-2016 task 3: Neural-based approaches for ranking in community question answering. In: Proc. of SemEval. pp. 828–835 (2016)
9. Mueller, J., Thyagarajan, A.: Siamese recurrent architectures for learning sentence similarity. In: Thirtieth AAAI Conference on Artificial Intelligence (2016)
10. Othman, N., Faiz, R., Smaïli, K.: Enhancing question retrieval in community question answering using word embeddings. In: Proc. of the 23rd International Conference on Knowledge-Based and Intelligent Information and Engineering Systems (KES) (2019)
11. Pascanu, R., Mikolov, T., Bengio, Y.: On the difficulty of training recurrent neural networks. In: International conference on machine learning. pp. 1310–1318 (2013)
12. Romeo, S., Da San Martino, G., Barrón-Cedeno, A., Moschitti, A., Belinkov, Y., Hsu, W.N., Zhang, Y., Mohtarami, M., Glass, J.: Neural attention for learning to rank questions in community question answering. In: Proc. of COLING. pp. 1734–1745 (2016)
13. Romeo, S., Da San Martino, G., Belinkov, Y., Barrón-Cedeño, A., Eldesouki, M., Darwish, K., Mubarak, H., Glass, J., Moschitti, A.: Language processing and learning models for community question answering in arabic. IPM (2017)
14. Singh, A.: Entity based q&a retrieval. In: Proc. of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning. pp. 1266–1277. ACL (2012)
15. Wang, K., Ming, Z., Chua, T.S.: A syntactic tree matching approach to finding similar questions in community-based qa services. In: Proc. of the 32nd international ACM SIGIR conference on Research and development in information retrieval. pp. 187–194. ACM (2009)
16. Xue, X., Jeon, J., Croft, W.B.: Retrieval models for question and answer archives. In: Proc. of the 31st annual international ACM SIGIR conference on Research and development in information retrieval. pp. 475–482. ACM (2008)
17. Ye, B., Feng, G., Cui, A., Li, M.: Learning question similarity with recurrent neural networks. In: 2017 IEEE International Conference on Big Knowledge (ICBK). pp. 111–118. IEEE (2017)
18. Zhang, K., Wu, W., Wu, H., Li, Z., Zhou, M.: Question retrieval with high quality answers in community question answering. In: Proceedings of the 23rd ACM International Conference on Information and Knowledge Management. pp. 371–380. ACM (2014)
19. Zhang, W.N., Ming, Z.Y., Zhang, Y., Liu, T., Chua, T.S.: Capturing the semantics of key phrases using multiple languages for question retrieval. IEEE Transactions on Knowledge and Data Engineering **28**(4), 888–900 (2016)
20. Zhou, G., Cai, L., Zhao, J., Liu, K.: Phrase-based translation model for question retrieval in community question answer archives. In: Proc. of the 49th Annual Meeting of the ACL: Human Language Technologies-Volume 1. pp. 653–662. ACL (2011)
21. Zhou, G., He, T., Zhao, J., Hu, P.: Learning continuous word embedding with metadata for question retrieval in community question answering. In: Proceedings of the 53rd Annual Meeting of the ACL and the 7th International Joint Conference on Natural Language Processing of the Asian Federation of Natural Language Processing. pp. 250–259 (2015)
22. Zhou, G., Liu, Y., Liu, F., Zeng, D., Zhao, J.: Improving question retrieval in community question answering using world knowledge. In: IJCAI. vol. 13, pp. 2239–2245 (2013)