



**HAL**  
open science

# Manhattan Siamese LSTM for Question Retrieval in Community Question Answering

Nouha Othman, Rim Faïz, Kamel Smaïli

► **To cite this version:**

Nouha Othman, Rim Faïz, Kamel Smaïli. Manhattan Siamese LSTM for Question Retrieval in Community Question Answering. The 18th International Conference on Ontologies, DataBases, and Applications of Semantics, Oct 2019, Rhodès, Greece. hal-02271338

**HAL Id: hal-02271338**

**<https://hal.science/hal-02271338>**

Submitted on 26 Aug 2019

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# Manhattan Siamese LSTM for Question Retrieval in Community Question Answering

Nouha Othman<sup>1</sup>, Rim Faiz<sup>2</sup>, and Kamel Smaili<sup>3</sup>

<sup>1</sup> LARODEC, University of Tunis, Tunisia

<sup>2</sup> LARODEC, University of Carthage, Tunisia

<sup>3</sup> LORIA, University of Lorraine, France

othmannouha@gmail.com, rim.faiz@ihec.rnu.tn, kamel.smaili@loria.fr

**Abstract.** Community Question Answering (cQA) are platforms where users can post their questions, expecting for other users to provide them with answers. We focus on the task of question retrieval in cQA which aims to retrieve previous questions that are similar to new queries. The past answers related to the similar questions can be therefore used to respond to the new queries. The major challenges in this task are the shortness of the questions and the word mismatch problem as users can formulate the same query using different wording. Although question retrieval has been widely studied over the years, it has received less attention in Arabic and still requires a non trivial endeavour. In this paper, we focus on this task both in Arabic and English. We propose to use word embeddings, which can capture semantic and syntactic information from contexts, to vectorize the questions. In order to get longer sequences, questions are expanded with words having close word vectors. The embedding vectors are fed into the Siamese LSTM model to consider the global context of questions. The similarity between the questions is measured using the Manhattan distance. Experiments on real world Yahoo! Answers dataset show the efficiency of the method in Arabic and English.

**Keywords:** Community question answering · Question retrieval · Word embeddings · Siamese LSTM

## 1 Introduction

Community Question Answering (cQA) platforms such as Yahoo! Answers<sup>4</sup>, Stackoverflow<sup>5</sup>, WikiAnswers<sup>6</sup>, Quora<sup>7</sup> and Google Ejabat<sup>8</sup> have become increasingly popular in recent years. Unlike traditional Question Answering (QA), users can interact and respond to other users' questions or post their own questions for other participants to answer. However, with the sharp increase of the cQA

---

<sup>4</sup> <http://answers.yahoo.com/>

<sup>5</sup> <http://stackoverflow.com/>

<sup>6</sup> <https://wiki.answers.com/>

<sup>7</sup> <https://fr.quora.com/>

<sup>8</sup> <https://ejaaba.com/>

archives, numerous duplicated questions have been amassed. Retrieving relevant previous questions that best match a new user’s query is a crucial task in cQA, known as question retrieval. If good matches are found, the answers to similar past questions can be used to answer the new query. This can avoid the lag time incurred by waiting for other users to respond, thus improving user satisfaction. The question retrieval task has recently sparked great interest [21, 3, 2, 19, 24, 22]. One big challenge for this task is the word mismatch between the queried questions and the existing ones in the archives [21]. Word mismatch means that similar questions can be phrased such that they have different, but related words. For example, the questions *How can we relieve stress naturally?* and *What are some home remedies to help reduce feelings of anxiety?* like in Arabic: كيف يمكننا تخفيف التوتر بشكل طبيعي؟

وماهي العلاجات المنزلية التي تساعد على تقليل الشعور بالقلق؟

have nearly the same meaning but different words and then may be regarded as dissimilar. This constitutes a barricade to traditional Information Retrieval (IR) models since users can formulate the same question employing different wording. Moreover, community questions have variable lengths, mostly short and usually have sparse representations with little word overlap. While many attempts have been made to dodge 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 syntactic and semantic relationships. Recent successes in question retrieval have been obtained using Neural Networks (NNs) [5, 12, 17, 9] which use a deep analysis of words and questions to take into account the semantics as well as the structure of questions in order to predict the text similarity. Motivated by the tremendous success of these emerging models, in this paper, we propose an approach based on NNs to detect the semantic similarity between the questions. The community questions are expanded with words having close embedding vectors. We use a variation of Long Short-Term Memory (LSTM) called Manhattan LSTM (MaLSTM) to analyze the entire question based on its words and its local contexts and predict the similarity between questions. We tested the proposed method on a large-scale real data from Yahoo! Answers in Arabic and English.

The remainder of this paper is structured as follows: Section (2) reviews the main related work on question retrieval in cQA. Section (3) describes our proposed LSTM based approach. Section (4) presents our experimental settings and discusses the obtained results. Section (5) concludes the paper and outlines areas for future research.

## 2 Related Work

Recently, a whole host of methods have been proposed to address the question retrieval task.

Early works were based on the vector space model referred to as VSM to calculate the cosine similarity between a query and archived questions [6, 3].

Nevertheless, the main limitation of VSM is that it favors short questions, while cQA services can handle a wide variety of questions not limited to factoid questions. In order to overcome this shortcoming, Okapi BM stands for Best Matching (Okapi BM25) has been used by search engines to estimate the relevance of questions to a given search query taking into account the question length [3]. Language Models (LM)s [4] 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. However, such models might not be effective when there are few common words between the questions. In order to handle the vocabulary mismatch problem faced by LMs, a model based on the concept of machine translation, referred in the following as translation model, was employed to learn correlation between words based on parallel corpora and it has obtained significant performance for question retrieval. The intuition behind translation-based models is to consider question-answer pairs as parallel texts then, relationships of words can be built by learning word-to-word translation probabilities like in [21, 2]. Within this context, Zhou et al. [26] attempted to enhance the word-based translation model by adding some contextual information when translating the phrases as a whole, instead of translating separate words. Singh et al. [19] extended the word-based translation model by integrating semantic information and explored strategies to learn the translation probabilities between words and concepts using the cQA archives and an entity catalog. Even though the above-mentioned basic models have yielded interesting results, questions and answers are not parallel in practice, rather they are different from the information they contain [24]. Further methods based on semantic similarity were proposed for question retrieval toward a deep understanding of short text to detect the equivalent questions. For instance, there have been a handful of works that have exploited the available category information for question retrieval such as in [4, 3, 27]. Although these attempts have proven to improve the performance of the language model for question retrieval, the use of category information was restricted to the language model. Wang et al [20] 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 a lot of training data and existing parsers are still not well-trained to parse informally written questions. Latent Semantic Indexing (LSI) was also used to address the given task like in [16]. Although LSI has proven to be effective in addressing the polysemy and synonymy by mapping terms related to the same concept close to each other, the efficiency of LSI depends on the data structure and both its training and inference are computationally expensive on large vocabularies. Recent works focused on the representation learning for questions, relying on an emerging model for learning distributed representations of words in a low-dimensional vector space called Word Embedding. This latter has recently been subject of a burgeoning interest and has shown great promise in several NLP tasks, As we believe that the representation of words is crucial for retrieving similar questions, we rely on word embeddings to represent the community questions. 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 cosine similarity scores. Within the same context, Mohtarami et al. [12] 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 [17] with an attention mechanism for capturing long dependencies in questions. Interestingly, the weights learned by the attention model 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 [9] 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 [13] for pairs of variable-length sentences named MaLSTM. This latter has accomplished excellent outcomes in the semantic text similarity task and inspire us in our question retrieval problem.

It is worth noting that work on cQA has been mostly carried out for other languages than Arabic. The most promising approach [12] used text similarities at both sentence and word level based on word embeddings. The similarities were computed between new and previous question, and between the new question and the answer related to the previous question  $p$ . A tree-kernel-based classifier was employed in [1] where the authors used supervised and unsupervised models that operated both at sentence and chunk levels for parse tree based representations. A supervised learning approach was adopted in [10] where learning-to-rank models were trained over word2vec features and covariance word embedding features produced from the training data. More recently, the given task was investigated by Romeo et al. [18] using advanced Arabic text representations made by applying tree kernels to constituency parse trees along with word embeddings and textual similarities.

### 3 Description of LSTMQR

We propose an approach called LSTM based Question Retrieval (LSTMQR) to retrieve the semantically similar questions in cQA. As depicted in Figure 1, our approach is composed of four main modules namely, question preprocessing, word embedding learning, question expansion and Manhattan LSTM (MaLSTM).

By and large, the basic intuition behind LSTMQR is to expand the filtered questions with words having close embedding vectors in order to have longer and richer word sequences. The word vectors of the expanded questions are fed to the Siamese LSTM to represent them in final hidden state encoding semantic meaning of the questions. Community questions are then ranked using the Manhattan similarity function based on the vector representation for each question.

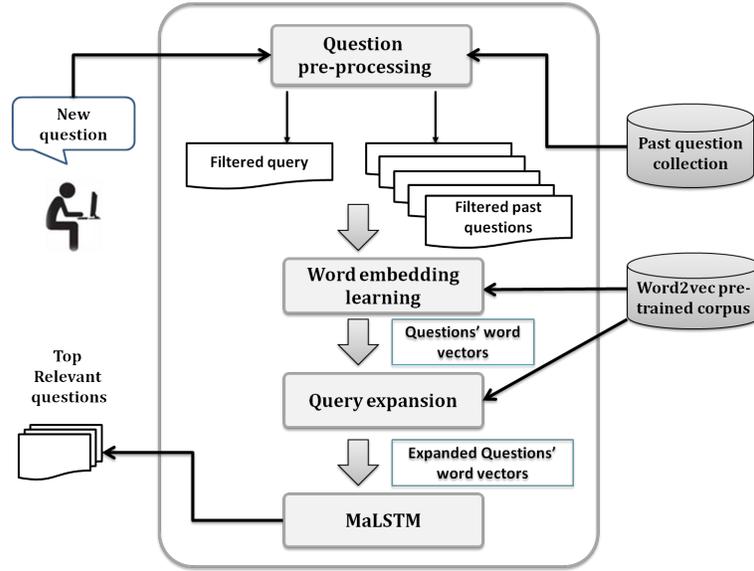


Fig. 1. LSTMQR pipeline for question retrieval in cQA

A previous posted question is considered to be semantically similar to a queried question if their corresponding LSTM representations lie close to each other according to the Manhattan similarity measure. The previous question with the highest Manhattan score will be returned as the most similar question to the new posted one. The components of LSTMQR are detailed below.

### 3.1 Question Preprocessing

Preprocessing is a crucial step in NLP tasks to assess and improve the quality of text data in order to ensure the reliability and validity of the statistical analysis. Our question preprocessing module intends to filter the natural language questions and extract the useful terms to represent them in a standard way. This module essentially encompasses text cleaning, tokenization, stopwords removal and stemming. We also remove punctuation marks, non letters, diacritics, and special characters such as &, #, \$ and £. English letters are lowercased while dates are normalized to the token *date* and numerical digits are normalized to the token *num*. At the end of the question preprocessing module, we obtain a set of filtered queries, each of which is formally defined as follows:  $q = \{t_1, t_2, \dots, t_Q\}$  where  $t$  represents a separate term of the query  $q$  and  $Q$  denotes the number of query terms. As for the Arabic question collection, in addition to the aforementioned tasks, orthographic normalization is required to reduce noise and ambiguity in the Arabic text data. This task includes Tachkil removal (ignoring arabic short vowels), Tatweel removal (deleting stretching symbol), and Letter

normalization (variant forms to one form conversion). Indeed, different spelling variants are sometimes inconsistently misused by writers, such as the Hamza; some may ignore it or employ a wrong Hamza variant. Hence, we normalize to one standard variant as follows: «أ، إ، آ، ؤ، ء، ء، ئ» are normalized to «ا». For example, people always write المروءة instead of المروءة. We then normalize it as follows: المرواة. In this way, words containing miswritten Hamzas will not be ignored.

### 3.2 Word Embedding Learning

Word embeddings are low-dimensional vector representations of words, learned by exploiting large amounts of text corpora using a shallow neural network. The word representations in a multidimensional space allow to capture the semantic and syntactic similarities between the corresponding words [11]. Two types of word embeddings were defined; Continuous Bag-of-Words model (CBOW) and Skip-gram model. The former one consists in predicting a current word according to its context, while the second does the inverse predicting the contextual words given a current pivot word in a sliding window. In the word embedding learning module, we map every word into a fix-sized vector using Word2Vec pretrained on an external corpus.

### 3.3 Question Expansion

One of the most challenges in the question retrieval task is the shortness of the community questions which leads to the word mismatch problem. To overcome this and improve the retrieval performance, we propose to add terms to the community questions. The additional words are those having similar embedding vectors. The number of the additional similar words is set as a variable parameter  $N_{sw}$ . More precisely, for each distinct word in a given question, we look for the  $N_{sw}$  words that have similar word vectors from the vocabulary and we add them to the question in order to have an expanded one while maintaining the order of words in the sequences. For example, let's consider the original question *Do chocolate really kill my dog?* containing 3 distinct words and  $N_{sw}$  is set to 3, the expanded query after preprocessing will have 12 words as follows: **chocolate kill dog eat death bitch candy toxic puppy food sick animal**. Similarly, we give an example of an Arabic query: الشوكولاته تقتل الكلب and its corresponding expanded version: الشوكولاته تقتل الكلب اكل موت كبة حلوى سامة جرو غذاء مريضة حيوان

We assume that expanding queries with additional words used in similar contexts may increase the chances of detecting equivalent questions.

### 3.4 Manhattan LSTM

Long Short-Term Memory (LSTM)[8], which is a powerful type of RNN used in deep learning, has gained wide attention in recent years owing to its capacity to

capture long-term dependencies and model sequential data. LSTM helps prevent the vanishing gradient problem [7] which is the main limitation of RNN. It is endowed with a memory to maintain its state over time, and internal mechanisms called gates which regulate the information flow. The main reason for choosing LSTM in our work is its proven performance in handling variable-length sequential data. Given input vector  $x_t$ , hidden state  $h_t$  and memory state  $c_t$ , the updates in LSTM are performed as follows:

$$i_t = \text{sigmoid}(W_i x_t + U_i h_{t-1} + b_i) \quad (1)$$

$$f_t = \text{sigmoid}(W_f x_t + U_f h_{t-1} + b_f) \quad (2)$$

$$\tilde{c}_t = \text{tanh}(W_c x_t + U_c h_{t-1} + b_c) \quad (3)$$

$$c_t = i_t \odot \tilde{c}_t + f_t \odot c_{t-1} \quad (4)$$

$$o_t = \text{sigmoid}(W_o x_t + U_o h_{t-1} + b_o) \quad (5)$$

$$h_t = o_t \odot \tanh(c_t) \quad (6)$$

where  $i_t$ ,  $f_t$ ,  $o_t$  are input, forget, and output gates at time  $t$ , respectively.  $W_k$ ,  $U_k$  are LSTM parameterized weight matrices,  $b_k$  represents the bias vector for each  $k$  in  $\{i, f, c, o\}$  and  $\odot$  denotes an element-wise product of matrices, known as the Hadamard product which is simply an entrywise multiplication.

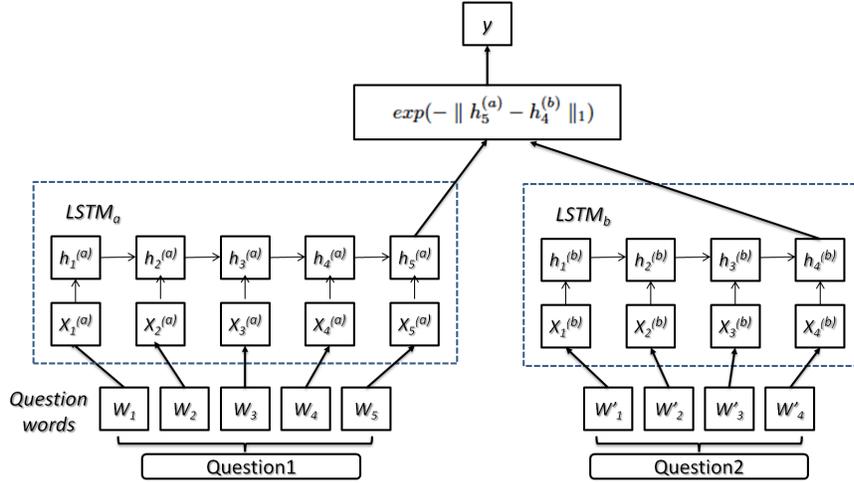


Fig. 2. General architecture of the MaLSTM model

The Manhattan LSTM (MaLSTM) refers to the fact that the Manhattan distance is used to compare the final hidden states of two standard LSTM layers instead of another distance such as Cosine and Euclidean distances. The overall

aim of MaLSTM is to compare a pair of sentences to decide whether or not they are semantically equivalent. MaLSTM uses the Siamese network [13] 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 as depicted in Figure 2. Note that we decided to use LSTM for each sub-network, but it is also possible to swap LSTM with GRU. In our work, MaLSTM 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  $d_{rep}$ . In other words, each question represented as a word vector sequence (e.g.,  $Q_1$  is represented by  $x_1, x_2, x_3$ ) is fed into the LSTM, which updates, at each sequence-index, its hidden state. The final state of LSTM for each question is a  $d_{rep}$ -dimensional vector, denoted by  $h$  in figure 2, which holds the semantic meaning of the question.

Unlike other language modeling RNN architectures which predict next words, the given network rather computes the similarity between pairs of sequences. A main 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. MaLSTM uses the Siamese structure along with the Manhattan distance, hence the name MaLSTM model. Once we have the two vectors that capture the underlying meaning of each question, we calculate the similarity between them using the following Manhattan similarity function:

$$y = \exp(- \| h^{(left)} - h^{(right)} \|_1) \quad (7)$$

Note that since we have an exponent of a negative, the Manhattan function scores will be between 0 and 1.

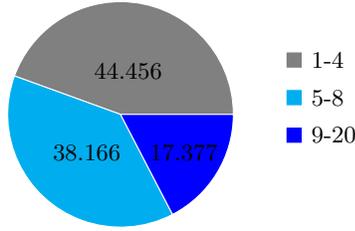
## 4 Experimental Evaluation

### 4.1 Datasets

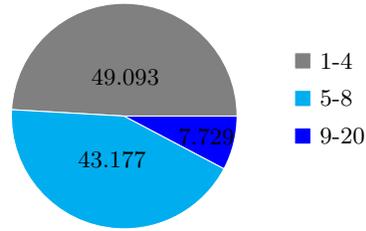
We performed experiments using the dataset released by [25] for evaluation. To build the dataset, the authors crawled questions from all categories in the popular Yahoo! Answers community platform, and then randomly splitted the questions into two sets while maintaining their distributions in all categories.

The former set is a question repository for question search containing 1,123,034 questions, while the second is the test set containing 252 queries and 1624 manually annotated related questions. Note that the number of relevant questions related to each original query varies from 2 to 30.

The questions in the collection are of different lengths varying from 2 to 20 words, in various structures and belonging to diverse categories e.g., Computers and Internet, Health, Sports, Diet and Fitness, Pets, Yahoo! Products, Travel, Entertainment and Music, Education and Reference, Business and Finance, etc.



**Fig. 3.** Distribution of questions' length in the English collection



**Fig. 4.** Distribution of questions' length in the Arabic collection

Annotators were hired to annotate the questions with *relevant* if a candidate question is considered semantically equivalent to the original query or *irrelevant* otherwise. In case of conflict, a third annotator will make judgement for the final decision. As there is no Arabic Quora dataset available for the question retrieval task, for our experiments in Arabic we used the same English collection translated using Google Translation, the most widely used machine translation tool. The Arabic collection contains the same number of questions as the English set. To train word2vec for Arabic, we used a large-scale data set from cQA sites, namely the Yahoo!Webscope dataset<sup>9</sup>, translated into Arabic including 1,256,173 questions with 1 2,512,034 distinct words. Note that the parameters of word2vec were fixed using a parallel dev set of 217 queries and 1317 annotated questions. Tables 1 and 2 give examples of queries and their corresponding related questions from the test sets in English and Arabic respectively.

**Table 1.** An example of community questions from the English test set.

<b>Query:</b>	I often feel restless, uneasy, loss memory, lack of concentration, lose my temper. Why?
<b>Category:</b>	Health care
<b>Topic:</b>	Memory loss
<b>Related questions</b>	- I get short memory loss what should I do? - What to do when you are restless? - How can I improve my concentration and my memory or any mental exercise? - What is the best way to sharpen my brain?

To train Siamese LSTM, we used the publicly available Quora Question Pairs dataset<sup>10</sup>. The collection contains 400,000 samples question duplicate pairs where each sample has a pair of questions along with ground truth about their

<sup>9</sup> The Yahoo! Webscope dataset Yahoo answers comprehensive questions and answers version 1.0.2, available at “[http://research.yahoo.com/Academic\\_Relations](http://research.yahoo.com/Academic_Relations)”

<sup>10</sup> [www.kaggle.com/quora/question-pairs-dataset](http://www.kaggle.com/quora/question-pairs-dataset).

**Table 2.** An example of community questions from the Arabic test set.

<b>Query:</b>	كيف أقوم بتدريب كلب بلدي بالغ من العمر سنة واحدة؟
<b>Category:</b>	Pets
<b>Topic:</b>	Puppy training
<b>Related questions</b>	- ما هي أفضل طريقة لتدريب جرو جديد؟ - هل لدى أحدهم اقتراح حول كيفية إيواء جرو عمره ١٠ أسابيع؟ - كيف تدرّب كلبًا يبلغ من العمر عامًا؟ - كيفية تدريب كلب صغير جدًا؟

similarity (1: similar, 0: dissimilar). A set of 40,000 pairs was used for validation. Our test set was organized as pairs of questions to be directly fed into MaLSTM. Note that data preprocessing was done using Python NLTK.

## 4.2 Word Embedding Learning

For English word embedding training, we resorted to the publicly available word2vec vectors<sup>11</sup>, with dimensionality of 300, that were trained on 100 billion words from Google News. As there is no Arabic version of Google News vectors, we train on the Yahoo!Webscope dataset using the CBOW model, since it has proven through experiments to be more efficient and performs better than Skip-gram with sizeable data. The training parameters of the Arabic CBOW model were set after several tests:

- Size=300: feature vector dimension. We tested different values in the range [50, 500] but did not get significantly different precision values. The best precision was achieved with size=300.
- Sample=1e-4: down sampling ratio for the words that are redundant a lot in the corpus.
- Negative samples=25: number of noise words
- min-count=1: minimum number of words which we set to 1 to make sure we do not throw away anything.
- Context window=5: fixed window size. We tested different window sizes using the dev set. The best accuracy was obtained with window equals 5.

## 4.3 LSTM Training

For LSTM training, we applied the Adadelta method [23] 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 [15]. Our LSTM layers' size is 50 and embedding layer's size is 300. We used the back propagation and small batches of size equals 64, to reduce the cross-entropy loss

<sup>11</sup> <https://code.google.com/p/word2vec/>

and we resorted to the Mean Square Error (MSE) as a common regression loss function for prediction. We trained our model for several epochs to observe how the results varied with the epochs. We found out that the accuracy changed with changing the number of epochs but stabilized after epoch 25. Note that the given parameters were set based on empirical tests; each parameter was tuned separately on a development set to pick out the best one. For developing our model we used Keras<sup>12</sup> and Scikit-learn<sup>13</sup>. Note that we used the same LSTM configuration for both languages.

#### 4.4 Evaluation Metrics

We used the Mean Average Precision (MAP), Precision@n (P@n) and Recall as they are the most widely used metrics for evaluating the performance of question retrieval in cQA. 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 returns the proportion of the top-n retrieved questions that are relevant. Recall is the proportion of relevant similar questions that have been retrieved over the total number of relevant questions. Accuracy was also used which returns the proportion of correctly classified questions as relevant or irrelevant.

#### 4.5 Main Results and Discussion

To evaluate the performance of LSTMQR, we compare against our previous approach called WEKOS as well as some competitive state-of-the-art question retrieval methods tested by Zhang et al. in [25] on the same dataset. The methods being compared are summarized below:

- **WEKOS** [14]: A word embedding based method which transforms words in each question into continuous vectors. The questions are clustered using Kmeans and the similarity between them was measured using the cosine similarity based on their weighted continuous valued vectors.
- **TLM** [21]: A translation based language model which combines a translation-based language model with a query likelihood approach for the language model for the question and the answer parts respectively. TLM integrates word-to-word translation probabilities learned by using different sources of information.
- **ETLM** [19]: An entity based translation language model, which is an extension of TLM where the main difference is the replacement of the word translation with entity translation in order to integrate semantic information within the entities.
- **PBTM** [26]: A phrase based translation model which uses machine translation probabilities assuming that question retrieval should be performed at

<sup>12</sup> <https://keras.io/>

<sup>13</sup> <https://scikit-learn.org>

the phrase level. PTLM learns the probability of translating a sequence of words in a historical question into another word sequence of words in a given query.

- **WKM** [29]: 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 queries and the previous questions. A concept thesaurus was built based on the semantic relations extracted from Wikipedia.
- **M-NET** [28]: A continuous word embedding based model, which incorporates the category information of the questions to get a category based word embedding, assuming that the representations of words belonging to the same category should be semantically equivalent.
- **ParaKCM** [25]: A key concept paraphrasing based approach which explores the translations of pivot languages and expands queries with the paraphrases. It assumes that paraphrases offer additional semantic connection between the key concepts in the queried question and those of the historical ones.

The results in Table 3, show that PBTM outperforms TLM which demonstrates that detecting contextual information in modeling the translation of entire phrases or consecutive word sequences is more effective than translating separate words, as there is a dependency between adjacent words in a phrase.

**Table 3.** Question retrieval performance comparison of different models in English.

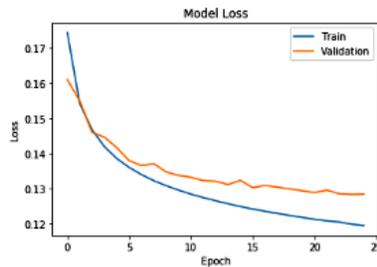
	<b>TLM</b>	<b>ETLM</b>	<b>PBTM</b>	<b>WKM</b>	<b>M-NET</b>	<b>ParaKCM</b>	<b>WEKOS</b>	<b>LSTMQR</b>
P@5	0.3238	0.3314	0.3318	0.3413	0.3686	0.3722	0.4338	<b>0.5023</b>
P@10	0.2548	0.2603	0.2603	0.2715	0.2848	0.2889	0.3647	<b>0.4188</b>
MAP	0.3957	0.4073	0.4095	0.4116	0.4507	0.4578	0.5036	<b>0.5739</b>

ETLM performs as good as PBTM, which proves that replacing the word translation by entity translation for ranking improves the performance of the translation language model. The performance of WKM is limited by the low coverage of the concepts of Wikipedia on the diverse users’ questions. ParaKCM achieves good precision by exploring the translations of pivot languages and expanding queries with the generated paraphrases for question retrieval. M-NET, based on word embeddings performs well owing to the use of metadata of category information to capture the properties of words. WEKOS also based on word embedding, TF-IDF weighting and kmeans achieves comparable results and further proves that the use of word embeddings get benefits from dense representation and reduce the negative impact of word mismatch by detecting semantic relations between words, while the other methods mostly do not capture enough information about semantic equivalence.

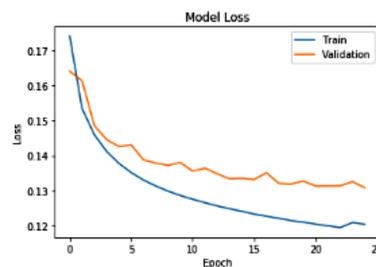
Our proposed approach LSTMQR outperforms in English all the compared methods on all criteria by returning a good number of relevant questions among

the retrieved ones. This good performance indicates that the use of Siamese LSTM along with query expansion and Manhattan similarity is effective in the question retrieval task. Word embeddings help to obtain an efficient input representation for LSTM, capturing syntactic and semantic information in a word level. Interestingly, our approach does not require an extensive feature generation due to the use of a pre-trained model. The results show that our Siamese based approach performs better than translation and knowledge based methods, which demonstrates that the question representations produced by the Siamese LSTM sub-networks can learn the semantic relatedness between pairs of questions and then are more suitable for representing questions in the question retrieval task. The Siamese network was trained using backpropagation-through-time under the MSE loss function which compels the LSTM sub-networks to capture textual semantic difference during training. A main virtue of LSTM is that it can accept variable length sequences and map them into fixed length vector representations which can resolve the length and structures problems in cQA.

In order to properly evaluate the MaLSTM model performance on our question similarity prediction problem, we plot training data vs validation data loss using the Matplotlib library. The loss is often computed on training and validation to diagnose its behavior and to check whether it is a good fit for the data or could perform better with a different configuration. It is a sum of the errors made for each instance in training or validation sets. From Figures 5 and 6, we can see that for both English and Arabic there is no considerable difference between the training and validation loss. The training loss keeps decreasing after every epoch which means that the model is learning to recognize the specific patterns. Similarly, the validation loss continues to decrease reaching 0.132 and 0.129 for English and Arabic respectively, thus our model is generalizing well on the validation sets. We can say that we have a good fit since both the train and validation loss decreased and leveled off around the same points.



**Fig. 5.** Epochs vs loss of MaLSTM on the English dataset



**Fig. 6.** Epochs vs loss of MaLSTM on the Arabic dataset

We used the simple Manhattan similarity function which forces the LSTM to entirely capture the semantic differences during training. In practice, our results

are fairly stable across different similarity functions namely cosine and Euclidean distances. We found out that Manhattan distance outperforms them as depicted in Tables 4 and 5 which proves that it is the most appropriate measure for the case of high dimensional text data.

**Table 4.** Comparison between similarity measures on the English dataset

	<b>P@5</b>	<b>Recall</b>
Manhattan	<b>0.5023</b>	<b>0.5385</b>
Cosine	<b>0.3883</b>	<b>0.4253</b>
Euclidean	<b>0.3383</b>	<b>0.3751</b>

**Table 5.** Comparison between similarity measures on the Arabic dataset

	<b>P@5</b>	<b>Recall</b>
Manhattan	<b>0.3692</b>	<b>0.4136</b>
Cosine	<b>0.2552</b>	<b>0.2997</b>
Euclidean	<b>0.2052</b>	<b>0.2496</b>

The cosine distance outperforms the Euclidean distance which demonstrates that it is better at catching the semantic of the questions, considering that the direction of the text points can be thought as its meaning, texts with similar meanings will have similar cosine score. Another reason is that Cosine distance is computed using the dot product and magnitude of each vector. So, it is only affected by the words the two vectors have in common, whereas the Euclidean measure has a term for every dimension which is non-zero in either vector. We can therefore say that the Cosine distance has meaningful semantics for ranking texts, based on mutual term frequency, whereas Euclidean distance does not.

Furthermore, we observed that LSTMQR 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*. In addition, LSTMQR was able to retrieve similar questions containing certain common misspelled terms like *recieve* instead of *receive* while it failed to detect 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 LSTMQR fails to detect semantic equivalence. Such cases include queries having only one similar question and most words of this latter do not appear in a similar context with those of the queried question, such as: *Which is better to aim my putter towards, the pole or the hole?* and *How do I aim for the target in golf?*.

**Table 6.** Question retrieval performance of LSTMQR in Arabic

	<b>WEKOS</b>	<b>LSTMQR</b>
P@5	0.3444	<b>0.3692</b>
P@10	0.2412	<b>0.2854</b>
MAP	0.4144	<b>0.4513</b>
Recall	0.3828	<b>0.4136</b>

Table 6 shows that our approach outperforms in Arabic the best compared system which gives evidence that it can perform well with complex languages.

However, our method ignores the morphological structure of Arabic words. Indeed, the Arabic language is an inflectional and a morphologically rich language with high character variation that has a significant impact on how influential a dataset is for producing good word embeddings. Therefore, exploiting the word internal structure is critical to detecting semantically similar words. For instance, the most similar words to «فعل» are all variants of the same word such as «نفعل، فعلنا، يفعلون، سنفعل، فاعل». Accordingly, endowing word embeddings with grammatical information (such as the person, gender, number, tense) could help to obtain more meaningful embeddings that capture morphological and semantic similarity. In terms of recall, LSTMQR reaches 0.4136 for Arabic which implies that the number of omitted similar questions is not big. We fine-tuned the parameter  $Nsw$  for the English and the Arabic corpora. We remarked that the query expansion has improved the results in terms of accuracy but also increases the execution time with the increase of the question size. With  $Nsw=5$ , the accuracy reaches 0.5377 and 0.3927 for English and Arabic respectively and then continues to slightly hover over this value but does not much increase. Thus, we set  $Nsw$  at 5 as an estimated value to avoid increasing the runtime. Interestingly, unlike traditional RNNs, Siamese LSTM can effectively handle the long questions and learn long range dependencies owing to its use of memory cell units that can store information across long input sequences. However, for very long sequences, LSTM may still fail to compress all information into its representation. Therefore, we envisage adding an attention mechanism to let the model attend to all past outputs and give different words different attention while modeling questions.

#### 4.6 Conclusion

Work on cQA has been mostly carried out for English, resulting in a lack of resources available for other languages, mainly Arabic. Motivated by this aspect, we tackled in this paper the task of question retrieval which is of great importance in real-world community question answering, in both English and Arabic. For this purpose, we proposed to use word embeddings to expand the questions and MaLSTM to capture the semantic similarity between them. Experiments conducted on large scale Yahoo! Answers datasets show that our approach can greatly improve the question matching task in English and Arabic and outperform some competitive methods tested on the same dataset. Interestingly, we showed that MaLSTM is capable of modeling complex semantics and covering the context information of question pairs. The word embedding based query expansion helped to enrich the questions and improve the performance of the approach. In the future, we look forward to improving the siamese architecture by adding an attention layer to calculate a weight for each word annotation according to its importance which offers a more meaningful representation of the

question. We also intend to enhance the Arabic word embedding by incorporating morphological features to the embedding model.

## References

1. Barrón-Cedeno, A., Da San Martino, G., Romeo, S., Moschitti, A.: Selecting sentences versus selecting tree constituents for automatic question ranking. In: Proceedings of COLING, the 26th International Conference on Computational Linguistics. pp. 2515–2525 (2016)
2. Cai, L., Zhou, G., Liu, K., Zhao, J.: Learning the latent topics for question retrieval in community qa. In: Proceedings of 5th International Joint Conference on Natural Language Processing. pp. 273–281 (2011)
3. 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: Proceedings of the 19th international conference on WWW. pp. 201–210. ACM (2010)
4. Cao, X., Cong, G., Cui, B., Jensen, C.S., Zhang, C.: The use of categorization information in language models for question retrieval. In: Proceedings of the 18th ACM conference on Information and knowledge management. pp. 265–274. ACM (2009)
5. Dos Santos, C., Barbosa, L., Bogdanova, D., Zadrozny, B.: Learning hybrid representations to retrieve semantically equivalent questions. In: Proceedings of ACL and the 7th International Joint Conference on NLP. vol. 2, pp. 694–699 (2015)
6. Duan, H., Cao, Y., Lin, C.Y., Yu, Y.: Searching questions by identifying question topic and question focus. In: ACL. vol. 8, pp. 156–164 (2008)
7. Hochreiter, S.: The vanishing gradient problem during learning recurrent neural nets and problem solutions. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems* **6**(02), 107–116 (1998)
8. Hochreiter, S., Schmidhuber, J.: Long short-term memory. *Neural computation* **9**(8), 1735–1780 (1997)
9. 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 (iSAI-NLP). pp. 1–7. IEEE (2019)
10. 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: Proceedings of SemEval. pp. 866–871 (2016)
11. Mikolov, T., Sutskever, I., Chen, K., Corrado, G.S., Dean, J.: Distributed representations of words and phrases and their compositionality. In: Advances in neural information processing systems. pp. 3111–3119 (2013)
12. 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: Proceedings of SemEval. pp. 828–835 (2016)
13. Mueller, J., Thyagarajan, A.: Siamese recurrent architectures for learning sentence similarity. In: Thirtieth AAAI Conference on Artificial Intelligence (2016)
14. Othman, N., Faiz, R., Smaili, K.: Enhancing question retrieval in community question answering using word embeddings. In: proceedings of the 23rd International Conference on Knowledge-Based and Intelligent Information Engineering Systems (KES) (2019)

15. Pascanu, R., Mikolov, T., Bengio, Y.: On the difficulty of training recurrent neural networks. In: International conference on machine learning. pp. 1310–1318 (2013)
16. Qiu, X., Tian, L., Huang, X.: Latent semantic tensor indexing for community-based question answering. In: ACL (2). pp. 434–439 (2013)
17. 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: Proceedings of COLING. pp. 1734–1745 (2016)
18. 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)
19. Singh, A.: Entity based q&a retrieval. In: Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning. pp. 1266–1277. ACL (2012)
20. Wang, K., Ming, Z., Chua, T.S.: A syntactic tree matching approach to finding similar questions in community-based qa services. In: Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval. pp. 187–194. ACM (2009)
21. Xue, X., Jeon, J., Croft, W.B.: Retrieval models for question and answer archives. In: Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval. pp. 475–482. ACM (2008)
22. 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)
23. Zeiler, M.D.: Adadelta: an adaptive learning rate method. arXiv preprint arXiv:1212.5701 (2012)
24. 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 Conference on Information and Knowledge Management. pp. 371–380. ACM (2014)
25. 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)
26. Zhou, G., Cai, L., Zhao, J., Liu, K.: Phrase-based translation model for question retrieval in community question answer archives. In: Proceedings of the 49th Annual Meeting of the ACL: Human Language Technologies-Volume 1. pp. 653–662. ACL (2011)
27. Zhou, G., Chen, Y., Zeng, D., Zhao, J.: Towards faster and better retrieval models for question search. In: Proceedings of the 22nd ACM international conference on Conference on information & knowledge management. pp. 2139–2148. ACM (2013)
28. 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)
29. 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)