Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 159 (2019) 485–494

Procedia Computer Science

www.elsevier.com/locate/procedia

23rd International Conference on Knowledge-Based and Intelligent Information & Engineering Systems

Enhancing Question Retrieval in Community Question Answering Using Word Embeddings

Nouha Othman[a*], Rim Faiz[b], Kamel Smaïli[c]

[a]*LARODEC, ISG Tunis, University of Tunis, Bardo, Tunisia*
[b]*LARODEC, IHEC Carthage, University of Carthage, Carthage Presidency, Tunisia*
[c]*LORIA, Campus Scientifique, University of Lorraine, Villers-lès-Nancy, F-54600, France*

Abstract

Community Question Answering (CQA) services have evolved into a popular way of online information seeking, where users can interact and exchange knowledge in the form of questions and answers. In this paper, we study the problem of finding historical questions that are semantically equivalent to the queried ones, assuming that the answers to the similar questions should also answer the new ones. The major challenge of question retrieval is the word mismatch problem between questions, as users can formulate the same question using different wording. Most existing methods measure the similarity between questions based on the bag-of-words (BOWs) representation capturing no semantics between words. Therefore, this study proposes to use word embeddings, which can capture semantic and syntactic information from contexts, to vectorize the questions. The questions are clustered using Kmeans to speed up the search and ranking tasks. The similarity between the questions is measured using cosine similarity based on their weighted continuous valued vectors. We run our experiments on real world data set from Yahoo! Answers in English and Arabic to show the efficiency and generality of our proposed method.

Keywords: Community Question Answering; Question retrieval; Word embeddings

1. Introduction

The goal of Question Answering (QA) systems is to automatically return succinct answers to questions posed in natural language. In recent years, Community Question Answering (cQA) has gained wide popularity as a viable method for seeking information online and leveraging user-generated content, such as Yahoo! Answers[1], Stackoverflow [2], and Quora [3]. CQA not only satisfies the question askers but also offers valuable references to other users with similar questions. However, such community services have accumulated large archives of question-answer pairs that are exponentially increasing including numerous duplicated questions. Therefore, users cannot easily find the answers they need and consequently post new questions that already exist in the archives. To make full use of the

* Nouha Othman. Tel.: +21650702011
E-mail address: othmannouha@gmail.com

[1] http://answers.yahoo.com/
[2] http://stackoverflow.com/
[3] https://fr.quora.com/

1877-0509
10.1016/j.procs.2019.09.203

huge archives of question-answer pairs and reduce the time lag required to get a new answer, it is critical to detect the historical questions that are semantically equivalent to the queried ones. If a similar question is found, its associated answer can be returned as a relevant response to the new posted query. The task of retrieving equivalent questions to the new queries from the cQA archives, known as the question retrieval task, has recently been subject to a burgeoning interest [18, 4, 3, 16, 20, 13, 19]. The major challenge is the lexical gap between the queried questions and the existing ones in the archives [18], which constitutes a barricade to traditional Information Retrieval (IR) models since users can formulate the same question employing different wording. For example, the questions: *How can I slow down signs of aging naturally?* and *What are some home remedies to keep your skin looking younger?* have the same meaning but different words and then may regarded as dissimilar. In order to bridge the lexical gap problem in cQA, most previous attempts focus on improving the similarity measure between questions while it is tricky to set a compelling similarity function for sparse and discrete representations of words. Recently, research efforts in distributional semantic representations, have led to the rise of word embeddings [9], which have been proved to be a valuable asset for various Natural Language Processing (NLP) tasks [12, 10, 24]. Word embeddings are neural network-based models which aim at mapping words from a vocabulary into real valued vectors in a low-dimensional space, where close vectors are supposed to indicate high semantic similarity between the corresponding words. Motivated by the tremendous success of these emerging models [1], in this paper, we propose a word embedding-based method to retrieve clustered questions. We tested the proposed method on a large-scale real data from Yahoo! Answers.

The remainder of this paper is organized as follows: Section (2) reviews the related work on question retrieval in cQA. Then, we describe in Section (3) our proposed word embedding based-method for question retrieval. Section (4) presents our experimental evaluation and Section (5) concludes the paper and outlines some ideas for future research.

2. Related Work

Recently, along with the flourishing of cQA archives, much attention has been paid to the question retrieval task. Here, we overview some of the recent approaches proposed to address this task.

Several works were based on the vector space model referred to as VSM to calculate the cosine similarity between a query and archived questions [6, 4]. However, the main limitation of VSM is that it favors short questions, while cQA services can handle a wide variety of questions not limited to factoïd questions. In order to overcome this shortcoming, BM25 has been used for question retrieval to take into consideration the question length [4]. Language Models (LM)s [5] 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. Nonetheless, such models might not be effective when there are few common words between the questions.

To handle the vocabulary mismatch problem faced by LMs, the translation model was employed to learn correlation between words based on parallel corpora and it has obtained significant performance for question retrieval. The basic 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 such as in [18, 3]. Within this context, Zhou et al. [22] tried to improve 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. [16] was extended the word-based translation model by incorporating semantic information and explored strategies to learn the translation probabilities between words and concepts using the cQA archives and an entity catalog. Although, the above-mentioned basic models have yielded good results, questions and answers are not parallel in practice, rather they are different from the information they contain [20].

Further approaches based on semantic similarity were required to bridge the lexical gap problem in question retrieval toward a deep understanding of short text to detect the equivalent questions. For instance, there were few attempts that have exploited the available category information for question retrieval like in [5, 4, 23]. Despite the fact that these attempts have proven to significantly improve the performance of the language model for question retrieval, the use of category information was restricted to the language model. Wang et al [17] used a parser to build syntactic trees of questions, and rank them based on the similarity between their syntactic trees and that of the query question. Nevertheless, such an approach requires a lot of training data. As observed by [17], existing parsers are still not well-trained to parse informally written questions. Latent Semantic Indexing (LSI) was also employed to address the given task like in [14]. While being effective to address the synonymy and polysemy by mapping words about the same concept next to each other, the efficiency of LSI highly depends on the data structure and both its training and inference are computationally expensive on large vocabularies.

Otherwise, 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 promise in several NLP tasks, in particular for question retrieval [24]. The main advantage of this unsupervised model is that it doesn't require expensive annotation; it only needs a huge amount of raw textual data for training. As we believe that the representation of words is crucial for retrieving similar questions and inspired by the success of the latter model, we rely on word embeddings to address the question retrieval task in cQA in both English and Arabic.

It is worth mentioning that most of the work on cQA has been carried out for other languages than Arabic. The most successful approach in Arabic [11] used text similarities at both sentence and word level on the basis of word embeddings. The similarities were calculated between new and past question, and between the new question and the answer related to the community question p. A tree-kernel-based classifier was used in [2] where the authors used supervised and unsupervised models that operated both at sentence and chunk levels for parse tree based representations. Malhas et al.[8] adopted a supervised learning approach where learning-to-rank models were trained over word2vec features and covariance word embedding features generated from the training data. Recently, Romeo et al. [15] addressed the same task with machine learning techniques using advanced Arabic text representations made by applying tree kernels to constituency parse trees along with textual similarities, and word embeddings.

3. Description of WEKOS

The basic intuition behind our proposed method for question retrieval, called WEKOS (Word Embedding, Kmeans and COSine based method), is to transform words in each question in the community collection into continuous vectors. Unlike traditional methods which represent questions as Bag Of Words (BOWs), we propose to represent each question as a Bag of-Embedded-Words (BoEW) in a continuous space. The continuous word representations are learned in advance using the continuous bag-of-words (CBOW) model [9]. The word embeddings of a question are weighted and averaged to get an overall representation of the question. Kmeans was used to create clusters from the collection of related questions. It provides a good strategy to decrease the data dimension and reduce the runtime cost for the ranking and search tasks. Each query is therefore compared to the questions contained within its closest cluster instead of the entire collection of the questions. Besides, the cosine similarity is used to calculate the similarity between the average of the word vectors corresponding to the queried question and that of each existing question in the given cluster. The historical questions are then ranked according to their cosine similarity scores in order to return the top ranking question as the most relevant one to the new query. As depicted in Figure 1, the WEKOS method consists of five steps namely, question preprocessing, word embedding learning, embedding vector weighting, question clustering and question ranking.

3.1. Question Preprocessing

Pre-processing aims at assessing and improving the quality of text data in order to ensure the validity and reliability of the statistical analysis. The question preprocessing module intends to process the natural language questions and extract the useful terms in order to represent them in a formal way. These latter are obtained by applying text cleaning, tokenization, stopwords removal and stemming. We also remove punctuation marks, diacritics, non letters and special characters such as #, $ and &. Letters are lowercased while numerical digits are normalized to the token 'Num'. Thus, at the end of the question preprocessing module, we obtain a set of filtered queries, each of which is formally defined as follows: $q = \{t_1, t_2, ..., t_Q\}$ where t represents a separate term of the query q and Q denotes the number of query terms. In order to preprocess the Arabic collection, in addition to the aforementionned tasks, orthographic normalization was required to reduce noise and ambiguity in the Arabic text data. This task includes Tatweel removal (deleting stretching symbol), Tachkil removal (ignoring arabic short vowels) and Letter normalization (variant forms to one form conversion). In fact, different spelling variants, such as the Hamza, are sometimes inconsistently misued by arabic writers; some users may ignore the Hamzas or employ a different Hamza variant. For this reason, we have chosen to normalize to one standard variant as follows: « أ ، إ ، آ ، ؤ ، ء ، ئ » are normalized to « ا ».

3.2. Word Embedding Learning

Word embeddings also known as distributed representations of words refer to a set of machine learning algorithms to build continuous word vectors based on their contexts in a large corpus using shallow neural network. They learn

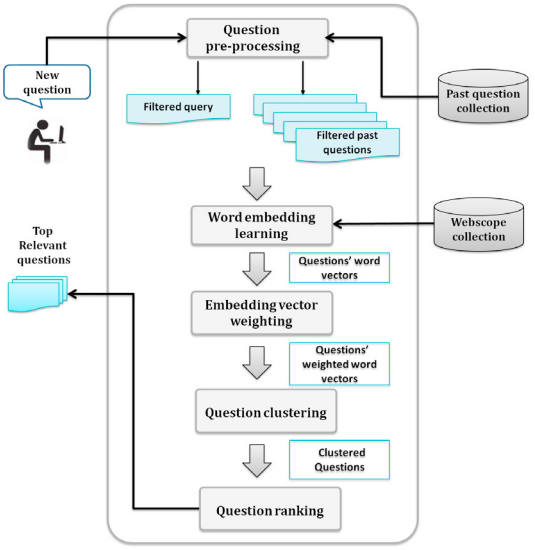

Fig. 1. WEKOS pipeline for question retrieval in cQA

a low-dimensional vector for each vocabulary term in which the similarity between the word vectors can capture the syntactic and semantic similarities between the corresponding words. Basically, there exist two main types of word embeddings namely Continuous Bag-of-Words model (CBoW) and Skip-gram model. The former one consists in predicting a current word given its context, while the second does the inverse predicting the contextual words given a target word in a sliding window. It is worth to note that, in this work, we consider the CBOW model [9] to learn word embeddings, since it has proven through our experiments to be more efficient and performs better with sizeable data than Skip-gram.

As shown in Figure 2, the CBOW model predicts the center word given the vector representation of its surrounding words using continuous distributed bag-of-words representation of the context, hence the name CBOW. It aims to find the probability of a word occurring in a context. For example, for the context *guy*, *attempt*, *over*, *puddle*, *fall*, CBOW is able to predict from these words, the center word *jump*.

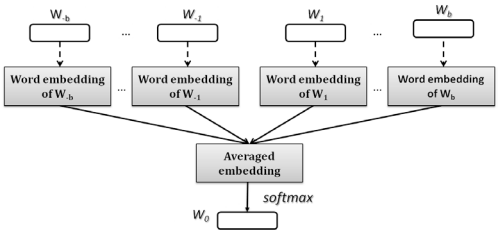

Fig. 2. Overview of the Continuous Bag-of-Words model.

The context vector is got by averaging the embeddings of each contextual word while the prediction of the center word w_0 is obtained by applying a softmax over the vocabulary V. Formally, let d be the word embedding dimension, the output matrix $O \in \Re^{|V| \times d}$ maps the context vector c into a $|V|$-dimensional vector representing the center word, and maximizes the following probability: $p(v_0 \mid w_{[-b,b]-\{0\}}) = \frac{\exp v_0^T O_c}{\sum_{v \in V} \exp v^T O_c}$ where b is a parameter defining the window of context words, O_c represents the projection of the context vector c into the vocabulary V and v is a one-hot representation. The strength of CBOW is that it does not rise substantially when we increase the window b.

3.3. Embedding Vector Weighting

Once the questions are represented as Bag of-Embedded-Words (BoEW), the continuous vectors are weighted using TF-IDF, which is one of the most widely used term weighting schemes in information retrieval systems owing to its simplicity and effectiveness. In other words, each embedding word is multiplied by the TF-IDF of the word it represents. TF-IDF is a statistic weighting function to estimate the importance of a word based on its relative frequency in a specific document and the inverse proportion of documents containing the word over the entire document collection. The TF-IDF weighting allows to have a suitable sentence representation for question comparison. As we work on questions, we adapt the basic function to our context by simply replacing documents with questions. Given a question collection C, a word w and a question q, TF-IDF is defined as follows: $tfidf(w, q, C) = tf(w, q) * idf(w, Q) = f_{w,q} * \log(\frac{|C|}{df_{w,C}})$

where $f_{w,q}$ is the number of times w appears in a question q, $|C|$ is the size of the question collection and $df_{w,C}$ is the total number of questions that contain the word w. We use TF-IDF to estimate how important is a word not only in a particular question, but rather in the whole collection of questions. Actually, some common words may occur several times in questions but they are not relevant as key-concepts to be indexed or searched. Intuitively, words that are common in a single or small set of questions will be assigned higher scores while words which appear frequently in questions tend to have low scores. The weighted embedding vectors of the query words are averaged to obtain the average vector V_q of the queried question as follows: $V_q = \frac{\sum_{i=1}^{|V|}(\upsilon_{w_i} \times tfidf(w_i,q,C))}{\sum_{i=1}^{|V|} tfidf(w_i,q,C)}$ where υ_{w_i} is the embedding vector of the word w_i generated by word2vec and $|V|$ is the number of word vectors in a given question q . Similarly, for each historical question, we compute its average vector V_d.

3.4. Question Clustering

To achieve better performance, we suggest to group similar questions together looking for the centers of each question cluster. In our work, we opt for the Kmeans [7] clustering algorithm owing to its simplicity and effectiveness. K-means partitions N items into K clusters according to the nearest mean calculation. The distance calculation in K-means refers to the Euclidean distance. The one single parameter we need to set is K. Note that text clustering was performed at sentence level where the averaged word embeddings of the questions are feeded to kmeans.

3.5. Question Ranking

The similarity between a queried question and a historical one in the vector space is calculated as the cosine similarity between V_q and V_d. Each query is compared to the questions contained within its closest kmeans cluster instead of the entire question collection contained in the community archive. Questions are ranked using cosine similarity scores based on their weighted vectors in order to return the top ranking questions having the maximum score, as the most relevant ones to the new query.

4. Experiments

4.1. Dataset

Our experiments were conducted using the dataset released by [21] for evaluation. To construct the dataset, the authors crawled questions from all categories in Yahoo! Answers, the most popular cQA platform, and then randomly splitted the questions into two sets while maintaining their distributions in all categories. The first set contains 1,123,034 questions as a question repository for question search, while the second is used as the test set and contains 252 queries and 1624 manually labeled relevant questions. The number of relevant questions related to each original query varies from 2 to 30. The questions are of different lengths varying from 2 to 15 words, in different structures and belonging to various categories e.g. Computers and Internet, Yahoo! Products, Entertainment and Music, Education and Reference, Business and Finance, Pets, Health, Sports, Travel, Diet and Fitness. Tables 1 and 2 show examples of queries and their corresponding related questions from the test sets in English and Arabic respectively.

Annotators were asked to label each query with “relevant” if a candidate question is considered semantically similar to the query or “irrelevant” otherwise. In case of conflict, a third annotator will make judgement for the final result. Note that the questions in the test data do not overlap with those in the retrieval data. To train the word embeddings,

Table 1. *Example of questions from the English test set.*

Query	Category	Topic	Related questions
How can I get skinnier without getting in a diet?	Diet and Fitness	Weight loss	- How do I get fit without changing my diet? - How can i get slim but neither diet nor exercise? - How do you get skinny fast without diet pills? - I need a solution for getting fit (loosing weight) and I must say I cant take tough diets ?

Table 2. *Example of questions from the Arabic test set.*

Query	Category	Topic	Related questions
لدي مشكلة فقدان الذاكرة كيف يمكنني حل المشكلة؟	Health and Disease	Memory loss	-افضل طريقة لتحسين التركيز وتقوية الذاكرة؟ -هل من حل لمشكلة النسيان و قلة الانتباه؟ -اي شخص اي علاجات صحية لفقدان الذاكرة؟ مرحبا لدي نقص التركيز ماذا يجب عمله؟

we resorted to another large-scale data set from cQA sites, namely the Yahoo! Webscope dataset[4], including 1,256,173 questions with 2,512,345 distinct words. 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 free online machine translation tool. The Arabic collection contains the same number of questions as the English set. The Arabic translated Yahoo! Webscope dataset, includes 1 2,512,034 distinct words, a bit fewer than the English set mostly due to internal voweling which can denotes passive constructors. For example, the phrase *He was dismissed* can be translated to one single Arabic word «فصل». Data preprocessing was performed before the experiments using NLTK [5]. After the preprocessing, the English corpus has been reduced by almost 15% and the Arabic one by nearly 20%. Note that the parameters of word2vec as well as kmeans were fixed using a parallel dev set of 217 queries and 1317 annotated questions.

4.2. Evaluation Metrics

In order to evaluate the effectiveness of our proposed method, we used Mean Average Precision (MAP) and Precision@n (P@n) as they are extensively used for evaluating the performance of question retrieval for cQA. Particularly, MAP is the most commonly used metric in the literature assuming that the user is interested in finding many relevant questions for each query. MAP 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. In order to fix the window size, we used the Accuracy which returns the proportion of correctly classified questions as relevant or irrelevant. Recall was also determined for our approach which is the proportion of relevant similar questions that have been retrieved over the total number of relevant questions.

4.3. Word Embedding Learning and Clustering

The word embeddings were trained on the whole Yahoo! Webscope dataset using word2vec in order to represent the words of the training data as continuous vectors which capture the contexts of the words. The training parameters of word2vec were set after several tests:

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

[5] https://www.nltk.org/

- Size=300: feature vector dimension. We tested different values in the range [50, 400] but did not get significantly different precision values. The best precision was achieved wih size=300.
- Sample=1e-4: this is the down sampling ratio for the words that are redundant a lot in corpus.
- Negative samples =25: the 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=10: fixed window size. Considering that the window size is a relevant parameter for improving the accuracy of the retrieval method, we tested different window sizes and report the accuracy values obtained when querying the word embeddings generated with Skip-gram and CBOW models. Figure 3 shows that with our English corpus, CBOW outperforms Skip-grams in terms of accuracy and for both models, the optimal window value is 10. In fact, larger window sizes increase the runtime needed to train word embedding models, so picking out the optimal window size can reduce the computing time.

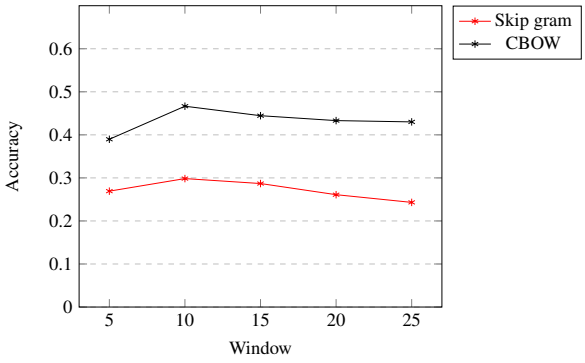

Fig. 3. Accuracy variations according to window size for CBOW and Skip-gram models with fixed dimensional vector model (size=300)

The number of clusters K is a crucial parameter to our method, used for clustering the vocabulary of words. These clusters are used subsequently to obtain the question representations. It is worth to mention that fixing k is challenging as we need to make a trade-off between cluster quality, memory and runtime to generate the clusters. After performing initial experiments with this parameter by varying it in the range of 25 to 200, we noted that the best results were obtained with K set to 100, hence we reported our results with this particular setting of the K parameter. Note that we run experiments on the English and Arabic corpora with the same word2vec and Kmeans parameters values.

4.4. Main Results

We compare the performance of WEKOS with the following competitive state-of-the-art question retrieval models tested by Zhang et al. in [21] on the same dataset:

- **TLM** [18]: A translation based language model which combines a translation-based language model with a query likelihood approach the language model for he question and the answer parts respectively. TLM integrates word-to-word translation probabilities learned by employing several sources of information.
- **ETLM** [16]: 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** [22]: A phrase based translation model which uses machine translation probabilities assuming that question retrieval should be performed at 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** [25]: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** [24]: 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** [21]: 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.

From Table 3, we can see that PBTM outperforms TLM which demonstrates that capturing contextual information in modeling the translation of phrases as a whole or consecutive sequence of words is more effective than translating single words in isolation. This is because, by and large, there is a dependency between adjacent words in a phrase.

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

	TLM	**ETLM**	**PBTM**	**WKM**	**M-NET**	**ParaKCM**	**WEKOS**	**WEKOS without TF-IDF**
P@5	0.3238	0.3314	0.3318	0.3413	0.3686	0.3722	**0.4338**	**0.3431**
P@10	0.2548	0.2603	0.2603	0.2715	0.2848	0.2889	**0.3647**	**0.2736**
MAP	0.3957	0.4073	0.4095	0.4116	0.4507	0.4578	**0.5036**	**0.4125**

The fact that ETLM (an extension of TLM) performs as good as PBTM proves that replacing the word translation by entity translation for ranking improves the performance of the translation language model. Although, ETLM and WKM are both based on external knowledge resource e.g. Wikipedia, WKM uses wider information from the knowledge source. Specifically, WKM builds a Wikipedia thesaurus, which derives the concept relationships (e.g. synonymy, hypernymy, polysemy and associative relations) based on the structural knowledge in Wikipedia. The different relations in the thesaurus are treated according to their importance to expand the query and then enhance the traditional similarity measure for question retrieval. Nevertheless, the performance of WKM and ETLM are limited by the low coverage of the concepts of Wikipedia on the various users' questions. M-NET, also based on continuous word embeddings performs well owing to the use of metadata of category information to encode the properties of words, from which similar words can be grouped according to their categories. The best compared system is ParaKCM, a key concept paraphrasing based approach which explores the translations of pivot languages and expands queries with the generated paraphrases for question retrieval.

The results show that our method WEKOS outperforms in English all the aforementioned methods on all criteria by returning a good number of relevant questions among the retrieved ones early. A possible reason behind this is that context-vector representations learned by word2vec can effectively address the word mismatch problem by capturing semantic relations between words, while the other methods do not capture enough information about semantic equivalence. We can say that questions represented by bag-of-embedded words can be captured more accurately than traditional bag-of-words models which cannot capture neither semantics nor positions in text. This good performance indicates that the use of word embeddings along with TF-IDF weighting and cosine similarity is effective in the question retrieval task. However, we find that sometimes, our method fails to retrieve similar questions: Out of 252 test questions, only 12 questions get P@10 values equal to zero. Most of these questions contain misspelled query terms. For instance, questions containing *sofwar* by mistake cannot be retrieved for a query containing the term *software*. Such cases show that our approach fails to address some lexical disagreement problems. Furthermore, there are few cases where WEKOS fails to detect semantic equivalence. Some of these cases include questions having one single 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?*. Obviously, further experiments with the dimensions of the embeddings are needed to improve the results.

Moreover, we tested our method with and without TF-IDF weighting (In Table 3, WEKOS and WEKOS without tfidf respectively) to examine its effect on question retrieval results. Through our experiments, we found that the use of TF-IDF allows to increase the P@5, P@10 and the MAP values. The reason behind this is that TF-IDF can detect questions that make frequent use of specific words and determine if they are relevant in the question. We can say that the discriminatory power of TF-IDF enables the retrieval engine to find relevant questions that could likely be similar to the new query. However, there are some cases when a word can be relatively common over the whole collection but still holds some importance throughout the question like the words *date* and *system*. Such common words get a low TF-IDF score, and thus are pretty much ignored in the search. Furthermore, TF-IDF doesn't take into account synonymy relations between terms. For example, if a user posted a question including the word *dwelling*, TF-IDF would not consider questions that might be similar to the query but instead use the word *bungalow*. TF-IDF can not resolve lexical ambiguity which is frequent in our community collection of informal and heterogeneous questions where the

same concept may be expressed in different ways. It is also worth mentioning that the computation complexity of TF-IDF is $O(nm)$, where n is the total number of words and m is the total number of questions in the corpus. For large collections like yours, this could present an escalating problem.

As it is expected, our method performed slightly worse in Arabic than in English as shown in Table 4.

Table 4. *Question retrieval performance of WEKOS in Arabic*

	WEKOS	**WEKOS without TF-IDF**
P@5	**0.3444**	**0.2545**
P@10	**0.2412**	**0.1933**
MAP	**0.4144**	**0.2916**

We consider that the major reason for that is that our word embedding based method ignores the morphological structure of Arabic words. In fact, the nature of the Arabic language as an inflectional and a morphologically rich language with high character variation has a significant impact on how influential a dataset is for producing good word embeddings. Accordingly, exploiting the word internal structure is crucial to detect semantically similar words. For example, the most similar words to the word «فعل» are all variants of the same word such as «فاعل ، سنفعل ، يفعلون ، فعلنا ، نفعل» . Therefore, enriching word embeddings with their main grammatical information (such as the word, gender, person, number, case, tense) could help to deliver more meaningful embeddings that capture morphological, context and semantic similarity. In terms of recall, we get 0.4677 and 0.3828 values for English and Arabic respectively, which imply that the number of omitted similar questions is not big.

We fine-tuned the parameter k within 25 to 200 for the English corpus. As shown in Figure 4, the more the k value increases, the more the clustering execution time increases, the more the search time decreases. It is worth observing

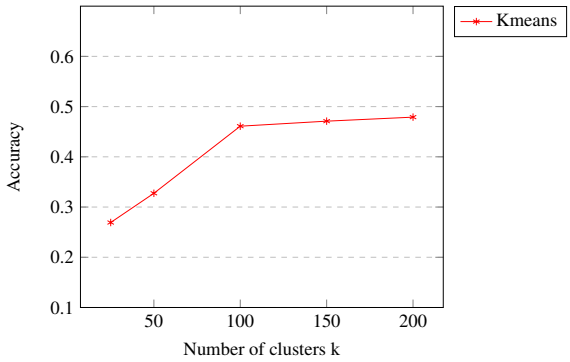

Fig. 4. Accuracy variations according to the number of clusters k

that the accuracy reaches 0.4877 with k=100 and then continues to slightly hover over this value but does not much increase. Thus, we set k to 100 as an estimated value to avoid increasing the clustering runtime. Further experiments are required to determine the actual best k value that satisfies the trade-off between cluster quality and runtime. Although being simple, linear and fast, the main drawback of kmeans is its non-deterministic nature since it requires to pre-specify the number of clusters and it randomly selects the initial centroids. Hence, hierarchical clustering could be a good alternative clustering approach since it does not require to pre-specify the number of clusters the way that k-means does.

5. Conclusion

In this paper, we address the problem of question retrieval which is of great importance in real-world community question answering tasks. In order to solve the word mismatch between questions a word embedding based method is proposed. Concretely, the question words are embedded in a continuous space and treated as a bag of embedded words. The produced word embeddings are learned using the CBOW model and weighted based on the frequency

of the words. The cosine similarity was used to calculate the similarity between the questions based on their vector-based word representations in a continuous space. K-means was performed on word embeddings to decrease the data dimension and improve the performance of the method. Experiments conducted on large-scale cQA data in English and Arabic show the effectiveness of our method mainly in English in detecting similar questions even if they share few common words. We have shown evidence that the TF-IDF weighting, though simple, can improve the search efficiency and the quality of the retrieval results. Nevertheless, there is a limit to represent word as one vector without considering lexical ambiguity. In the future, it would be of interest to enhance the Arabic word embedding by incorporating morphological features to the embedding model. In addition, we look forward to running our experiments with more different languages and larger corpora.

References

[1] Abidi, K., Smaïli, K., 2018. An automatic learning of an algerian dialect lexicon by using multilingual word embeddings, in: 11th edition of the Language Resources and Evaluation Conference, LREC 2018.

[2] Barrón-Cedeno, A., Da San Martino, G., Romeo, S., Moschitti, A., 2016. Selecting sentences versus selecting tree constituents for automatic question ranking, in: Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers, pp. 2515–2525.

[3] Cai, L., Zhou, G., Liu, K., Zhao, J., 2011. Learning the latent topics for question retrieval in community qa., in: Proceedings of 5th International Joint Conference on Natural Language Processing, pp. 273–281.

[4] Cao, X., Cong, G., Cui, B., Jensen, C.S., 2010. A generalized framework of exploring category information for question retrieval in community question answer archives, in: Proceedings of the 19th international conference on World Wide Web, ACM. pp. 201–210.

[5] Cao, X., Cong, G., Cui, B., Jensen, C.S., Zhang, C., 2009. The use of categorization information in language models for question retrieval, in: Proceedings of the 18th ACM conference on Information and knowledge management, ACM. pp. 265–274.

[6] Duan, H., Cao, Y., Lin, C.Y., Yu, Y., 2008. Searching questions by identifying question topic and question focus., in: ACL, pp. 156–164.

[7] Hartigan, J.A., Wong, M.A., 1979. Algorithm as 136: A k-means clustering algorithm. Journal of the Royal Statistical Society. Series C (Applied Statistics) 28, 100–108.

[8] Malhas, R., Torki, M., Elsayed, T., 2016. Qu-ir at semeval 2016 task 3: Learning to rank on arabic community question answering forums with word embedding, in: Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016), pp. 866–871.

[9] Mikolov, T., Chen, K., Corrado, G., Dean, J., 2013a. Efficient estimation of word representations in vector space. arXiv preprint arXiv:1301.3781 .

[10] Mikolov, T., Sutskever, I., Chen, K., Corrado, G.S., Dean, J., 2013b. Distributed representations of words and phrases and their compositionality, in: Advances in neural information processing systems, pp. 3111–3119.

[11] Mohtarami, M., Belinkov, Y., Hsu, W.N., Zhang, Y., Lei, T., Bar, K., Cyphers, S., Glass, J., 2016. Sls at semeval-2016 task 3: Neural-based approaches for ranking in community question answering, in: Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016), pp. 828–835.

[12] Musto, C., Semeraro, G., de Gemmis, M., Lops, P., 2016. Learning word embeddings from wikipedia for content-based recommender systems, in: European Conference on Information Retrieval, Springer. pp. 729–734.

[13] Nakov, P., Hoogeveen, D., Màrquez, L., Moschitti, A., Mubarak, H., Baldwin, T., Verspoor, K., 2017. Semeval-2017 task 3: Community question answering, in: Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017), pp. 27–48.

[14] Qiu, X., Tian, L., Huang, X., 2013. Latent semantic tensor indexing for community-based question answering., in: ACL (2), pp. 434–439.

[15] Romeo, S., Da San Martino, G., Belinkov, Y., Barrón-Cedeño, A., Eldesouki, M., Darwish, K., Mubarak, H., Glass, J., Moschitti, A., 2017. Language processing and learning models for community question answering in arabic. Information Processing & Management .

[16] Singh, A., 2012. Entity based q&a retrieval, in: Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning, ACL. pp. 1266–1277.

[17] Wang, K., Ming, Z., Chua, T.S., 2009. 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, ACM. pp. 187–194.

[18] Xue, X., Jeon, J., Croft, W.B., 2008. Retrieval models for question and answer archives, in: Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval, ACM. pp. 475–482.

[19] Ye, B., Feng, G., Cui, A., Li, M., 2017. Learning question similarity with recurrent neural networks, in: 2017 IEEE International Conference on Big Knowledge (ICBK), IEEE. pp. 111–118.

[20] Zhang, K., Wu, W., Wu, H., Li, Z., Zhou, M., 2014. 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, ACM. pp. 371–380.

[21] Zhang, W.N., Ming, Z.Y., Zhang, Y., Liu, T., Chua, T.S., 2016. Capturing the semantics of key phrases using multiple languages for question retrieval. IEEE Transactions on Knowledge and Data Engineering 28, 888–900.

[22] Zhou, G., Cai, L., Zhao, J., Liu, K., 2011. 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, ACL. pp. 653–662.

[23] Zhou, G., Chen, Y., Zeng, D., Zhao, J., 2013a. Towards faster and better retrieval models for question search, in: Proceedings of the 22nd ACM international conference on Conference on information & knowledge management, ACM. pp. 2139–2148.

[24] Zhou, G., He, T., Zhao, J., Hu, P., 2015. 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.

[25] Zhou, G., Liu, Y., Liu, F., Zeng, D., Zhao, J., 2013b. Improving question retrieval in community question answering using world knowledge., in: IJCAI, pp. 2239–2245.