Expert Systems With Applications 137 (2019) 432–442

Contents lists available at ScienceDirect

Expert Systems With Applications

journal homepage: www.elsevier.com/locate/eswa

A multi-approach to community question answering

Yassine El Adlouni [a,*], Horacio Rodríguez [b], Mohammed Meknassi [a], Said Ouatik El Alaoui [a], Noureddine En-nahnahi [a]

[a] Laboratoire Informatique et Modélisation (LIM), Faculté des Sciences Dhar El Mahraz (FSDM), Université Sidi Mohammed Ben Abdellah (USMBA), Fez, Morocco
[b] UPC, Barcelona, Spain

ARTICLE INFO

Article history:
Received 12 May 2018
Revised 11 July 2019
Accepted 11 July 2019
Available online 12 July 2019

Keywords:
Community question answering
Information retrieval
Arabic natural language processing
Learning to rank
Deep learning

ABSTRACT

In this paper we face the problem of Community Question Answering for the Arabic language. In this setting, a member of the community posts an initial query expressed in Natural Language. Other participants post their own interventions: answers, comments, additional questions, etc. contributing to building a rather tangled thread of nodes containing Natural Language short texts. The task consists in answering the initial query using the thread as the space of possible answers. The task can be approached as a classification, a regression or a ranking problem. In our case we select the set of possible candidates, we assign a relevance score to each candidate and we rank them accordingly.

We propose a bunch of unsupervised models and show that a model based on Latent Semantic Indexing approach outperforms state of the art models for this task. We also use transfer learning to power the embeddings layer of various deep learning models and prove that the pairwise approaches outperform their pointwise counterparts. All the proposed models have been evaluated on Semeval 2017 Task 3 Subtask D: Arabic Community Question Answering and achieve state of the art or near performance.

1. Introduction

Community Question Answering (cQA) has attracted much attention in recent years due to the dramatic increase of Question Answering (QA) websites, Yahoo!Answers, Quora, Reddit and StackExchange for instance, where users ask questions answered by other users or experts. Besides the obvious purpose of getting the answer to a question, the purpose of cQA includes helping the user to check similar questions before submitting her request which could help save considerable time, avoid waiting for answers, reduce unanswered questions, minimize duplicate questions and, thus, contributing to the efficiency of the platform.

The growth of Natural Language Processing (*NLP*) applications in recent years has resulted in a parallel increase in the accuracy requirements for such *NLP* systems. To deal with these requirements, a semantic analysis of text has become practically mandatory. In this setting, the most important efforts have been made in the framework of *SemEval* challenges which started in 2007.[1] Until 2012 the challenges were organized every three years; since then the evaluations have become annual. Many semantic-related facets have been tackled in these years including representations of meaning (May, 2016), distributional, formal and linguistic semantics (Krebs, Lenci, & Paperno, 2018), lexical semantics (Camacho-Collados et al., 2018), lexical resources and ontologies (Jurgens & Pilehvar, 2016), semantic parsing, semantic role labeling (Ruppenhofer, Sporleder, Morante, Baker, & Palmer, 2010), opinion mining and sentiment analysis of tweets (Rosenthal, Farra, & Nakov, 2017), reading comprehension and reasoning (Ostermann, Roth, Modi, Thater, & Pinkal, 2018), different forms of QA, including cQA (Nakov et al., 2017), multilingual and cross-lingual issues (Camacho-Collados, Pilehvar, Collier, & Navigli, 2017) and others. Much research in recent years has focused on English cQA, previous work related to Arabic cQA has focused on using latent models as Weighted Textual Matrix Factorization (WTMF) (AlMarwani & Diab, 2017), Latent Semantic Indexing[2] (LSI) (Adlouni et al., 2017) and using similarity features (Torki, Hasanain, & Elsayed, 2017).

* Corresponding author.
E-mail addresses: yeladlouni@gmail.com (Y.E. Adlouni), horacio@cs.upc.edu (H. Rodríguez).

[1] Previous evaluations, named Senseval, started in 1998, were reduced to Word Sense Disambiguation task.
[2] Also named Latent Semantic Analysis (LSA).

https://doi.org/10.1016/j.eswa.2019.07.024
0957-4174/

The performance of Arabic cQA is highly influenced by the unbalanced nature of target datasets, the lengthy aspect of questions and answers, the technical topics discussed on these fora, the scarcity of linguistic resources characterizing Arabic and usage of Latin, transliterated and noisy words, these issues produce low values of mAP and accuracy compared to English cQA.

These approaches can be time-consuming as they require feature engineering, furthermore, they rely on pointwise approaches while pairwise and listwise approaches are more natural for ranking. The pointwise, the pairwise, and the listwise differ with regards to the loss function and are used for ranking where for the first two ones, the group structure is ignored transforming the ranking problem into classification, regression or ordinal classification while for the listwise approach, ranking lists are used to harness the group structure. The loss function is defined as follows for each approach:

- The pointwise loss function is defined on a single object mapping a feature vector $\vec{x}$ to a label y.
- The pairwise loss function is defined on a pair object mapping a pair of feature vectors $(\vec{x}_1, \vec{x}_2)$ to a label y.
- The listwise loss function is defined on a tuple object mapping a tuple feature vectors $(\vec{x}_1, \vec{x}_2 \ldots \vec{x}_n)$ to a label y.

Relatively little information exists regarding the effectiveness of applying pairwise approaches combined with deep learning methods for cQA. To help address these gaps, we seek to check the following hypotheses:

- A topic modeling approach could be effective for cQA.
- Reducing the features space could help reduce overfitting.
- Pretrained word embeddings could help improve the results.
- Using the pairwise approach for cQA is more natural than using the pointwise approach.

Our contributions are three-fold:

- The LSA based model was able to outperform the state of the art system for the targeted dataset following an unsupervised approach. Unlike the SemEval 2017 winning system, the LSA based model does not require external resources and therefore is more practical and more efficient for this task.
- The overfitting of neural networks models was partially reduced using pre-trained word embeddings trained on a curated list of medical domain articles, it was also alleviated by reducing the features space using the overlappings terms of the triplets instead of their concatenation.
- Using the pairwise approach for cQA helps improving the results for deep learning models.

The rest of this paper is organized as follows: Section 2 surveys the related works. In Section 3, we present the followed methodology. The used approaches are depicted in Section 4. The experimental setup is introduced in Section 5 and the obtained results are discussed in Section 6. Finally, the conclusions are presented in Section 7.

2. Related work

QA, i.e. querying a computer using *NL*, is an old objective of *NLP*. Though initially *QA* systems focused on factual questions (who, where, when, Y/N, etc.), increasingly, the scope of *QA* has become wider, facing complex questions, list questions, definitional, why questions, etc. In parallel, the *QA* systems have suffered a process of specialization: domain-restricted *QA* (Mollá & Vicedo, 2007), *QA* for comprehension reading (Wu et al., 2019), *QA* over Linked Data (Unger et al., 2015), or *cQA*, our objective in this article.

cQA differs from conventional *QA* systems basically on four aspects:

- The source of the possible answers, that are the threads of queries and answers activated from the original query. So, the document retrieval or passage retrieval components, needed in other *QA* systems can be avoided or highly simplified. Instead, the topology of the thread and the dialog-based relations occurring in it could be used as features for the task.
- The structure of the threads and the available metadata can be exploited for the task. For instance, when the participants are identifiable, from their profile, i.e. topics of interest, the frequency of participation, wording, etc, they can be classified as an expert or a lay and this information could be important for the task.
- The types of questions include the frequent use of complex questions, as definitional, why, consequences, how_to_proceed, etc.
- Topics are ordinarily technical, answers are usually larger than questions and may contain references or hyperlinks to other threads, questions paraphrasing may introduce differences in wordings by the expert while spam answers may introduce irrelevant and noisy information.

In the case of Arabic, we have to point out the known difficulties of processing this language, e.g. usually unvowelized texts, complex morphology, scarcity of resources and processors. Besides these generic difficulties for processing this language, Mohtarami et al. (2016) argument on the difficulty of the CQA task because of the origin (social fora), differences in lengths, differences in wording, e.g. professional vs lay wording, the inclusion of irrelevant information, as personal digressions, religious formulas, the frequent inclusion of foreign terms especially medications, and so forth. See also (Abouenour, 2014) that discusses in depth the QA task using Arabic language.

Most *cQA* systems use supervised (*ML*), techniques and differ basically regarding the features used by the classifiers (or regressors or rankers). Between the most used we find Support Vector Machines, *SVM*, (with different Kernels), as in Severyn and Moschitti (2016), Barrón-Cedeño et al. (2016), or Belinkov, Mohtarami, Cyphers, and Glass (2015), *Logistic Regression*. Uva, Bonadiman, and Moschitti (2018) and Romeo, Da San Martino, Barrón-Cedeño, and Moschitti (2018) use Tree Kernel methods. Different variants of Conditional Random Fields (*CRF*) (Joty, Màrquez, & Nakov, 2016), Random Forests (Malhas, Torki, & Elsayed, 2016), and several types of rankers, as L2Rank library, Malhas et al. (2016). These techniques have been used alone or in combination.

Some systems use nontextual features when available, as the metadata of the threads, temporal features, author profiles, the topology of the thread structure of messages, etc. An interesting system, based on the exclusive use of this kind of features is Jeon, Croft, Lee, and Park (2006).

Most of the systems use, as core features or combined with others, textual features, superficial (string-based), syntactic, and, less frequently, lexico-semantic (knowledge-based), usually reduced to similarity or relatedness measures between the textual components of the thread (query, query/answer pairs), Gomaa and Fahmy (2013) present an excellent survey of these classes of features. Gomaa and Fahmy (2014) present a survey of distance measures applied to Arabic. Some systems for which this kind of features are important are Kashyap et al. (2016), Belinkov et al. (2015), Mohtarami et al. (2016), Baldwin et al. (2016), and Wu and Lan (2016).

There is increasing use of features based on distributional semantics, dimension reduction, embedding, etc. The use of dimensionality reduction techniques (as *LSA*, or Latent Dirichlet Annotation, *LDA*) is frequent in these approaches. Embedding techniques are used both as features for conventional similarity mea-

sures and as the first layer in Neural Networks, *NN*, models. The most popular embeddings are *Word2Vec*[3] as in Wang and Poupart (2016), Malhas et al. (2016), Franco-Salvador, Kar, Solorio, and Rosso (2016), and *Glove*,[4] as in Wu and Lan (2016), or Wang and Poupart (2016) or (Belinkov et al., 2015). Frequently, at least for English, existing embeddings are used without further retraining, in other cases, always in the case of Arabic, new embeddings are obtained training the models with appropriate corpora as Belinkov et al. (2015) and Mihaylov and Nakov (2016).

Machine Reading approaches as in Wang, Yan, and Wu (2018) have obtained good results recently. Look at the proceedings of the ACL 2018 Workshop on Machine Reading for Question Answering.[5]

As an alternative (or a complement) to dimension reduction or embeddings some systems group together the words using clustering or topic modeling. Tran, Tran, Vu, Nguyen, and Pham (2015) use Topic Models (obtained using *LDA*) and combine them with embeddings. Mohtarami et al. (2016) combine a clustering approach, using *Brown clustering*, with Topic Modeling, using *non-negative matrix factorization*. Also, Wu and Lan (2016) extract Topic Models, in this case using *GibbsLDA++*. Mihaylov and Nakov (2016) also combine word clusters with *LDA* Topic models. Also conventional *IR* techniques have been used, like *Lucene*, used by Wu and Lan (2016)). In (Attardi, Carta, Errica, Madotto, & Pannitto, 2017) the system ThReeNN is presented. The proposed model exploits both syntactic and semantic information to build a single and meaningful embedding space creating sequences of inputs for a Recurrent Neural Network, which is then used for the ranking purposes of the Task. Some systems focus on the likely most significant parts of the query for performing the comparison with the answer candidates: Higurashi, Kobayashi, Masuyama, and Murao (2018) perform an Extractive Headline Generation, (Wu, SUN, & WANG, 2018) propose Question Condensing Networks (QCN) for answer selection. Query expansion techniques. Some systems expand the terms of the query using lexical resources. Magooda et al. (2016), for instance, use *Google synonyms* for expanding English queries and *Arabic WP* for Arabic. Also wordnets (both *WN* and *AWN*) have been used for expanding the query.

Systems based on shallow or deep *NN* have grown dramatically for this task. Practically all the *NN* families have been applied, in general with good results.

We can find systems based on simple Feed Forward *NN, FFNM*, Wang and Poupart (2016), Malhas et al. (2016), Convolutional *NN, CNN*, sometimes applied in parallel over the original query and each of the associated queries, *2D-cNN*, sometimes over the concatenation *1D-cNN*. Frequently convolution is followed by a *max-pooling* layer. Different kind of filters are applied in the *CNN* layer, Wu and Lan (2016), Mohtarami et al. (2016), Severyn and Moschitti (2016), Baldwin et al. (2016).

Recurrent *NN, RNN*, Mohtarami et al. (2016), and recursive *NN, RecNN* for representing the sentences, are very popular, too. The most used systems from this family are the Long Short Term Models, *LSTM*, Wu and Lan (2016). Tan, Xiang, and Zhou (2015) used neural attention over a bidirectional LSTM neural network in order to generate better answer representations given the questions.

Another example is the work of Kateryna Tymoshenko and Moschitti. (2016), who combined neural networks with syntactic kernels. In most systems, a final *SoftMax* layer is included for performing the classification or ranking step.

Mohtarami et al. (2016) use two *LSTM*, over the concatenation of the original query with each query (resp. answer) in the thread, followed by a *Multilayer Perceptron* and a *SoftMax* layer. Zhou, Hu, Chen, Tang, and Wang (2015) use a *2D-cNN* for representing question-answer pairs, followed by an *LSTM* for learning the answer sequence.

Although some systems use in-house implementations of the neural models (using standard python or matlab libraries), some higher-level libraries as *Theano*,[6] *Chainer*,[7] *TensorFlow*,[8] *PyTorch*,[9] *Penne*,[10] or *KERAS*[11] are used as well.

Comparison between the original query and the other interventions (especially queries and answers) in the thread is sometimes performed using paraphrase techniques, as Gleize and Grau (2013), alignments between the original query and the attached queries or answers (or their concatenations), as in Franco-Salvador et al. (2016), and sometimes Machine Translation measures, as *METEOR, BLEU, TER, NIST*, or *Asiya* (in systems as Vo, Magnolini, and Popescu (2015), Zhao, Zhu, and Lan (2014), Wu and Zhang (2016), Barrón-Cedeño et al. (2016), or Wu and Lan (2016)).

Some systems combine several of these Knowledge Sources. A notable system, obtaining the best results in 2015 challenge, is *QCRI* Nicosia et al. (2015), that combines lexical, syntactic (using tree kernels), and distributional semantics. Also, *ECNU* Yi, Wang, and Lan (2015) combines these types of features. Other systems using a combination of simple approaches are Baldwin et al. (2016), Barrón-Cedeño et al. (2016), Magooda et al. (2016), Lahbari, Rodríguez, and Alaoui (2018), or Belinkov et al. (2015).

Different approaches have been used for measuring the similarities from the original query. Let us denote as q the original query, $q.qa_i$ the i-th pair of question/answering triggered by q, and $q.qa.q_i$ and $q.qa.a_i$ the question and answer components of such pairs. With this notation, and using $\otimes$ for concatenation operator, many possibilities arise: sim(q, $q.qa.q_i$), sim(q, $q.qa.a_i$), sim(q, $q.qa.q_i \otimes q.qa.a_i$), etc. Malhas et al. (2016) present a nice comparison of these measures.

In the frequent case of representing questions and answers as vectors, the most popular distance measures are just the inner product or the cosine over the original vectors or their transformations or embeddings. Sometimes, however, algebraic transformations are performed for weighting the different dimensions. Mahalanobis matrices, learned or estimated, are used for such purpose. Malhas et al. (2016), for instance, use directly the covariance matrix of the input vectors.

By far, the most popular suites for linguistic processing are *Stanford Core*[12] for English and *Madamira*[13] for Arabic. Other choices include *DKPro*[14] and *Berkeley's parser*.[15]

To sum up, attention-based neural models prove extremely useful due to their capacity to detect similarities between the question and the answer, they are resilient to differences between lengths by focusing on the relevant part of the answer and are further improved by using embeddings trained over huge corpora which are able to cope with noisy and irrelevant content. Differences in wording are tackled using knowledge bases combined with graph-based models. Though classical approaches based on lexical similarity are less powerful due to differences in length and wordings,

[3] http://deeplearning4j.org/word2vec.html.
[4] http://nlp.stanford.edu/projects/glove/.
[5] https://www.aclweb.org/portal/content/acl-2018-workshop-machine-reading-question-answering.
[6] http://deeplearning.net/software/theano/.
[7] http://chainer.org/.
[8] https://www.tensorflow.org/.
[9] https://pytorch.org/.
[10] https://nlp.nd.edu/penne/.
[11] https://keras.io/.
[12] https://stanfordnlp.github.io/CoreNLP/.
[13] http://www1.cs.columbia.edu/~rambow/software-downloads/MADA_Distribution.html.
[14] https://dkpro.github.io/.
[15] https://github.com/slavpetrov/berkeleyparser.

Table 1
Tf-Idf parameters.

Parameter/Hyperparameter	Value
max_features	10,000
min_df	1
max_df	0.1
sublinear_tf	True
norm	L2

Table 2
Word2vec Parameters/Hyperparameters.

Parameter/Hyperparameter	Value
Embeddings size	128
Vocabulary size	10,000
Batch size	128
Number of steps	300,000
Skip window	1
Number of skips	2
Number of samples	64

they are usually backed by a syntactic analysis, they are also preceded by a summarization step to yield tantamount content.

3. Methodology

A community question answering task could be defined formally as a tuple of three elements $\mathcal{S} = \{(\mathcal{Q}, \mathcal{D}), y\}$. $\mathcal{Q}$ denotes a set of queries. $\mathcal{D}$ denotes the pairs of question answers. $y \in \mathcal{Y}$ is a label representing the level of relatedness between the query and the pair where $\mathcal{Y} = \{Direct, Relevant, Irrelevant\}$ is a graded set. Given a query $q_i \in \mathcal{Q}$, consisting of a query, and a set of question-answer pairs $\mathcal{D}_i \in \mathcal{D}$, the task of cQA is to re-rank the question-answer pairs according to the conditional probability $Pr(y_i|q_i, \mathcal{D}_i)$.

Fig. 4 illustrates the architecture of our supervised model where training, validation and test datasets are parsed from XML, preprocessed yielding tokenized and indexed terms for each original question, question or answer, the relations files contain for each dataset the list of relations between them. The output from the preprocessing step is combined with the embeddings then fed to different neural networks models as inputs to Keras or Scikit-Learn in order to predict the similarity score and relevance.

3.1. Preprocessing

To tokenize the Arabic text, we have used both the word punkt included in nltk and the Stanford CoreNLP splitter and tokenizer. The results from the latter, used henceforth, outperform those obtained using the former.

3.2. Feature engineering

During our study, we have used various features as inputs to our models depending on the model.

3.2.1. Bag of words

We have used Tf-Idf (scikit-learn implementation) as inputs for the topic modeling approaches based on LSA, LDA, and Negative Matrix Factorization denoted NMF, the tuned parameters are reported in 1 where :

- max_features: Specify the size of the vocabulary considering the most frequent used terms.
- min_tf: Used to ignore terms with a document frequency strictly lower than the given threshold.
- max_tf: Used to ignore terms with a document frequency strictly higher than the given threshold. As the value used is a float number (0.1), it is considered as a proportion and therefore multiplied by the document count.
- sublinear_th: We used $1 + logf_{t,d}$ rather than $f_{t,d}$.
- norm: We used the euclidean norm for terms vectors.

$$tfidf_{t,d} = (1 + logf_{t,d}) * (log\frac{N}{1+n_t})$$

Table 3
Models hyperparameters.

Parameter/Hyperparameter	PyramidNet	DotNet	BiGRU
Embeddings Size	128		
Vocabulary Size	89,150	69,414	10,000
Query Length	64	64	1000
Question Length	64	64	314
Batch size	100	100	128
Query per iteration	50	50	–
Batch per iteration	5	5	–
Epochs	100	100	20
Optimizer	Adam		
Learning Rate	0.01		
Number of hidden layers	6	1	1
Sizes of the hidden layers	[128, 64, 32, 16, 8, 1]	[32]	[8]
Margin	1	1	–
Dropout	–	–	0.5

Table 4
Summmary of notations.

Notation	Description
$\mathcal{Q}$	Query set
$\mathcal{D}$	Pairs set
$\mathcal{Y}$	Label set/grade set
$\mathcal{S} = \{(q_i, \mathcal{D}_i), y_i\}$	Dataset
$q_i \in \mathcal{Q}$	i-th query in a dataset
$\mathcal{D}_i = \{d_{i,1}, d_{i,2}, \ldots, d_{i,n_i}\}$	Pairs set of the query q_i
$x_i = \Phi(q_i, d_{i,j})$	Feature vector from $(q_i, d_{i,j})$
$\mathbf{\Pi_i}$	Set of possible rankings on $\mathcal{D}_i$ with regard to q_i
$\boldsymbol{\pi_i} \in \mathbf{\Pi_i}$	Permutation on $\mathcal{D}_i$ with regard to q_i
$\mathcal{C}_i$	Set of combinations of 2 pairs on $\mathcal{D}_i$
$\equiv$	Modulo symbol
$\succ$	Ranking symbol
$\frown$	Concatenation symbol

Table 5
SemEval 2017 Task 3 subtask D.

Dataset	Questions	Pairs
Training	1031	30,411
Validation	250	7384
Test	1400	12,581

3.2.2. Lexical features

We have used lexical features including Longest Common Subsequence, Jaro–Winkler, Levenshtein, Damerau–Levenshtein, Sorensen–Dice coefficient, N-Gram and Q-Gram to measure the similarity between a pair of text chunks. These features were used for the SVM^{Rank} model.

3.2.3. Distributional embeddings

For neural models, BiGRU, DotNet and PyramidNet, we have used distributional embeddings as they have a dense structure and encapsulate the semantics of words. We used word2vec as described in Mikolov, Chen, Corrado, and Dean (2013) on a scraped corpus of Arabic Medical articles containing 314,000 unique terms from various topics like diseases, body parts, and drugs. The parameters and hyperparameters for Word2vec as depicted in Table 2 are fine-tuned based on the validation set performance.

Fig. 2 visualizes a projection using Principle Components Analysis with three components of the obtained embeddings for the word Sodium/where similar minerals or drugs containing this mineral are adjacents.

Using the t-distributed stochastic neighbor embedding 'tSNE' as depicted in Fig. 3, we notice that similar words to Sodium/صوديوم are clustering in the same region.

4. Approaches

In our study, two approaches were explored:

4.1. Unsupervised approach

The unsupervised approach is based on topic modeling models: LSA, LDA, and NMF. As depicted in Algorithm 1 , to build our model, we first preprocess the query text by removing the stopwords and tokenizing the text using the ISRI light stemmer (Taghva, Elkhoury, & Coombs, 2005), finally, we apply a matrix decomposition using the SVD technique to the queries set, where the number of components is 900, this value is chosen based on the performance of the model on our validation set. The number of iterations for the SVD solver is 3, the decomposition is followed by a normalization step.

$$similarity(d, q) = cosine(x_{i,d}, x_{i,q})$$

where $x_{i,q} = \Phi(q_i)$ are the vector features obtained from the query q_i and $x_{i,d} = \Phi(d_{i,j})$ are those obtained from the question d_{ij}.

The cosine similarity is used as a score to rank our pairs, a threshold of 0.5 is chosen to deduce the relevance. The tunning of this threshold is chosen to maximize the performance on the validation set.

4.2. Supervised approaches

Unlike the pointwise method where each original question q_i is compared to a pair of question-answer denoted (d_j^q, d_j^a) in order to measure their similarity, the pairwise method is based on a set of pairs of questions belonging to the same original question thus taking into account the relationships that could exist among questions. We transform our dataset in a way that makes it easy to learn the rankings. The outputs for the query q_i are $\mathcal{D}^+$ and $\mathcal{D}^-$ where the former contains pairs of questions (d_m, d_n) and $y_m \succ y_n$ while the latter contains pair of questions (d_m, d_n) where $y_n \succ y_m$ and $\succ$ denote the ranking operator, the set of pairs denoted $\mathcal{C}_i$ are obtained by combining, for the same original question q_i, questions that have different labels $y_m \neq y_n$. We chose 1 as the positive label and 0 as the negative label. We interleave $\mathcal{D}^+$ and $\mathcal{D}^-$ in order to obtain positive labels in even positions and negative labels in odd positions.

Algorithm 1: LSA based algorithm.

```
components ← 900
iterations ← 3
M ← TFIDF(Q)
W ← SVD(M, components, iterations);
for i ← 1 to m do
    for j ← 1 to l do
        q ← q_i · W
        d_ij ← TFIDF(d_ij^0 ⌢ d_ij^1)
        d ← d_ij · W
        σ ← max(0, 1 − cosine(q, d))
        if σ > 0.5 then
            ρ ← true
        else
            ρ ← false
        end
    end
end
```

Therefore the transformed dataset is composed of:

feature vectors: $(x_i)^{(1)}$, $(x_i)^{(2)}$ where: $(x_i)^{(1)} = \Phi(d_0)$ and $(x_i)^{(2)} = \Phi(d_1)$

labels:

$$y_i = f((x_i)^{(1)}, (x_i)^{(2)})$$

The value of y_i is 1 if i is even and 0 otherwise.

For Φ, the features extraction function, we have tried 2 approaches, one based on a word tokenizer while the other is based on trigrams. We have used different deep learning architectures for training following both pointwise and pairwise methods. These models accept as input sequences of text and output a score or a label depending on the target measure (ranking or classification). We proceed in 3 steps:

- A training step where many batches of queries, question-answer pairs and its associated labels are presented to the model in order to minimize a loss function. We denote a transformed training dataset containing m queries by:

  $$S' = \{(\mathbf{x_i^1}, \mathbf{x_i^2}, \mathbf{y_i})\}_{i=1}^{m}$$

- A validation step where a validation set is used to tune the different hyperparameters in a way to get high values for our target metrics, the mAP and Accuracy.

  $$V' = \{(\mathbf{x_i^1}, \mathbf{x_i^2}, \mathbf{y_i})\}_{i=1}^{m}$$

- A test step where for a new query q_{m+1} the transformed test dataset is denoted by:

  $$T' = \{(\mathbf{x_{m+1}^1}, \mathbf{x_{m+1}^2})\}$$

  Where the label y_{m+1} should be predicted.

The architectures shown are deep models composed of several layers of different sizes combining various building blocks of neural networks such as Embedding, GRU and Fully Connected Layers denoted MLP. The intuition behind using these architectures is to detect semantic similarity by merging the embeddings layers directly which is equivalent to compare the left question and the right question at the word level as depicted in Fig. 7, merging a flat representation of the embeddings which is equivalent to compare the left question and right question at the sentence level as depicted in Fig. 6 or applying the embeddings to the overlapping to reduce the dimensionality of the network as depicted in Fig. 5.

- **SVM$^{\text{Rank}}$** is a pairwise method based on SVM for ranking. We used the lexical features described in 3.2.2 and a linear kernel.
- **BiGRU** denotes the architecture described in Fig. 5 where the input layer accepts as input either sequences taken either from the intersection or the concatenation of the original question terms and the union of the question-answer pair terms. These sequences are then embedded to get dense vectors using the previously described word2vec method. The next layer is a Bidirectional GRU (Chung, Gülçehre, Cho, & Bengio, 2014) containing 8 units in both directions with a RELU activation. The last layer is a fully connected network containing one unit and squashing the output through the sigmoid function. During training, we use the cross-entropy as a loss function

  $$\mathcal{L} = -(ylog(\hat{y} + (1-y)log(1-\hat{y})))$$

  where y is the true value and $\hat{y}$ is the predicted value, we also use Adam (Kingma & Ba, 2014) as the optimizer. Table 3 shows the parameters and hyperparameters used to train the network.

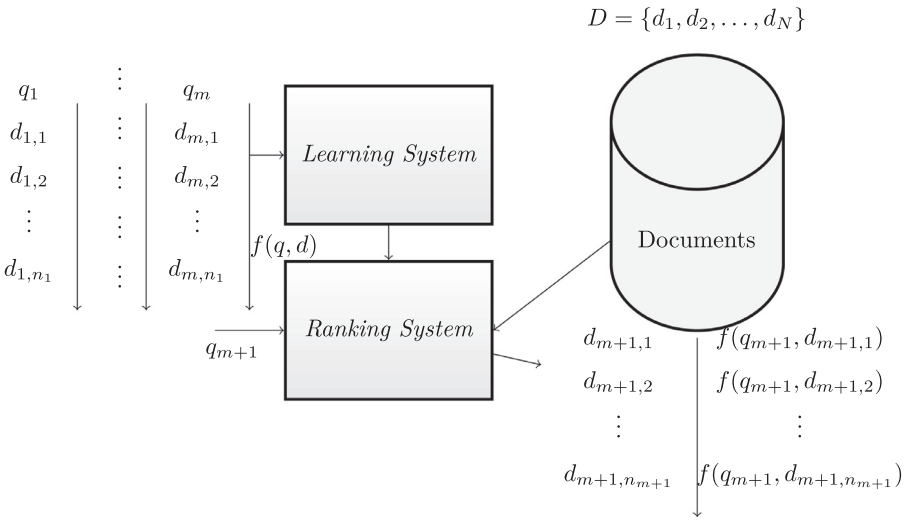

Fig. 1. Ranking workflow.

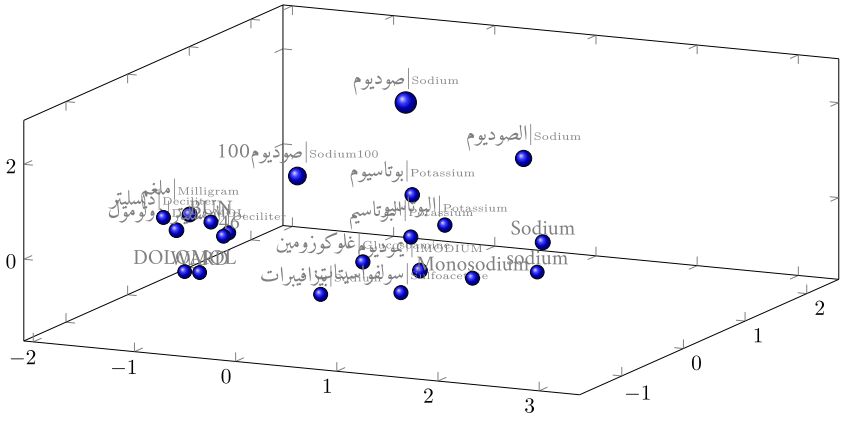

Fig. 2. PCA projection for Sodium Embeddings.

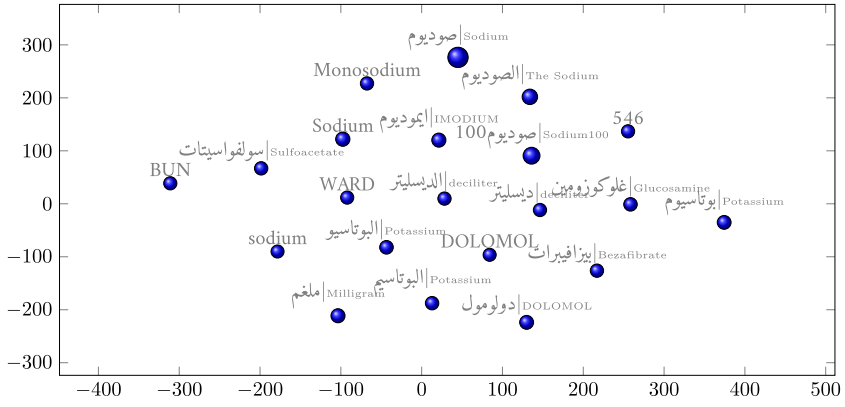

Fig. 3. tSNE projection for Sodium Embeddings.

- **DotNet** denotes the architecture described in Fig. 6 where the inputs are trigram sequences for each input question related to the same query as presented in the pairwise method paragraph. These sequences are then embedded where the embeddings are trained during the task. The next layer is a dense layer containing 10 units where the activation function is tanh and using a dropout of 0.2. The output layer is a scalar obtained using a normalized dot product of its inputs. For training we use adam as the optimizer and ranking hinge loss as a loss function, more formally defined as:

$$\mathcal{L} = max(0, m - \sigma(q, d^-) + \sigma(q, d^+))$$

where σ is the ranking function, $q \in \mathcal{Q}$, $d^- \in \mathcal{D}^-$ and $d^+ \in \mathcal{D}^+$ and m is called the margin.
Table 3 exposes the other parameters and hyperparameters used to train the network.
- **PyramidNet** denotes the architecture described in Fig. 7 where the inputs are sequences of words obtained

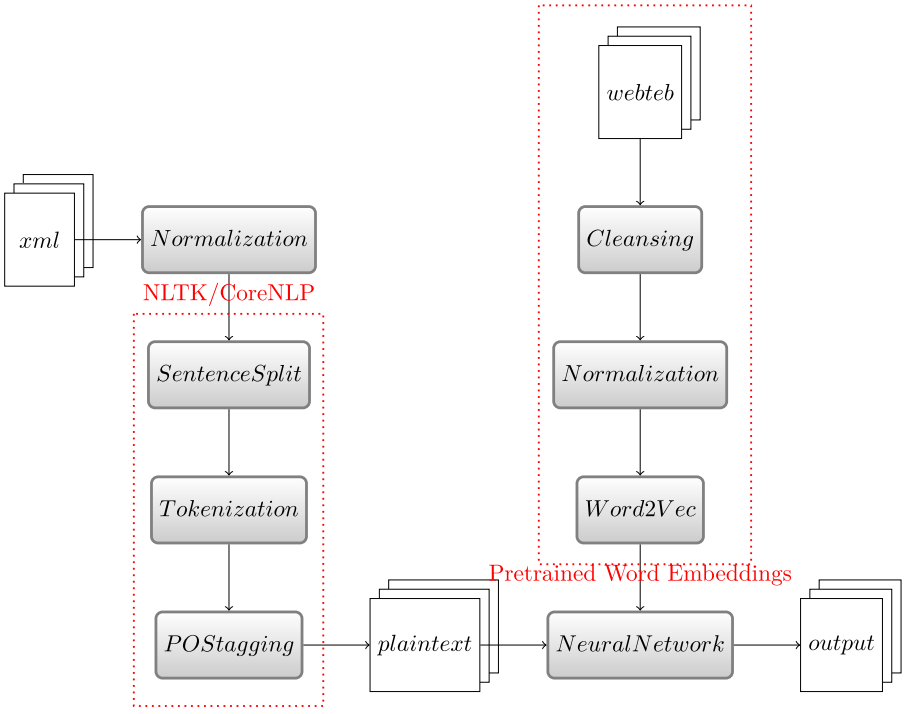

Fig. 4. System architecture.

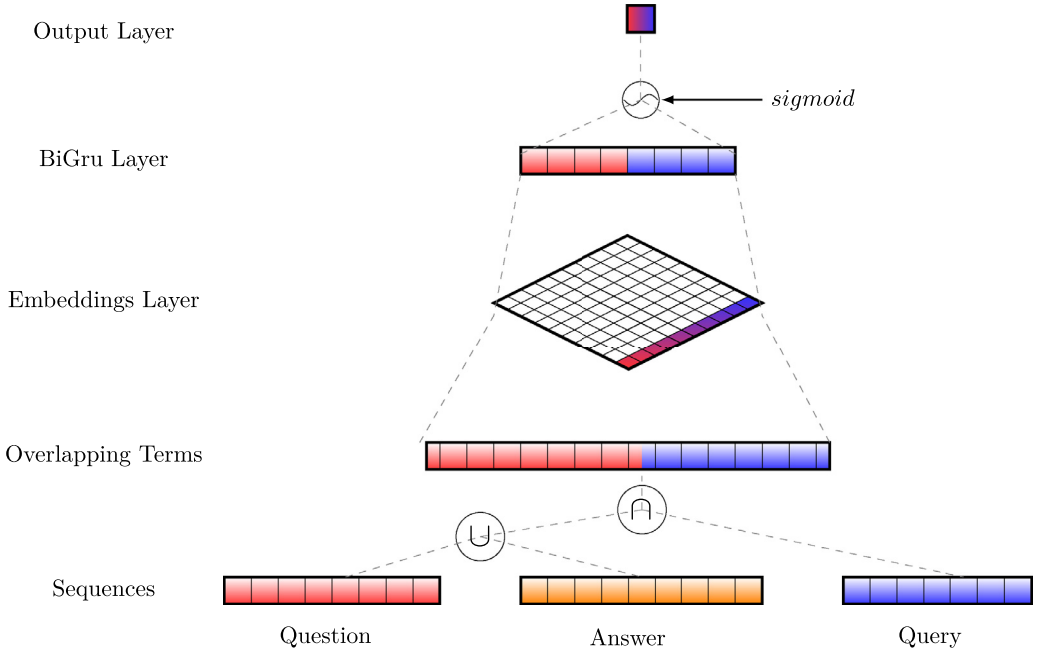

Fig. 5. BiGRU architecture.

from the inputs questions that are later embedded using word2vec pre-trained embeddings described previously. The next layer denoted hidden layer in Fig. 7 is composed of two legs, the first leg use element-wise multiplication of the inputs embeddings, later on, the k largest entries are processed through multiple dense layers with shrinking sizes as mentioned in Table 3. and softplus as the activation function. The second leg process the first question using a softmax function. Both legs flatten their outputs and element-wise multiplication is used to produce the rankings.

5. Experiments and results

In this section, we evaluate the proposed methods on Semeval 2017 Task 3 Subtask D benchmark datasets. This dataset was extracted from Arabic medical fora. The challenge provided a dataset containing three XML files for training, validation, and test, Table 6 is a flat representation of the XML datasets while Table 5 mention for each dataset the number of the original questions and the number of pairs. For training and validation datasets, each query tag (Question) identified by QID relate to up to 30 QA pairs (QApair) identified by QAID while for the test dataset each

Table 6
SemEval 2017 Task 3 Subtask D Training Excerpt.

Qid	Qaid	Question	Qaquestion	Qaanswer	qarel	qaconf
200,634	312,689	Who gets tooth decay من يصاب بتسوس الاسنان	Tooth decay in children تعفن الاسنان عند الاطفال	Children's teeth are exposed to tooth decay ... تتعرض اسنان الطفل لمرض تسوس الاسنان...	D	1.0
200,634	37,871	Who gets tooth decay من يصاب بتسوس الاسنان	I have a problem of caries in the two front عندي مشكلة التسوس في السنتين الاماميتين من... teeth ...	Must follow the scientific specificities by the يجب اتباع الخصوصيات العلميه من قبل طبيب الاسنان... dentist ...	R	0.6641

Table 7
Semeval 2017 Task 3 Subtask D Competition Results.

Model	MAP	Acc	P	R	F1
GU_QA	61.16	60.77	00.00	00.00	00.00
UPC_USMBA	57.73	66.24	63.41	33.00	0.4341
QU_BIGIR	56.69	49.64	41.59	70.16	52.22

Table 8
Test dataset results.

Model	MAP	Acc	P	R	F1
LDA	53.58	63.06	60.94	16.25	25.66
NMF	56.42	62.52	58.95	14.75	23.59
LSA+NLTK	61.31	62.16	83.08	04.47	08.49
LSA+CoreNLP	61.66	62.34	85.87	04.80	09.09
SVM^{rank}	56.22	61.08	100	00.81	06.10
BiGRU-intersection	56.93	59.07	48.58	73.58	58.52
BiGRU-concatenation	48.53	39.23	39.23	100	56.36

query relates to up to 10 QA pairs. The query text is contained in Qtext while the question and answer are contained in QAquestion and QAanswer respectively, they are extracted by parsing the XML files. A QA pair contains two attributes, QArel which denotes the relevance of the pair regarding the query on a graded scale (I if irrelevant, R if relevant and D if direct). QAconf refers to the relevance of the answer with respect to the question which is used to rank the pairs by decreasing relevance.

Given a new query and its related QA pairs, this task purpose is to predict these two attributes and therefore could be reduced to the following subtasks:

- Classification: Classify the QA pairs based on their similarity with the query.
- Ranking: Rank the QA pairs by decreasing similarity with regard to the query.

We used SemEval scorer to evaluate the obtained results where the official indicator is an extension of the widely used measure in IR which is the MAP (Mean Average Precision) as described in Liu, Moffat, Baldwin, and Zhang (2016).

For a specific query q_i, its related documents $\mathcal{D}_i$, its associated labels $\mathbf{y_i}$ and the related ranking list π_i, the average precision denoted AP is:

$$AP = \frac{\sum_{j=1}^{n_i} P(j).y_{i,j}}{\sum_{j=1}^{n_i} y_{i,j}}$$

where P(j) is the precision until the position of $d_{i,j}$:

$$P(j) = \frac{\sum_{k:\pi_i(k)\leq\pi_i(j)} y_{i,k}}{\pi_i(j)}$$

where $\pi_i(j)$ is the position of $d_{i,j}$ in π_i.

The MAP is thus the mean over all the queries contained in the test dataset. The other reported indicators are Accuracy, Precision, Recall and the F1 measure. (Fig. 1 and Table 4)

Table 7 reports the results obtained by the participating teams for SemEval 2017 Task 3 Subtask D. To the best of our knowledge, these works were the only ones published for the target dataset.

Table 8 shows the results obtained using the topic modeling approaches, SVM^{Rank}, and Bidirectional GRU. Table 9 is a comparison between the pairwise approach and the pointwise approach for the DotNet and PyramidNet models.

For each architecture, BiGRU, DotNet, and PyramidNet, we report the parameters and the hyperparameters used during the training phase in Table 3.

6. Discussion

The results show that the LSA based model using the algorithm and parameters described in Section 4.1 can lead to better results compared to the state of the art model for this task, using the Stanford CoreNLP tokenizer improves further the results. Though the insufficient evidence according to a Welch's t-test $t(2797.7) = -0.35$, $p < 0.363$ to support a true difference between our system (mAP = 0.6166) and the state of the art one (mAP = 0.6116) highlighted in Table 7, overall, according to a Welch's t-test $t(25159) = -2.56$, $p < 0.005$, our system (Accuracy = 0.6234) outperforms the state of the art one (Accuracy = 0.6077). Using other topic modeling approaches such as NMF and LDA, for instance, leads to comparable accuracies but low mAPs.

Table 9
Pairwise vs pointwise results.

Model	Pairwise					Pointwise				
	MAP	Acc	P	R	F1	MAP	Acc	P	R	F1
DotNet	53.71	56.68	43.88	37.34	40.35	52.44	56.04	42.93	36.53	39.47
PyramidNet	57.57	58.89	47.19	40.15	43.39	55.00	57.30	44.81	38.13	41.20

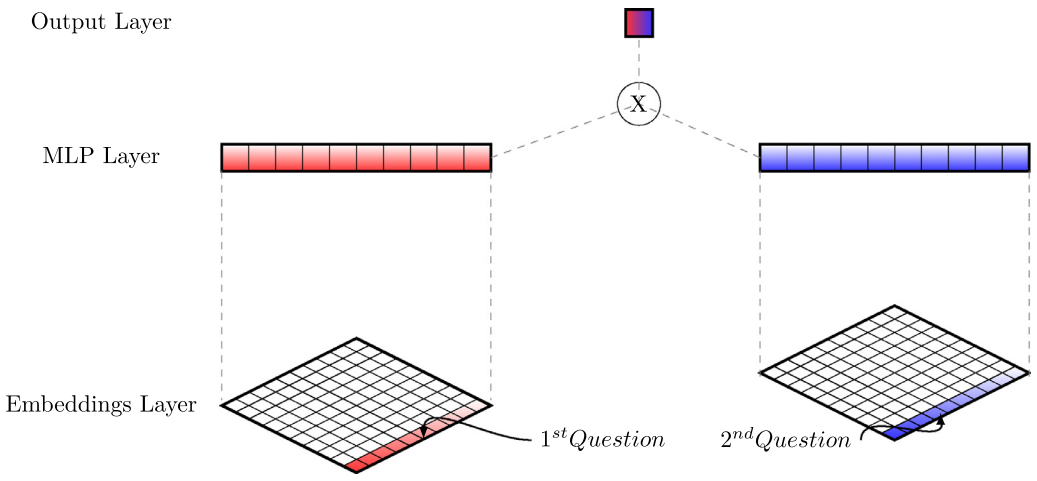

Fig. 6. DotNet architecture.

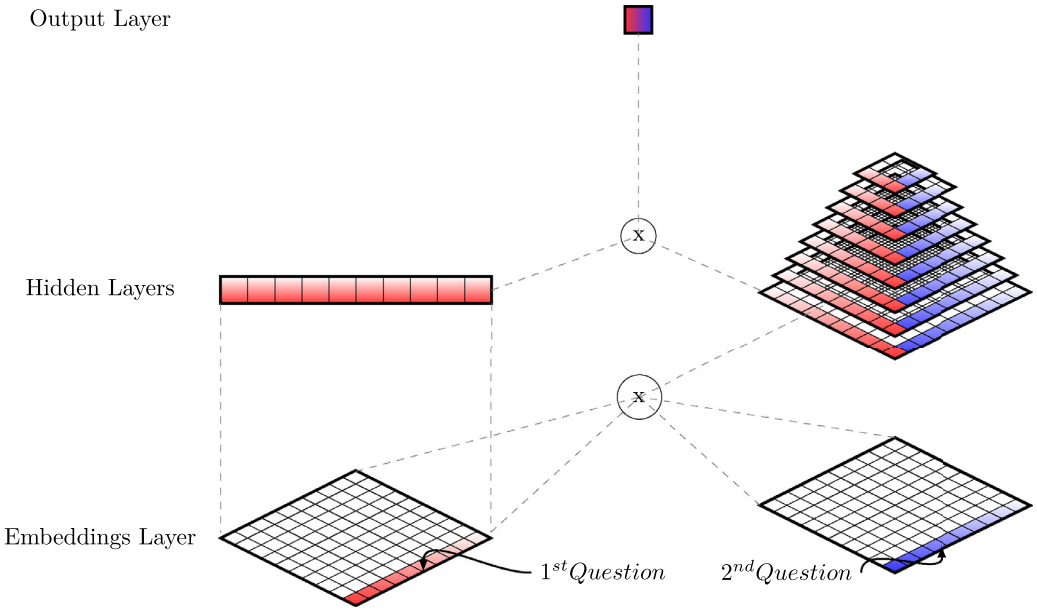

Fig. 7. PyramidNet architecture.

Using SVM^{Rank}, the ubiquitous method for ranking, produced a high precision but a low mAP, this could be explained by the unbalanced nature of the dataset and the length of text chunks.

The BiGRU model is usually used to encode sentences, the encoded sentences are then fed to a dense layer for classification, a comparison between the results obtained using the intersection of the triplets as input are considerably better than those obtained using the concatenation as portrayed in Table 8, this could be explained by the ratio of the dataset size and the features vector size, deep learning methods require a huge training corpus in order to learn model weights while avoiding overfitting. Minimizing features using triplets intersection reduce also the training time.

The comparison between the pairwise approach and the pointwise counterpart as depicted in Table 9 shows that the former outperforms the latter for all evaluation metrics which confirm our initial hypothesis. For the two models, we have used the same architecture and applied the same hyperparameters.

To sum up:

- The LSA based model was able to produce a good MAP following an unsupervised approach. Unlike the SemEval 2017 winning system, the LSA based model does not require external resources and therefore is more practical and more efficient for this task.
- The overfitting of neural networks models was partially reduced using pre-trained word embeddings trained on a curated list of medical domain articles, it was also alleviated by reducing the features space using the overlappings terms of the triplets instead of their concatenation.
- Using the pairwise approach for cQA helps to improve the results for deep learning models.

7. Conclusion

Community Question Answering systems are witnessing increasing popularity recently. Much of this interest stems from their capacity to represent a hub for communities to ask questions, seek information and share knowledge around various topics in a trustworthy and timely manner. Content in these online fora is characterized by a high amount of redundancy, presence of misspelled words, and usage of paraphrasing which leads to low accuracies.

Motivated by these aspects, in this paper we addressed the problem of cQA for an Arabic medical domain forum by harnessing

the connection between the original question and the thread question. Our main idea is to (i) adjust the LSA model to leverage the relationships between the triplet components (ii) use stemming to alleviate the impact of noise and misspelled words on the performance of the system (iii) use pairwise neural networks to improve the ranking performance.

Using the LSA model, we were able to obtain high values for the mAP, the accuracy, and the precision thus outperforming the winning system for this task. Using the pointwise approach for the BiGRU model, we were bound to use overlapping terms as features to get comparable performance to the pairwise approaches which confirm our initial hypothesis that pairwise methods perform better than pointwise methods.

In the future, we plan to (i) devote our effort to leveraging the syntactic structure of sentences by using kernel trees to compute the similarity measures between constituency and dependency trees (ii) combine machine translation evaluation metrics such as Meteor, BLEU, TER with lexical features which would be of great value to harness the lexical similarity (iii) experiment with graph-based models using a graph presentation for the question and the related pairs of question answers combined with knowledge bases as a huge proportion of terms in the medical domain are written in English or transliterated from English (iv) use augmentation as a technique to tackle overfitting and unbalanced datasets (v) use automatic summarization and topic detection to alleviate the problem of questions and answers length by extracting the most relevant part of the text (vi) use the tri-letter root of words in parallel with the conventional word-based embeddings as most words in Arabic derive from a tri-letter verb.

Declaration of competing interest

We wish to draw the attention of the Editor to the following facts which may be considered as potential conflicts of interest and to significant financial contributions to this work.

Credit authorship contribution statement

Yassine El Adlouni: Conceptualization, Funding acquisition, Formal analysis, Writing - original draft, Validation. **Horacio Rodríguez:** Conceptualization, Investigation, Validation, Writing - review & editing. **Mohammed Meknassi:** Validation, Writing - review & editing. **Said Ouatik El Alaoui:** Validation, Writing - review & editing. **Noureddine En-nahnahi:** Validation, Writing - review & editing.

Acknowledgements

All persons who have made substantial contributions to the work reported in the manuscript (e.g., technical help, writing and editing assistance, general support), but who do not meet the criteria for authorship, are named in the Acknowledgements and have given us their written permission to be named. If we have not included an Acknowledgements, then that indicates that we have not received substantial contributions from non-authors.

References

Abouenour, L. (2014). *Three-levels approach for arabic question answering systems* Ph.D. thesis.

Adlouni, Y. E., Lahbari, I., Rodríguez, H., Meknassi, M., Alaoui, S. O. E., & Ennahnahi, N. (2017). UPC-USMBA at semeval-2017 task 3: Combining multiple approaches for CQA for arabic. In S. Bethard, M. Carpuat, M. Apidianaki, S. M. Mohammad, D. M. Cer, & D. Jurgens (Eds.), *Proceedings of the 11th international workshop on semantic evaluation, semeval@acl 2017, Vancouver, Canada, august 3–4, 2017* (pp. 275–279). Association for Computational Linguistics. doi:10.18653/v1/S17-2044.

AlMarwani, N., & Diab, M. T. (2017). Gw_qa at semeval-2017 task 3: Question answer re-ranking on arabic fora. In S. Bethard, M. Carpuat, M. Apidianaki, S. M. Mohammad, D. M. Cer, & D. Jurgens (Eds.), *Proceedings of the 11th international workshop on semantic evaluation, semeval@acl 2017, Vancouver, Canada, august 3–4, 2017* (pp. 344–348). Association for Computational Linguistics. doi:10.18653/v1/S17-2056.

Attardi, G., Carta, A., Errica, F., Madotto, A., & Pannitto, L. (2017). Fa3l at semeval-2017 task 3: A three embeddings recurrent neural network for question answering. In *Proceedings of the 11th international workshop on semantic evaluation (semeval-2017)* (pp. 299–304). Association for Computational Linguistics. doi:10.18653/v1/S17-2048.

Baldwin, T., Liang, H., Salehi, B., Hoogeveen, D., Li, Y., & Duong, L. (2016). Unimelb at semeval-2016 task 3: Identifying similar questions by combining a CNN with string similarity measures. In *Proceedings of the 10th international workshop on semantic evaluation, semeval@naacl-hlt 2016, San Diego, CA, USA, june 16–17, 2016* (pp. 851–856). http://aclweb.org/anthology/S/S16/S16-1131.pdf.

Barrón-Cedeño, A., Martino, G. D. S., Joty, S. R., Moschitti, A., Al-Obaidli, F., Romeo, S., et al. (2016). Convkn at semeval-2016 task 3: Answer and question selection for question answering on arabic and english fora. In *Proceedings of the 10th international workshop on semantic evaluation, semeval@naacl-hlt 2016, San Diego, CA, USA, june 16–17, 2016* (pp. 896–903). http://aclweb.org/anthology/S/S16/S16-1138.pdf.

Belinkov, Y., Mohtarami, M., Cyphers, S., & Glass, J. R. (2015). Vectorslu: A continuous word vector approach to answer selection in community question answering systems. In *Proceedings of the 9th international workshop on semantic evaluation, semeval@naacl-hlt 2015, Denver, Colorado, USA, june 4–5, 2015* (pp. 282–287). http://aclweb.org/anthology/S/S15/S15-2048.pdf.

Camacho-Collados, J., Delli Bovi, C., Espinosa Anke, L., Oramas, S., Pasini, T., Santus, E., et al. (2018). SemEval-2018 Task 9: Hypernym discovery. In *Proceedings of the 12th international workshop on semantic evaluation* (pp. 712–724). New Orleans, Louisiana: Association for Computational Linguistics. http://www.aclweb.org/anthology/S18-1115.

Camacho-Collados, J., Pilehvar, M. T., Collier, N., & Navigli, R. (2017). SemEval-2017 Task 2: Multilingual and cross-lingual semantic word similarity. In *Proceedings of the 11th international workshop on semantic evaluation (SemEval-2017)* (pp. 15–26). Vancouver, Canada: Association for Computational Linguistics. http://www.aclweb.org/anthology/S17-2002.

Chung, J., Gülçehre, Ç., Cho, K., & Bengio, Y. (2014). Empirical evaluation of gated recurrent neural networks on sequence modeling. arXiv:1412.3555.

Franco-Salvador, M., Kar, S., Solorio, T., & Rosso, P. (2016). UH-PRHLT at semeval-2016 task 3: Combining lexical and semantic-based features for community question answering. In *Proceedings of the 10th international workshop on semantic evaluation, semeval@naacl-hlt 2016, San Diego, CA, USA, june 16–17, 2016* (pp. 814–821). http://aclweb.org/anthology/S/S16/S16-1126.pdf.

Gleize, M., & Grau, B. (2013). LIMSIILES: Basic english substitution for student answer assessment at SemEval 2013. In *Second joint conference on lexical and computational semantics (*sem), volume 2: Proceedings of the seventh international workshop on semantic evaluation (semeval 2013)* (pp. 598–602). Atlanta, Georgia, USA: Association for Computational Linguistics.

Gomaa, G., & Fahmy, A. (2014). Arabic short answer scoring with effective feedback for students. *International journal of computer applications (09758887), volume 86 no 2.*

Gomaa, W. H., & Fahmy, A. A. (2013). Article: A survey of text similarity approaches. *International Journal of Computer Applications, 68*(13), 13–18. Full text available

Higurashi, T., Kobayashi, H., Masuyama, T., & Murao, K. (2018). Extractive headline generation based on learning to rank for community question answering. In *Proceedings of the 27th international conference on computational linguistics* (pp. 1742–1753). Association for Computational Linguistics. http://aclweb.org/anthology/C18-1148.

Jeon, J., Croft, W. B., Lee, J. H., & Park, S. (2006). A framework to predict the quality of answers with non-textual features. In *Proceedings of the 29th annual international acm sigir conference on research and development in information retrieval* (pp. 228–235).

Joty, S. R., Màrquez, L., & Nakov, P. (2016). Joint learning with global inference for comment classification in community question answering. In *NAACL HLT 2016, the 2016 conference of the north american chapter of the association for computational linguistics: Human language technologies, San Diego California, USA, june 12–17, 2016* (pp. 703–713). http://aclweb.org/anthology/N/N16/N16-1084.pdf.

Jurgens, D., & Pilehvar, M. T. (2016). SemEval-2016 Task 14: semantic taxonomy enrichment. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)* (pp. 1092–1102). San Diego, California: Association for Computational Linguistics. http://www.aclweb.org/anthology/S16-1169.

Kashyap, A. L., Han, L., Yus, R., Sleeman, J., Satyapanich, T. W., Gandhi, S. R., & Finin, T. (2016). Robust semantic text similarity using LSA, machine learning and linguistic resources. *Language Resources and Evaluation, Special Issue: Computational Semantic Analysis of Language: SemEval-2014 and Beyond, 50*(1), 125–161.

Kateryna Tymoshenko, D. B., & Moschitti. , A. (2016). Learning to rank nonfactoid answers: Comment selection in web forums.. In *Proceedings of the 25th acm international on conference on information and knowledge management. Indianapolis, Indiana, USA, CIKM 16* (pp. 2049–2052).

Kingma, D. P., & Ba, J. (2014). *Adam: A method for stochastic optimization*. arXiv: 1412.6980.

Krebs, A., Lenci, A., & Paperno, D. (2018). SemEval-2018 Task 10: Capturing discriminative attributes. In *Proceedings of The 12th International Workshop on Semantic Evaluation* (pp. 732–740). New Orleans, Louisiana: Association for Computational Linguistics. http://www.aclweb.org/anthology/S18-1117.

Lahbari, I., Rodríguez, H., & Alaoui, S. O. E. (2018). Not all the questions are (equally) difficult. an hybrid approach to CQA in arabic. *Procesamiento del Lenguaje Natural, 60*, 21–28. http://journal.sepln.org/sepln/ojs/ojs/index.php/pln/article/view/5554.

Liu, F., Moffat, A., Baldwin, T., & Zhang, X. (2016). Quit while ahead: Evaluating truncated rankings. In *Proceedings of the 39th international ACM SIGIR conference on research and development in information retrieval SIGIR '16* (pp. 953–956). New York, NY, USA: ACM. doi:10.1145/2911451.2914737.

Magooda, A., Gomaa, A., Mahgoub, A. Y., Ahmed, H., Rashwan, M., Raafat, H. M., et al. (2016). Rdi_team at semeval-2016 task 3: RDI unsupervised framework for text ranking. In *Proceedings of the 10th international workshop on semantic evaluation, semeval@naacl-hlt 2016, San Diego, CA, USA, june 16–17, 2016* (pp. 822–827). http://aclweb.org/anthology/S/S16/S16-1127.pdf.

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@naacl-hlt 2016, San Diego, CA, USA, june 16–17, 2016* (pp. 866–871). http://aclweb.org/anthology/S/S16/S16-1134.pdf.

May, J. (2016). SemEval-2016 Task 8: Meaning representation parsing. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)* (pp. 1063–1073). San Diego, California: Association for Computational Linguistics. http://www.aclweb.org/anthology/S16-1166.

Mihaylov, T., & Nakov, P. (2016). Semanticz at semeval-2016 task 3: Ranking relevant answers in community question answering using semantic similarity based on fine-tuned word embeddings. In *Proceedings of the 10th international workshop on semantic evaluation, semeval@naacl-hlt 2016, San Diego, CA, USA, june 16–17, 2016* (pp. 879–886). http://aclweb.org/anthology/S/S16/S16-1136.pdf.

Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013). Efficient estimation of word representations in vector space. arXiv:1301.3781.

Mohtarami, M., Belinkov, Y., Hsu, W., Zhang, Y., Lei, T., Bar, K., … 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@naacl-hlt 2016, San Diego, CA, USA, june 16–17, 2016* (pp. 828–835). http://aclweb.org/anthology/S/S16/S16-1128.pdf.

Mollá, D., & Vicedo, J. L. (2007). Question answering in restricted domains: An overview. *Comput. Linguist., 33*(1), 41–61. doi:10.1162/coli.2007.33.1.41.

Nakov, P., Hoogeveen, D., Mrquez, 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). Vancouver, Canada: Association for Computational Linguistics. http://www.aclweb.org/anthology/S17-2003.

Nicosia, M., Filice, S., no, A. B.-C., Saleh, I., Mubarak, H., Gao, W., … Magdy, W. (2015). QCRI: Answer selection for community question answering - experiments for arabic and english. In *Proceedings of 9th international workshop on semantic evaluation (semeval-2015)* (pp. 203–209).

Ostermann, S., Roth, M., Modi, A., Thater, S., & Pinkal, M. (2018). SemEval-2018 Task 11: Machine comprehension using commonsense knowledge. In *Proceedings of The 12th International Workshop on Semantic Evaluation* (pp. 747–757). New Orleans, Louisiana: Association for Computational Linguistics. http://www.aclweb.org/anthology/S18-1119.

Romeo, S., Da San Martino, G., Barrón-Cedeño, A., & Moschitti, A. (2018). A flexible, efficient and accurate framework for community question answering pipelines. In *Proceedings of ACL 2018, system demonstrations* (pp. 134–139). Association for Computational Linguistics. http://aclweb.org/anthology/P18-4023.

Rosenthal, S., Farra, N., & Nakov, P. (2017). SemEval-2017 Task 4: Sentiment Analysis in Twitter. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)* (pp. 502–518). Vancouver, Canada: Association for Computational Linguistics. http://www.aclweb.org/anthology/S17-2088.

Ruppenhofer, J., Sporleder, C., Morante, R., Baker, C., & Palmer, M. (2010). SemEval-2010 Task 10: Linking events and their participants in discourse. In *Proceedings of the 5th International Workshop on Semantic Evaluation* (pp. 45–50). Uppsala, Sweden: Association for Computational Linguistics. http://www.aclweb.org/anthology/S10-1008.

Severyn, A., & Moschitti, A. (2016). Modeling relational information in question-answer pairs with convolutional neural networks. arXiv:1604.01178.

Taghva, K., Elkhoury, R., & Coombs, J. S. (2005). Arabic stemming without A root dictionary. In *International symposium on information technology: Coding and computing (ITCC 2005), volume 1, 4–6 april 2005, Las Vegas, Nevada, USA* (pp. 152–157). IEEE Computer Society. doi:10.1109/ITCC.2005.90.

Tan, M., Xiang, B., & Zhou, B. (2015). Lstm-based deep learning models for non-factoid answer selection. arXiv:1511.04108.

Torki, M., Hasanain, M., & Elsayed, T. (2017). QU-BIGIR at semeval 2017 task 3: Using similarity features for arabic community question answering forums. In S. Bethard, M. Carpuat, M. Apidianaki, S. M. Mohammad, D. M. Cer, & D. Jurgens (Eds.), *Proceedings of the 11th international workshop on semantic evaluation, semeval@acl 2017, Vancouver, Canada, august 3–4, 2017* (pp. 360–364). Association for Computational Linguistics. doi:10.18653/v1/S17-2059.

Tran, Q. H., Tran, V., Vu, T., Nguyen, M., & Pham, S. B. (2015). JAIST: Combining multiple features for Answer Selection in Community Question Answering. In *Proceedings of 9th international workshop on semantic evaluation (semeval-2015)* (pp. 215–219).

Unger, C., Forascu, C., López, V., Ngomo, A. N., Cabrio, E., Cimiano, P., & Walter, S. (2015). Question answering over linked data (QALD-5). *Working notes of CLEF 2015 - conference and labs of the evaluation forum, Toulouse, France, september 8–11, 2015*. http://ceur-ws.org/Vol-1391/173-CR.pdf.

Uva, A., Bonadiman, D., & Moschitti, A. (2018). Injecting relational structural representation in neural networks for question similarity. In *Proceedings of the 56th annual meeting of the association for computational linguistics (volume 2: Short papers)* (pp. 285–291). Association for Computational Linguistics. http://aclweb.org/anthology/P18-2046.

Vo, N. P. A., Magnolini, S., & Popescu, O. (2015). FBK-HLT: A new framework for semantic textual similarity. In *Proceedings of 9th international workshop on semantic evaluation (semeval-2015)* (pp. 102–106).

Wang, H., & Poupart, P. (2016). Overfitting at semeval-2016 task 3: Detecting semantically similar questions in community question answering forums with word embeddings. In *Proceedings of the 10th international workshop on semantic evaluation, semeval@naacl-hlt 2016, San Diego, CA, USA, june 16–17, 2016* (pp. 861–865). http://aclweb.org/anthology/S/S16/S16-1133.pdf.

Wang, W., Yan, M., & Wu, C. (2018). Multi-granularity hierarchical attention fusion networks for reading comprehension and question answering. In *Proceedings of the 56th annual meeting of the association for computational linguistics (volume 1: Long papers)* (pp. 1705–1714). Association for Computational Linguistics. http://aclweb.org/anthology/P18-1158.

Wu, G., & Lan, M. (2016). ECNU at semeval-2016 task 3: Exploring traditional method and deep learning method for question retrieval and answer ranking in community question answering. In *Proceedings of the 10th international workshop on semantic evaluation, semeval@naacl-hlt 2016, San Diego, CA, USA, june 16–17, 2016* (pp. 872–878). http://aclweb.org/anthology/S/S16/S16-1135.pdf.

Wu, J., Yang, Y., Deng, C., Tang, H., Wang, B., Sun, H., et al. (2019). *Sogou machine reading comprehension toolkit*. arXiv: 1903.11848.

Wu, W., Sun, X., & Wang, H. (2018). Question condensing networks for answer selection in community question answering. In *Proceedings of the 56th annual meeting of the association for computational linguistics (volume 1: Long papers)* (pp. 1746–1755). Association for Computational Linguistics. http://aclweb.org/anthology/P18-1162.

Wu, Y., & Zhang, M. (2016). ICL00 at semeval-2016 task 3: Translation-based method for CQA system. In *Proceedings of the 10th international workshop on semantic evaluation, semeval@naacl-hlt 2016, San Diego, CA, USA, june 16–17, 2016* (pp. 857–860). http://aclweb.org/anthology/S/S16/S16-1132.pdf.

Yi, L., Wang, J., & Lan, M. (2015). ECNU: Using multiple sources of CQA-based information for answers selection and YES/NO response inference. In *Proceedings of 9th international workshop on semantic evaluation (semeval-2015)* (pp. 236–241).

Zhao, J., Zhu, T. T., & Lan, M. (2014). ECNU: One stone two birds: ensemble of heterogenous measures for semantic relatedness and textual entailment. In *Proceedings of 8th international workshop on semantic evaluation (semeval-2014)* (pp. 271–277).

Zhou, X., Hu, B., Chen, Q., Tang, B., & Wang, X. (2015). Answer sequence learning with neural networks for answer selection in community question answering. In *Proceedings of the 53rd annual meeting of the association for computational linguistics and the 7th international joint conference on natural language processing of the asian federation of natural language processing, ACL 2015, july 26–31, 2015, Beijing, China, volume 2: Short papers* (pp. 713–718). http://aclweb.org/anthology/P/P15/P15-2117.pdf.