A Multiple-Instance Learning Approach to Sentence Selection for Question Ranking

Salvatore Romeo, Giovanni Da San Martino,
Alberto Barrón-Cedeño, and Alessandro Moschitti

Qatar Computing Research Institute, HBKU, Doha, Qatar
{sromeo, gmartino, albarron, amoschitti}@hbku.edu.qa

Abstract. In example-based retrieval a system is queried with a document aiming to retrieve other similar or relevant documents. We address an instance of this problem: question retrieval in community Question Answering (cQA) forums. In this scenario, both the document collection and the queries are relatively short multi-sentence documents subject to noise and redundancy, which makes it harder for learning-to-rank algorithms to build upon the proper text representation.
In order to only exploit the relevant fragments of the query and collection documents, we treat them as a sequence of sentences, in a multiple-instance learning fashion. By automatically pre-selecting the best sentences for our tree-kernel-based learning model, we improve over using full text performance on the dataset of the 2016 SemEval cQA challenge in terms of accuracy and speed, reaching the state of the art.

1 Introduction

The most common text-based search engines operate with relatively short queries: a user inputs keywords or a short phrase into the engine expecting to obtain a (small set of) document(s) satisfying her information need. In other retrieval scenarios (e.g., in near-duplicate detection [29]), the query is yet another document, similar in nature to those in the document collection. Unlike other genres, in social media —such as cQA forums— the documents are short, informal, and noisy (e.g., ungrammatical, redundant, and off-topic). As a result, the contents from both query and collection documents have to be carefully filtered and selected in order to come out with proper representations for learning-to-rank algorithms.

We experiment with the evaluation framework of the SemEval 2016 Task 3 on cQA [26]. Task B of the challenge can be defined as follows. Let D be a collection of questions, previously posted to the forum. Let q be a freshly-posted question. Rank the documents in D according to their relevance against q. In general, a document $d \in D$ has associated a thread of answers, previously posted by other users. Therefore, retrieving a question $d \in D$ which is equivalent or similar to q may fulfill the user's information need and may prevent the posting of a near-duplicate question to the forum. We address this task as a learning-to-rank problem. Our system relies on a paraphrase identification model based on tree kernels (TK) applied to relational syntactic structures [15]. Such approach

was originally intended to deal with pairs of sentences, whereas questions in cQA are in general multi-sentence noisy paragraphs.

Our main contribution is the selection of the best sentences to learn the model upon, and we do it on the basis of a two-step multiple-instance learning strategy (MIL). Firstly, each question gathers together a number of instances (sentences) from which we learn a fast model for identifying the least-noisy, most-relevant ones only using vectorial representations. Secondly, we compute a more expensive syntactic and vectorial representations of the resulting text to learn a binary classifier at question level. We use the latter as a reranking function of our retrieval system. Sentence selection is performed with: (*i*) unsupervised methods based on scalar products with and without TF×IDF weights and (*ii*) supervised approaches based on an automatic selector of sentence pairs. Our experiments show that the MIL-based sentence selection model produces a better representation for the question re-ranking model based on TKs. Sentence selection allows our re-ranker to improve by up to 1.82 MAP points over using the full texts and potential improve the best system of the SemEval challenge.

The rest of the paper is distributed as follows. Section 2 puts the ground on tree kernels and multiple instance learning. Section 3 describes our multiple-instance learning approach to both sentence selection and question re-ranking. Section 4 discusses the experimental settings and the obtained results. Section 5 overviews related work. Finally, Section 6 includes conclusions and final remarks.

2 Background

In this section, we introduce the concepts that we use in the remainder of the paper: tree kernels in Section 2.1 and multiple-instance learning in Section 2.2.

2.1 Tree Kernel Models

Kernel methods do not require an explicit data representation in terms of feature vectors. The input of a kernel method is a function —called kernel function— representing the degree of similarity between two items. Kernel machines, e.g., SVM, can be expressed as a convex optimization problem, provided that the kernel function is positive semidefinite [8]. Tree kernels are functions that measure the similarity between tree structures. In this work, we apply the partial tree kernel [24], which computes the similarity between two trees in terms of the number of their shared subtrees, as follows:

$$K(T_1, T_2) = \sum_{n_1 \in N_{T_1}} \sum_{n_2 \in N_{T_2}} \Delta(n_1, n_2), \quad (1)$$

where N_{T_1} (N_{T_2}) is the set of nodes in tree T_1 (T_2). $\Delta(n_1, n_2)$ is computed as

$$\Delta(n_1, n_2) = \begin{cases} 0 & \text{if the labels in nodes } n_1 \text{ and } n_2 \text{ are different} \quad (2) \\ 1 + \sum_{\boldsymbol{B}_1, \boldsymbol{B}_2 \atop |\boldsymbol{B}_1| = |\boldsymbol{B}_2|} \prod_{i=1}^{|\boldsymbol{B}_1|} \Delta(c_{n_1}[\boldsymbol{B}_{1i}], c_{n_2}[\boldsymbol{B}_{2i}]) & \text{otherwise} \quad (3) \end{cases}$$

where $\boldsymbol{B}_1 = \langle B_{11}, B_{12}, B_{13}, .. \rangle$ and $\boldsymbol{B}_2 = \langle B_{21}, B_{22}, B_{23}, .. \rangle$ are index sequences associated with the ordered child sequences c_{n_1} of n_1 and c_{n_2} of n_2, respectively. $\boldsymbol{B}_{1i}$ and $\boldsymbol{B}_{2i}$ point to the i-th children in the two sequences, and $|\boldsymbol{B}|$ represents the length of the sequence $\boldsymbol{B}$.

2.2 Multiple Instance Learning

In multiple-instance learning examples are represented as sets (bags) of instances (feature vectors) [2]. In supervised learning, the bag has an associated target label, whereas the label of its members remains unknown. Indeed, some of the instances conforming a bag may be meaningless to discriminate the bag's target label. MIL can be formalized as follows. Let $\{X_1, \ldots, X_L\} = \mathcal{X}$ be the set of examples (bags) and $\{x_1, \ldots, x_l\}$ be the set of instances of an example $X \in \mathcal{X}$ (here l varies across examples). Given a training set $\{(X_1, Y_1), \ldots, (X_L, Y_L)\}$, where $Y_i \in \mathcal{Y}$ is the label of X_i, the goal is to learn a function $F : \mathcal{X} \to \mathcal{Y}$.

MIL approaches can be roughly divided into instance- and bag-level. In the instance-level approaches, the decision $F(X)$ results from the aggregation of the decisions of local discriminative functions $f(x_i)$ $\forall x_i \in X$ (cf.[6, 10] for examples). In the bag-level approaches X is mapped into a suitable representation and classified directly. Two bag-level classes have been proposed [2]:

(i) the *embedded space* paradigm, where all the instances are first mapped into a single feature vector and then a standard learning technique is applied. Typically, the representation is obtained by clustering the instances (e.g., k-means), and then forming a vectorial representation of the bag as a function of the clustering, e.g., a vector where the i-th element corresponds to the number of instances represented by the i-th cluster [28].

(ii) The *bag space* paradigm, which requires the definition of a distance or kernel function between bags for applying a learning algorithm, such as k-NN and SVMs. For example, [17] proposed the following kernel:

$$K(X, X') = \sum_{x \in X, x' \in X'} k(x, x')^p, \quad (4)$$

where $k(x, x')$ is a kernel function between instances and the kernel parameter p allows for combinations of features within the kernel $k()$.

We can cast question re-ranking as an instance of a MIL problem using kernels as similarity functions. The set of bags in our setting is composed of pairs of query and forum questions: $X = (q, d)$. Let $S_q = \{s_{q,1}, \ldots, s_{q,|S_q|}\}$ ($S_d = \{s_{d,1}, \ldots, s_{d,|S_d|}\}$) be the set of sentences in q (d). Then the instances are all the pairs of sentences $x_{i,j} = (s_{q,i}, s_{d,j})$.

3 Question Re-Ranking Model

3.1 Base Model

Our base learning model is a function $c : \mathcal{Q} \times \mathcal{Q} \rightarrow \mathbb{R}$. Since a document d in the collection is simply labeled as relevant or irrelevant with respect to q, we use a binary SVM [20] whose classification is the sign of the $c(\cdot)$ function. (We also explored with SVMrank [21], but the results were comparable.) The kernel function input to the SVM is a combination of two kernel functions on the parse-tree representations and vectors of similarities. We depart from the model proposed in [30], which combines the tree kernels K^T of Eq. (1):

$$K((q_I, d_I), (q_J, d_J)) = K^T(t(q_I, d_I)), t(q_J, d_J))) + K^T(t(d_I, q_I)), t(d_J, q_J))), \quad (5)$$

where d_I and d_J are the I^{th} and J^{th} retrieved questions and $t(x_1, x_2)$ extracts the syntactic tree from text x_1 and enriches it with REL tags. A REL tag is added to the words shared by x_1 and x_2. The REL tag is propagated up to the phrase level in the syntactic tree [15, 30]. Figure 1 (bottom) shows an example. Eq. (5) is the sum of two kernels applied to two $\{q, d\}$ pairs: one partial-tree kernel applied to the two query questions and one to the two forum questions.

To refine the outcome, we enhance the TK-based model on syntactic trees with 20 similarities $sim(q, d)$ at lexical level [27]. We use word n-grams ($n = [1, \ldots, 4]$), after stopword removal, to compute greedy string tiling [34], longest common subsequence [1], Jaccard coefficient [18], word containment [23], and cosine. We also include a similarity over the syntactic trees of the pair $\{q, d\}$ using the partial tree kernel, i.e., $K^T(t(q, d), t(d, q))$. Note that the operands of the kernel function are members of the same pair. The corpus includes the position of question d in the ranking obtained when the forum is queried with q with the Google search engine. We integrate this feature as the inverse of the position of d. All these similarities are used over an RBF kernel function [25].

3.2 A Multiple-Instance Approach to Question Re-Ranking

We integrate the model in Section 3.1 with a two-step MIL approach [17]. Firstly, we follow the instance-based paradigm, in which the instances are pairs of sentences $\{s_q, s_d\}$. Secondly, we follow the embedded space paradigm to build document-level classifiers, out of which the final ranking is computed.

Let $S \subseteq S_q \times S_d$ be a subset of size u of the Cartesian product between S_q and S_d; i.e., S is the set of selected sentences (we use S_X when we refer to a specific example X). Let $q^* = \prod(\{s_{q,i} | (s_{q,i}, s_{d,j}) \in S\})$ be the concatenation of the sentences in S_q appearing in S ($\prod$ denotes the concatenation operator). Similarly, let d^* be the concatenation of the sentences in S_d. We apply a kernel function to pairs (q^*, d^*) instead of pairs (q, d).

We now show the relationship between our approach and that of Eq. (4). The kernel in Eq. (5) is a combination of tree kernels, including the one in Eq. (1). For simplicity, we focus our discussion on Eq. (1), which can be decomposed as

$$\sum_{n_2 \in N_{T_2}} \Delta(r(T_1), n_2) + \sum_{n_1 \in N_{T_1} \setminus r(T_1)} \Delta(n_1, r(T_2)) + \sum_{n_1 \in N_{T_1} \setminus r(T_1)} \sum_{n_2 \in N_{T_2} \setminus r(T_2)} \Delta(n_1, n_2)$$

where $r(T)$ is the root of a tree T. The parse trees of all the sentences in the text hang from the root-labeled node, which is always the same and unique in every tree. As a consequence, considering the definition of $\Delta()$ in Eqs. (2) and (3), Eq. (1) can be further simplified as

$$\Delta(r(T_1), r(T_2)) + \sum_{n_1 \in N_{T_1} \setminus r(T_1), n_2 \in N_{T_2} \setminus r(T_2)} \Delta(n_1, n_2)$$

Thus, the kernel between two query questions, according to Eq. (1), would be

$$= \Delta(r(T(q_1^*)), r(T(q_2^*))) + \sum_{\{s_1 | (s_1,s_2) \in S_{q_1^*}\}} \sum_{\{s_1' | (s_1,s_2) \in S_{q_2^*}\}} \sum_{n_1 \in N_{T(s_1)}} \sum_{n_2 \in N_{T(s_2)}} \Delta(n_1, n_2)$$

$$= \Delta(r(T(q_1^*)), r(T(q_2^*))) + \sum_{\{s_1 | (s_1,s_2) \in S_{q_1^*}\}} \sum_{\{s_1' | (s_1,s_2) \in S_{q_2^*}\}} K^T(s_1, s_2)$$

where $T(x)$ is a function that creates a parse tree from a sentence x. As we are dealing with multiple-sentence documents, each T includes an additional root node that links together all the sentences' trees into a macro-tree. The second term of the summation resembles Eq. (4), but in this case the kernel is computed only on the top-u pairs.

The core function of the model is the TK and we select the texts representing q and d before feeding them into the model. We aim to identify those sentences which better represent each question towards the learning process to produce S. Our sentence-selection is based on a scoring function $c_s : S_q \times S_d \rightarrow \mathbb{R}$, which differs slightly from the $c(\cdot)$ function described in Section 3.1. The target label of the pair of sentences is the one of the corresponding bag [6]. We use the same similarities as in the question-level model —plus four new features: given the position of a sentence s in a question, we consider three Boolean features: whether s appears (*i*) in position 1, (*ii*) between positions 2 and 4 (inclusive), or (*iii*) after position 4. These features are duplicated for both s_q and s_d. An additional real-valued feature computes $1/position$. Hereinafter, we will call them *positional features*. We do not use TKs in the sentence-level classifier as in preliminary experiments (not reported), the outcome of the classifier deteriorated.

Finally, given a pair $\{q, d\}$, we compute $c(s_{q,i}, s_{d,j})$ and use the score to rank sentences: only the top-k sentence pairs are used to represent the question in the final re-ranking process. Figure 1 shows the automatically-selected sentences from a pair $\{q, d\}$ and the resulting parse-tree representation for the ranking of d. As observed, sentences which give context and are not essential to estimate the relevance of d are discarded from the parse-tree representation. The scores for the training set are computed by 5-fold cross validation. The scores for the development and test sets are obtained by holdout, after learning on the training set. Our MIL approach lies between the two mentioned paradigms, since it extracts a representation for the bag that depends not only on the instances themselves, but also on the prediction scores of a classifier at instance level.

q : **car taking to india.** I wish to take my Car(Toyota corolla 2003) to india; is it expensive?

d : Shiping CAR from Qatar to India. I am using Nissan Altima for past two years. **I am planning to settle back India. Is it possible to ship my car to India?** Is it advisable. Any one did earlier.

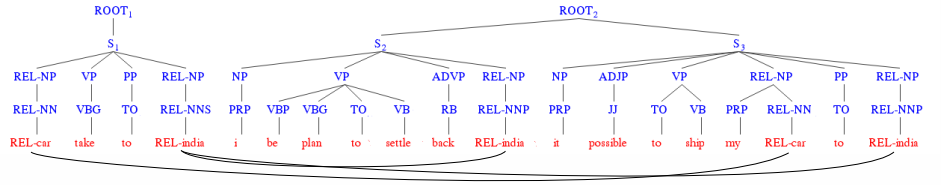


Fig. 1. Top: a pair of questions $\{q, d\}$ with automatically-selected sentences. One sentence is selected from q and two from d (highlighted). Bottom: representation of the questions' selected sentences as syntactic macro-trees (including multiple sentences). The representation is enriched with REL tags linking matches *car* and *india*.

4 Experiments

In this section, we present and discuss the results obtained with our model. We describe our evaluation framework in Section 4.1. The experiments both at sentence and at question level are discussed in Sections 4.2 and 4.3.

4.1 Evaluation Framework

We use the SemEval 2016 cQA corpus and evaluation settings to run our experiments [26]. This corpus contains a pool of 387 query questions, each of which includes 10 potentially-related forum questions. The forum questions were originally gathered using the Google search engine, which represents the task baseline. The binary gold annotations —Relevant or not— were crowdsourced. The class distribution is 40% relevant vs 60% irrelevant. We use the same training/dev/test partition as in the original dataset.[1] Following [26], we evaluate with Mean Average Precision (MAP), and Mean Reciprocal Rank (MRR).

We employ binary SVMs using the KeLP toolkit [13] in all the experiments. The TK over the parse trees is complemented with an RBF kernel over the similarity features. In all the experiments we set the C parameter of the SVMs to 1 and the parameters of tree and RBF kernels to the default values.

4.2 Selecting Sentences

First, we describe the experiments on sentence selection using the approaches from Section 3.2. We annotated sentence pairs from a subset of questions with CrowdFlower[2] to generate a gold standard to evaluate our sentence-level classifier. We selected only the 25 pairs of questions in the development set in which

[1] This corpus is available at `http://alt.qcri.org/semeval2016/task3/`.

[2] `http://www.crowdflower.com/`

Classifier	Acc	P	R	F1	MAP	MRR
TFIDF	-	-	-	-	60.83	63.43
sim_{RBF}	65.88	44.44	14.29	21.62	60.15	64.22
$sim_{RBF} + pos_{lin}$	68.24	53.85	25.00	34.15	61.13	64.22
$sim_{RBF} + pos_{RBF}$	71.76	59.09	46.43	52.00	62.84	66.67

Table 1. Performance of the sentence-level classifier with various feature combinations.

the forum question contained five or more sentences. The annotators were presented with one query-question sentence and five related-question sentences. The task consisted of determining which of the related sentences expressed the same information or idea as the query one. Each instance was annotated three times, with an inter-annotator agreement of 85.33.[3]

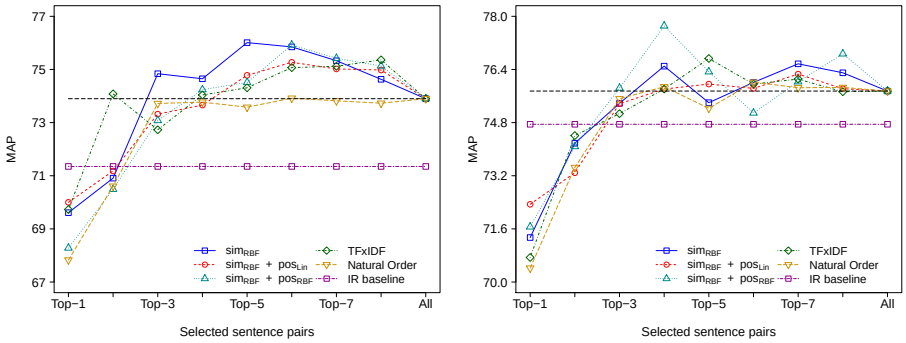
We selected sentences with SVMs, considering three different feature sets and kernel settings: (*i*) an RBF kernel on similarities (sim_{RBF}), (*ii*) a linear combination of similarities with a linear kernel on positional features ($sim_{RBF} + pos_{lin}$), and (*iii*) a linear combination of kernel (*i*) with an RBF kernel on positional features ($sim_{RBF} + pos_{lin}$). We attached the Google-provided position to the positional features. The score of the unsupervised model is computed as the cosine similarity between the TF×IDF vectors of each pair of sentences (TFIDF).

Table 1 shows the performance of the different configurations. Comparing the models using the positional features or not, we observe that such features improve the performance w.r.t. all the evaluation metrics. The performance of TFIDF in terms of MAP is similar to the ones of classifiers sim_{RBF} and $sim_{RBF} + pos_{lin}$. Using the positional features in an RBF kernel produces a better performance than other models, obtaining an improvement in terms of MAP equal to 2.01, 2.62 and 1.71, w.r.t. TFIDF, sim_{RBF} and $sim_{RBF} + pos_{lin}$, respectively.

4.3 Ranking Questions

We focus the rest of the experiments on the impact of the sentence selection for generating smaller trees to be used in TKs. We ran one question re-ranker feeding the TKs with the outcome of each of the sentence classifiers at hand to find out if MAP can be improved by selecting sentences. We kept the original input texts to compute the similarity features. Figures 2(a) and 2(b) show the results of the re-rankers obtained on the development and test sets with increasing number of selected sentences. For comparison, the MAP obtained when considering full texts —without any sentence selection— is 73.60 on the development set and 75.89 on the test set. They are represented in the converging points on the right-hand side of the plots. The natural order is our sentence selection baseline —k sentences are taken from left to right. Its best performance is achieved with 6 sentences: MAP of 73.92 and 76.02 on the development and test sets, respectively. On dev. set (Figure 2(a)), the best model is sim_{RBF}, which performs best with 5 sentences, i.e., a MAP of 76.01. The second best system

[3] This dataset is available at `http://alt.qcri.org/resources/iyas`.



(a) On the development set.

(b) On the test set.

Fig. 2. MAP evolution for different sentence selection strategies. *All* stands for the system considering full texts (without sentence selection).

is $sim_{RBF} + pos_{RBF}$, reaching the best outcome with 6 sentences, for a MAP of 75.92. Models $sim_{RBF} + pos_{lin}$ and TFIDF show the best results only until 6 and 8 sentences are used, with MAP values of 75.27 and 75.36, respectively. In general, identifying the most similar sentence pairs in advance allows for the best results; and the least sentences considered, the faster the TK operates.

Regarding the results on the test set (Figure 2(b)), the best performance is obtained by $sim_{RBF} + pos_{RBF}$ with only 4 sentences: MAP = 77.71. This shows that our approach can potentially highly improve the state of the art, i.e., 76.70 (see Tab. 2). However, the different model behavior observed in dev. and test sets suggest some challenges for estimating the optimal number of sentences.

The TFIDF, S_r and $sim_{RBF} + pos_{RBF}$ approaches have similar performance, i.e., 76.73, 76.57 and 76.26 of MAP, but after using 5 or more sentences. When our best sentence selector —$sim_{RBF} + pos_{RBF}$— is used, our model outperforms the best systems submitted to SemEval (cf. Table 2; Section 5) —being the only statistically different to the IR baseline (confidence=90%).

Finally, it should be noted that selecting the sentences to represent q and d not only boosts the performance of our question ranker but, as a side effect, applying tree kernels to shorter text, makes training/testing up to 30% faster (e.g., when using our most accurate model).

5 Related Work

Different approaches have been proposed to overcome the lexical chasm when assessing the similarity between two questions. Early approaches used statistical machine translation (SMT) techniques to compute the semantic similarity between two questions. For instance, [19] used a language model based on word translation probabilities to compute the likelihood of generating a query question given a target (forum) question. [35] showed that models based on phrases

Classifier		MAP	MRR
UH-PRHLT-primary	[16]	76.70	83.02
ConvKN-primary	[5]	76.02	84.64
Kelp-primary	[14]	75.83	82.71
IR Baseline	[26]	74.75	83.79

Table 2. Performance of the best systems submitted to SemEval 2016 Task 3(B) on question ranking; i.e., on our test set (cf. Section 5 for models' details).

are more effective than models based on words, as they are able to capture contextual information. However, approaches based on SMT typically require large amounts of data for parameter estimation.

Both [7] and [12] presented algorithms that try to go beyond simple text representation. [7] compute the similarity between two questions on Yahoo! Answers by using a smoothed language model that exploits the category structure of the forum. [12] searched for questions that are semantically similar to the user's question by identifying the question's topic and focus.

[33] presented an approach exploiting the questions' syntactic information. They proposed to find semantically-related questions by computing the similarity between their syntactic-tree representations. The tree similarity is computed as the number of sub-structures shared between two trees. The main difference with respect to our model is that we use more complex structural models, encoding relational structures and processing them by means of tree kernels. The latter captures effective structure relations, which boosts the performance of our reranker based on standard features.

Recent work has shown the effectiveness of neural models for question similarity [11] in cQA. For instance, [11] used CNN and bag-of-words (BOW) representations of query and forum questions to compute cosine similarity scores. Recently, [4] presented a neural attention model for machine translation and showed that the attention is helpful when dealing with long sentences.

The 2016 edition of the SemEval Task 3 on cQA [26] triggered a manifold of approaches to question retrieval. The top-three participants opted for SVMs as learning models. The top-ranked [16] used SVM^{rank} [22], the first [5] and second [14] runners up used KeLP [13] to combine various kernels. The amount of knowledge these models use is pretty different. [16] relies heavily on distributed representations and semantic information sources, such as Babelnet and Framenet. The others do not. No statistically-significant differences were observed in the performance of these systems with respect to the baseline. Their performance is included in Table 2 for comparison with our results.

6 Conclusions

In this paper we described a learning-to-rank model based on tree kernels to rank a set of forum questions given a new question. Such a component allows Web forums to avoid posting near-duplicate questions and to answer to the user's

information quest at no time. We proposed a model to pre-select a subset of the sentences composing each question in order to feed them into a tree-kernel-based question-ranking model. The reason is that tree-kernel models are affected by noisy text and redundant information, which is typically added by Web users when formulating or answering forum questions.

We expressed both the sentence selection and question ranking steps as a multiple-instance learning (MIL) instantiation. Our results on the SemEval 2016 cQA corpus showed that MIL models can improve the quality of the ranking by coming out with a better representation of the documents. As a result, our tree-kernel model learn better the parameters of the ranking function (as noise is filtered out from the texts), both boosting the performance of the ranker and speeding it up. Our proposed model outperforms the top systems submitted to the SemEval 2016 task on community Question Answering, however additional work is needed to reliably estimating the best number of sentences for each test set. In the future, we would like to explore more powerful kernels such as the smoothed partial tree kernel [9] as well as the most advanced tree kernel models applied in QA, e.g., [32, 31].

References

1. Allison, L., Dix, T.: A Bit-string Longest-common-subsequence Algorithm. Inf. Process. Lett. 23(6), 305–310 (Dec 1986)
2. Amores, J.: Multiple instance classification: Review, taxonomy and comparative study. Artificial Intelligence 201, 81 – 105 (2013)
3. Association for Computational Linguistics: Proceedings of the 10th International Workshop on Semantic Evaluation, SemEval '16 (June 2016)
4. Bahdanau, D., Cho, K., Bengio, Y.: Neural Machine Translation by Jointly Learning to Align and Translate. arXiv preprint arXiv:1409.0473 (2014)
5. Barrón-Cedeño, A., Da San Martino, G., Joty, S., Moschitti, A., Al-Obaidli, F., Romeo, S., Tymoshenko, K., Uva, A.: 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 [3], pp. 896–903
6. Bunescu, R.C., Mooney, R.J.: Multiple instance learning for sparse positive bags. In: Proceedings of the 24th international conference on Machine learning. pp. 105–112. ACM (2007)
7. Cao, X., Cong, G., Cui, B., Jensen, C.S., Zhang, C.: The use of categorization information in language models for question retrieval. In: Proceedings of the 18th ACM conference on Information and knowledge management. pp. 265–274. ACM (2009)
8. Cristianini, N., Shawe-Taylor, J.: An Introduction to Support Vector Machines and Other Kernel-based Learning Methods. Cambridge University Press, 1 edn. (2000)
9. Croce, D., Moschitti, A., Basili, R.: Structured lexical similarity via convolution kernels on dependency trees. In: Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing. pp. 1034–1046. Association for Computational Linguistics, Edinburgh, Scotland, UK. (July 2011)
10. Dietterich, T.G., Lathrop, R.H., Lozano-Pérez, T.: Solving the multiple instance problem with axis-parallel rectangles. Artif. Intell. 89(1-2), 31–71 (Jan 1997)

11. dos Santos, C., Barbosa, L., Bogdanova, D., Zadrozny, B.: Learning Hybrid Representations to Retrieve Semantically Equivalent Questions. In: Zong and Strube [36], pp. 694–699
12. Duan, H., Cao, Y., Lin, C.Y., Yu, Y.: Searching Questions by Identifying Question Topic and Question Focus. In: Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics and the Human Language Technology Conference. pp. 156–164. ACL-HLT '08, Association for Computational Linguistics, Columbus, OH (June 2008)
13. Filice, S., Castellucci, G., Croce, D., Da San Martino, G., Moschitti, A., Basili, R.: KeLP: a Kernel-based Learning Platform in Java. In: Proceedings of the workshop on Machine Learning Open Source Software: Open Ecosystems. International Conference of Machine Learning, Lille, France (2015)
14. Filice, S., Croce, D., Moschitti, A., Basili, R.: KeLP at SemEval-2016 Task 3: Learning Semantic Relations between Questions and Answers. In: Proceedings of the 10th International Workshop on Semantic Evaluation [3], pp. 1116–1123
15. Filice, S., Da San Martino, G., Moschitti, A.: Structural Representations for Learning Relations between Pairs of Texts. In: Zong and Strube [36], pp. 1003–1013
16. Franco-Salvador, M., Kar, S., Solorio, T., Rosso, P.: 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 [3]
17. Gärtner, T., Flach, P.A., Kowalczyk, A., Smola, A.J.: Multi-instance kernels. In: Sammut, C., Hoffmann, A.G. (eds.) Machine Learning, Proceedings of the Nineteenth International Conference (ICML 2002), University of New South Wales, Sydney, Australia, July 8-12, 2002. pp. 179–186. Morgan Kaufmann (2002)
18. Jaccard, P.: Étude comparative de la distribution florale dans une portion des Alpes et des Jura. Bulletin del la Société Vaudoise des Sciences Naturelles 37, 547–579 (1901)
19. Jeon, J., Croft, W.B., Lee, J.H.: Finding Similar Questions in Large Question and Answer Archives. In: Herzog, O., Schek, H., Fuhr, N., Chowdhury, A., Teiken, W. (eds.) Proceedings of the 14th ACM International Conference on Information and Knowledge Management. pp. 84–90. Bremen, Germany (2005)
20. Joachims, T.: Making Large-scale Support Vector Machine Learning Practical. In: Schölkopf, B., Burges, C.J.C., Smola, A.J. (eds.) Advances in Kernel Methods, pp. 169–184. MIT Press, Cambridge, MA, USA (1999)
21. Joachims, T.: Optimizing Search Engines Using Clickthrough Data. In: Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. pp. 133–142. ACM, New York, NY (2002)
22. Joachims, T.: Training Linear SVMs in Linear Time. In: Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. pp. 217–226. KDD '06, ACM, New York, NY (2006)
23. Lyon, C., Malcolm, J., Dickerson, B.: Detecting Short Passages of Similar Text in Large Document Collections. In: Proceedings of the Conference on Empirical Methods in Natural Language Processing. pp. 118–125. EMNLP '01, Pittsburgh, PA (2001)
24. Moschitti, A.: Efficient Convolution Kernels for Dependency and Constituent Syntactic Trees. In: Proceedings of the 17th European Conference on Machine Learning. pp. 318–329. ECML '06, Springer-Verlag Berlin Heidelberg, Berlin, Germany (2006)

25. Müller, K.R., Mika, S., Rätsch, G., Tsuda, K., Schölkopf, B.: An introduction to kernel-based learning algorithms. IEEE transactions on neural networks / a publication of the IEEE Neural Networks Council 12(2), 181–201 (jan 2001)
26. Nakov, P., Màrquez, L., Moschitti, A., Magdy, W., Mubarak, H., Freihat, a., Glass, J., Randeree, B.: SemEval-2016 Task 3: Community Question Answering. In: Proceedings of the 10th International Workshop on Semantic Evaluation [3], pp. 525–545
27. Nicosia, M., Filice, S., Barrón-Cedeño, A., Saleh, I., Mubarak, H., Gao, W., Nakov, P., Da San Martino, G., Moschitti, A., Darwish, K., Màrquez, L., Joty, S., Magdy, W.: QCRI: Answer Selection for Community Question Answering - Experiments for Arabic and English. In: Proceedings of the 9th International Workshop on Semantic Evaluation. SemEval 2015, Association for Computational Linguistics, Denver, CO (2015)
28. Nowak, E., Jurie, F., Triggs, B.: Sampling Strategies for Bag-of-Features Image Classification. In: Leonardis, A., Bischof, H., Pinz, A. (eds.) Computer Vision – ECCV 2006: 9th European Conference on Computer Vision, Graz, Austria, May 7-13, 2006, Proceedings, Part IV. pp. 490–503. Springer Berlin Heidelberg, Berlin, Heidelberg (2006)
29. Potthast, M., Stein, B.: New Issues in Near-duplicate Detection. In: Preisach, C., Burkhardt, H., Schmidt-Thieme, L., Decker, R. (eds.) Data Analysis, Machine Learning and Applications. Selected papers from the 31th Annual Conference of the German Classification Society (GFKL 07). pp. 601–609. Studies in Classification, Data Analysis, and Knowledge Organization, Springer, Berlin Heidelberg (2008)
30. Severyn, A., Moschitti, A.: Structural Relationships for Large-scale Learning of Answer Re-Ranking. In: Proceedings of the 35th International ACM SIGIR Conference on Research and Development in Information Retrieval. pp. 741–750. SIGIR '12, Portland, OR (2012)
31. Tymoshenko, K., Bonadiman, D., Moschitti, A.: Convolutional neural networks vs. convolution kernels: Feature engineering for answer sentence reranking. In: Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. pp. 1268–1278. Association for Computational Linguistics, San Diego, California (June 2016)
32. Tymoshenko, K., Moschitti, A.: Assessing the impact of syntactic and semantic structures for answer passages reranking. In: Proceedings of the 24th ACM International Conference on Information and Knowledge Management, CIKM 2015, Melbourne, VIC, Australia, October 19 - 23, 2015. pp. 1451–1460 (2015)
33. Wang, K., Ming, Z., Chua, T.S.: A Syntactic Tree Matching Approach to Finding Similar Questions in Community-based QA Services. In: Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval. pp. 187–194. ACM (2009)
34. Wise, M.: YAP3: Improved Detection of Similarities in Computer Program and Other Texts. In: Proceedings of the Twenty-seventh SIGCSE Technical Symposium on Computer Science Education. pp. 130–134. SIGCSE '96, New York, NY (1996)
35. Zhou, G., Cai, L., Zhao, J., Liu, K.: Phrase-based translation model for question retrieval in community question answer archives. In: Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1. pp. 653–662 (2011)
36. Zong, C., Strube, M. (eds.): Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing. ACL-HLT '15, Association for Computational Linguistics, Beijing, China (July 2015)