# Learning to Re-Rank Questions in Community Question Answering Using Advanced Features

Giovanni Da San Martino[†], Alberto Barrón-Cedeño[†], Salvatore Romeo[†], Antonio Uva[‡], Alessandro Moschitti[†]
[†]Qatar Computing Research Institute, Hamad bin Khalifa University, Doha, Qatar
[‡]Department of Computer Science and Information Engineering, University of Trento, Trento, Italy
[†]{gmartino, albarron, sromeo, amoschitti}@qf.org.qa, [‡]antono.uva@unitn.it

## ABSTRACT

We study the impact of different types of features for question ranking in community Question Answering: bag-of-words models (BoW), syntactic tree kernels (TKs) and rank features. It should be noted that structural kernels have never been applied to the question reranking task, i.e., question to question similarity, where they have to model paraphrase relations. Additionally, the informal text, typically present in forums, poses new challenges to the use of TKs. We compare our learning to rank (L2R) algorithms against a strong baseline given by the Google rank (GR). The results show that (i) our shallow structures used in TKs are robust enough to noisy data and (ii) improving GR requires effective BoW features and TKs along with an accurate model of GR features in the used L2R algorithm.

## Keywords

Community Question Answering; Learning to Rank; Syntactic Structures

## 1. INTRODUCTION

In recent years, there has been a renewed interest in IR for Community Question Answering (cQA). This combines traditional Question Answering with a modern Web scenario, where users pose questions expecting to receive answers from other users. The most critical problem arises when an original question, asked by a user, has not been asked before. In this case the retrieval system will search for relevant comments to other questions in order to find an appropriate answer. In this noisy and complex setting, powerful search engines (e.g., Google), have a hard time to retrieve comments that can correctly answer the original question. We approach this problem by searching similar questions rather than directly retrieving comments. Indeed, if an original user question $q_o$ is similar to a sought question $q_s$, which has been answered before, we can just return the answers associated with $q_s$.

The automatic retrieval of relevant questions requires models different from typical search engines. Firstly, both queries and documents are short texts: questions may include descriptions or subquestions but they are not typically larger than one or two paragraphs. Thus, on one hand, word similarities are needed for dealing with data sparseness; on the other hand, representations based on parse-tree can be easily modeled. Secondly, the syntactic structure of the questions is rather important for learning to recognize paraphrases and thus selecting the right candidates. For instance, the corpus question, *How do I get a visa for Qatar to visit my wife?*, has roughly the same BoW representation as question *How do I get a visa for my wife to have her visit Qatar?*, but their answers are totally different. The structure of the questions can be exploited to detect the difference between these difficult cases.

The study of cQA models for question–question similarity has typically relied on annotations provided by users, which could imply low reliability. Recently, a new resource has been released for cQA for the shared task at SemEval 2016 on Answer Selection in cQA [10].[1] Task B of the challenge provides a dataset of questions, each associated with ten candidate similar questions. The candidates were (i) retrieved and ranked with the Google search engine, using the question as input in their Web API and (ii) manually labelled by human annotators. Thus, this dataset enables a reliable testing of the most advanced methods for question reranking on top of the baseline rank generated by the currently most powerful search engine.

In this paper, we provide insights on reranking the output of a Web search engine using advanced features. In particular, we use (i) text similarity features, derived from the classical BoW representations, e.g., n-grams, skip-grams; (ii) syntactic/structural relational features injected by tree kernels (TKs), which have been shown to achieve the state of the art on the related task of answer sentence reranking [12]; and (iii) features for modeling the initial rank provided by the search engine, which represents a strong baseline.

Our extensive experimentation produced the following results: (i) the BoW features based on similarity measures alone do not improve GR. (ii) Our TKs applied to questions alone outperform the models based on similarity measures and when jointly used with the rank features improve all the models. In particular, they outperform GR by 1.72 with respect to MAP (95% of statistical confidence).

## 2. RELATED WORK

The first step to automatically answer questions on cQA sites is to retrieve a set of questions similar to the user's

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

*CIKM '16, October 24–28, 2016, Indianapolis, IN, USA.*
© 2016 ACM. ISBN 978-1-4503-4073-1/16/10…$15.00
DOI: http://dx.doi.org/10.1145/2983323.2983893

[1] http://alt.qcri.org/semeval2016/task3/

**Table 1: A reranking example. For each candidate the Google rank (GR), the binary gold standard (GS) relevance, and our rank (R) are reported.**

$q_o$: What are the tourist places in Qatar? I'm likely to travel in the month of June. Just wanna know some good places to visit.

| *GR* | *GS* | *R* | *Question Text* |
|---|---|---|---|
| 1 | -1 | 8 | The Qatar banana island will be transfered by the end of 2013 to 5 stars resort called Anantara. Has anyone seen this island? Where is it? Is it near to Corniche? |
| 2 | +1 | 2 | Is there a good place here where I can spend some quality time with my friends? |
| 3 | -1 | 7 | Where is the best beach in Qatar? Maybe a silent and romantic bay? Where to go for it? |
| 4 | -1 | 9 | Any suggestions on what are the happenings in Qatar on Holidays? Something new and exciting suggestions please? |
| 5 | -1 | 3 | Where in Qatar is the best place for Snorkeling? I'm planning to go out next friday but don't know where to go. |
| 6 | -1 | 6 | Can you give me some nice places to go or fun things to do in Doha for children 17-18 years old? Where can we do some watersports (just for once, not as a member), or some quad driving? Let me know please. Thanks. |
| 7 | +1 | 1 | Which all places are there for tourists to Qatar? My nephew 18 years on visit. |
| 8 | -1 | 10 | Could you suggest the best holiday destination in the world? |
| 9 | -1 | 5 | I really would like to know where the best place to catch fish here in Qatar is. But of course from the beach. I go every week to Umsaeed but rerly i catch somthing! So experianced people your reply will be appreciated. |

**Table 2: Class distribution in the training, development, and test partitions.**

| Class | train | dev | test | overall |
|---|---|---|---|---|
| Relevant | 1,083 | 214 | 233 | 1,530 |
| Irrelevant | 1,586 | 286 | 467 | 2,339 |
| Total | 2,669 | 500 | 700 | 3,869 |

**Table 3: Distribution of Relevant and Irrelevant questions at different R ranking positions of GR.**

| R | train | dev | test | overall |
|---|---|---|---|---|
| 1 | $0.21 \pm 0.05$ | $0.24 \pm 0.07$ | $0.40 \pm 0.11$ | $0.25 \pm 0.07$ |
| 2 | $0.14 \pm 0.03$ | $0.18 \pm 0.02$ | $0.12 \pm 0.02$ | $0.14 \pm 0.03$ |
| 3 | $0.11 \pm 0.02$ | $0.10 \pm 0.01$ | $0.08 \pm 0.01$ | $0.10 \pm 0.02$ |
| 4 | $0.12 \pm 0.03$ | $0.08 \pm 0.01$ | $0.10 \pm 0.03$ | $0.11 \pm 0.03$ |
| 5 | $0.09 \pm 0.02$ | $0.09 \pm 0.01$ | $0.08 \pm 0.02$ | $0.09 \pm 0.02$ |
| 6 | $0.08 \pm 0.02$ | $0.09 \pm 0.02$ | $0.05 \pm 0.01$ | $0.08 \pm 0.02$ |
| 7 | $0.08 \pm 0.02$ | $0.07 \pm 0.01$ | $0.05 \pm 0.01$ | $0.07 \pm 0.02$ |
| 8 | $0.06 \pm 0.01$ | $0.04 \pm 0.01$ | $0.03 \pm 0.00$ | $0.05 \pm 0.01$ |
| 9 | $0.07 \pm 0.02$ | $0.06 \pm 0.01$ | $0.04 \pm 0.01$ | $0.07 \pm 0.02$ |
| 10 | $0.05 \pm 0.01$ | $0.05 \pm 0.01$ | $0.04 \pm 0.01$ | $0.05 \pm 0.01$ |

input. The set of similar questions is later used to extract possible answers for the input question. However, determining question similarity remains one of the main challenges in cQA due to problems such as the "lexical gap". Different approaches have been proposed to overcome this problem. Early methods used statistical machine translation techniques to compute semantic similarity between two questions. For instance, [15] applied a phrase-based translation model. Their experiments on Yahoo! Answers showed that models based on phrases are more effective than those using words, as they are able to capture contextual information. However, approaches based on SMT have the problem of requiring lots of data in order to estimate parameters.

Algorithms that try to go beyond simple text representation are presented in [3] and [4]. In [3] a similarity between two questions on Yahoo! Answers is computed by using a language model with a smoothing method based on the category structure of Yahoo! Answers. In [4], the authors search for semantically-similar questions identifying the question topic and focus. More specifically, they compute a similarity between the questions' topics, which represent general users interests, and the questions' focus. Here, the authors use LDA topic modeling to learn the latent semantic topics that generate question/answer pairs and use the learned topics distribution to retrieve similar questions. The quality of the ranking returned by all these systems was measured on a set of test questions from Yahoo! Answers, with question relevancy judgment annotated by users, sometimes assigned automatically based on heuristics.

The methods above only exploited language models or general knowledge given by Yahoo! categories or LDA topics, whereas we model the syntactic/semantic relations between pairs of questions using shallow syntactic parsing and lexical matching. Thus the most similar work to ours is [13], where the authors found semantically related questions by computing the similarity between the syntactic trees of the two questions. They used a tree similarity computed in terms of the number of substructures shared between two trees and the results derived on an annotated dataset showed the effectiveness of this approach.

Different from such approach, we use pairs of questions, $(q_o, q_s)$, as learning instances, thus defining relational models connecting the syntactic trees of $q_o$ and $q_s$. This way, the learning algorithms learn transformations that suggest if questions constituted by similar words have similar (paraphrases) or different semantics.

## 3. PROBLEM DESCRIPTION

Conceptually, question retrieval is not much different from a standard retrieval task. Given the asked (original) question $q_o$, a search engine seeks the Web (or a specific Web forum) for relevant webpages. In cQA, webpages are threads containing the questions $q_s$, with their user comments, where the latter can provide information for answering $q_o$. For example, Table 1 shows an original question followed by some questions retrieved by the search engine. The main difference with standard document retrieval is the document scoring function. Indeed, although both question and comments are part of the candidate webpage (thread), the question's text provides more synthetic and precise information for inferring if the candidate thread is relevant for $q_o$ or not.

Table 2 gives class-distribution statistics of the SemEval cQA corpus we used [10]. The corpus is composed of 387 user questions, each of which includes 10 potentially-related questions. The task organizers used the Google search engine, which represents also the strong baseline for the task, to select potentially relevant forum questions. Table 3 shows the distribution of relevant/irrelevant forum questions per ranking position. Although relevant questions tend to be concentrated towards the top of the GR, they are fairly spread over the entire ranking.

## 4. L2R WITH ADVANCED FEATURES

The ranking function can be implemented by a scoring function $r : Q \times Q \rightarrow \mathbb{R}$, where $Q$ is the set of questions. Function $r$ can be implemented by a linear function $r(q_o, q_s) = \vec{w} \cdot \phi(q_o, q_s)$, where $\vec{w}$ is a linear model and $\phi()$ provides a feature vector representation of pair $(q_o, q_s)$.

We adopt binary SVMs to learn $r$ from examples. Even if SVMrank has been proposed specifically for ranking tasks, we ran several experiments to compare them and observed similar results. We model $\phi(q_o, q_s)$ with three different fea-

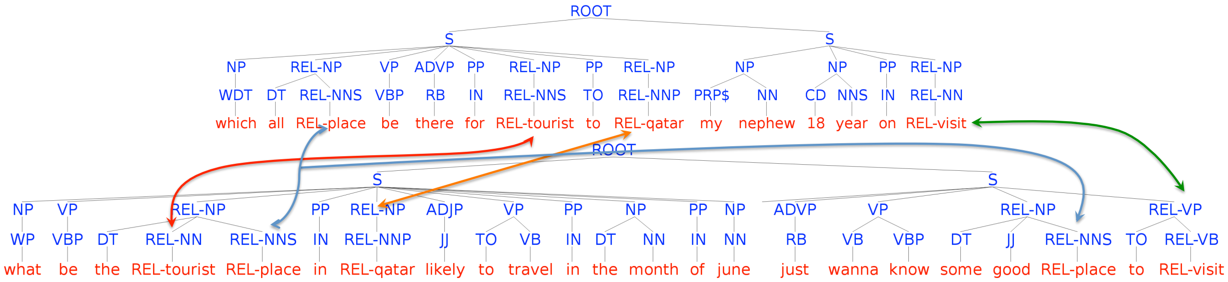
ROOT
S
S
NP REL-NP VP ADVP PP REL-NP PP REL-NP NP NP PP REL-NP
WDT DT REL-NNS VBP RB IN REL-NNS TO REL-NNP PRP$ NN CD NNS IN REL-NN
which all REL-place be there for REL-tourist to REL-qatar my nephew 18 year on REL-visit
ROOT
S
S
NP VP REL-NP PP REL-NP ADJP VP PP NP PP NP ADVP VP REL-NP REL-VP
WP VBP DT REL-NN REL-NNS IN REL-NNP JJ TO VB IN DT NN IN NN RB VB VBP DT JJ REL-NNS TO REL-VB
what be the REL-tourist REL-place in REL-qatar likely to travel in the month of june just wanna know some good REL-place to REL-visit

**Figure 1: Our representation based on syntactic trees for the $q_o$–$q_s$ pairs enriched with REL links.**

ture sets: (i) tree kernels applied to the syntactic structures of question pairs, (ii) similarity features computed between $q_o$ and $q_s$, and (iii) rank features, i.e., kernels over the question position in the rank produced by the search engine.

## 4.1 Tree Kernel Models

Tree kernels are functions that measure the similarity between tree structures. We essentially used the model proposed in [11], originally proposed for ranking passages. Different from [11], our questions may contain multiple subquestions, a subject, greetings, and elaborations; thus they are composed of several sentences. We merge the whole question text in a macro-tree using a fake root node connecting the parse trees of all the sentences. Since we represent pairs of questions, we connect the constituents of two macro-trees corresponding to $(q_o, q_s)$, respectively. For example, given the original question $q_o$ in Table 1 along with the seventh candidate, $q_s^7$, we build the graph in Figure 1. We link the two macro-trees by connecting phrases, e.g., NP, VP, PP, when there is at least lexical match between the phrases of $q_o$ and $q_s$. Note that such links are marked with the presence of a REL tag. Finally, we apply a partial tree kernel (PTK) [9] and obtain the following kernel:

$$K((q_o, q_s^i), (q_o, q_s^j)) = TK(t(q_o, q_s^i), t(q_o, q_s^j)) \\ + TK(t(q_s^i, q_o), t(q_s^j, q_o)),$$

where $t(x, y)$ extracts the syntactic tree from text $x$, enriching it with REL tags computed with respect to $y$. Thus $t$ is an asymmetric function.

## 4.2 Feature Vectors

Our L2R approach relies on three subsets of features to derive the relationship between the original and the forum questions: text similarities, PTK similarity, and Google rank.

**Text Similarity Features.** We compute a total of 20 similarities, $sim(q_o, q_j)$, using word $n$-grams ($n = [1, \ldots, 4]$), after stopword removal, using greedy string tiling [14], longest common subsequences [1], Jaccard coefficient, word containment [8], and cosine similarity.

**PTK Features.** Another similarity is obtained by comparing syntactic trees with PTK, i.e., $TK(t(q_o, q_s^i), t(q_s^i, q_o))$. Note that, different from the model in Section 4.1, PTK here is applied to the questions of the same pair and thus only produces one feature.

**Ranking-based Features.** Our ranking feature is based on the ranking generated by Google. Each forum question is located in one position in the range $[1, \ldots, 10]$. We try to exploit this information in two ways "as-is" ($pos$) or the inverse ($pos^{-1}$).[2]

**Table 4: Performance of models using ranking-based features combined with linear and RBF kernels. † shows statistically different results (at 95%) wrt. GR.**

| | DEV | | | TEST | | |
|---|---|---|---|---|---|---|
| Model | MAP | AvgRec | MRR | MAP | AvgRec | MRR |
| GR baseline | 71.35† | 86.11 | 76.67† | 74.75† | 88.30 | 83.79 |
| Sim. | 64.80 | 82.52 | 73.73 | 70.70 | 85.78 | 80.58 |
| TK | 69.97 | 86.86 | 77.73 | 73.98 | 88.90 | 82.55 |
| TK + Sim | 71.07 | 87.72 | 78.14 | 73.81 | 89.21 | 82.86 |
| **Linear Kernel** | | | | | | |
| Sim + $pos$ | 68.04 | 85.07 | 76.00 | 71.99 | 87.92 | 81.19 |
| Sim + $pos^{-1}$ | 70.17 | 85.98 | 78.17 | 75.15 | 89.19 | 84.29 |
| TK + $pos$ | 71.77 | 88.46 | 78.12 | 75.34 | 90.67 | 83.19 |
| TK + $pos^{-1}$ | 72.64 | 87.69 | 75.58 | 76.18† | 90.62 | 84.62 |
| **RBF Kernel** | | | | | | |
| Sim. + $pos$ | 70.42 | 86.38 | 78.50 | 74.61 | 89.10 | 83.81 |
| Sim. + $pos^{-1}$ | 69.82 | 85.91 | 77.17 | 74.58 | 89.09 | 83.57 |
| TK + $pos$ | 72.93 | 87.95 | 77.54 | 75.72 | 90.80 | 83.86 |
| TK + $pos^{-1}$ | 73.65† | 88.78 | 79.58† | 76.41† | 91.14 | 84.62 |

# 5. EXPERIMENTS

We use the evaluation framework of SemEval 2016 Task 3 [10]. It consists in reranking questions based on their similarity with the one provided by a user. The data enables a comparison between our models and the systems of the challenge. In a set of preliminary experiments, we compared a reranker, SVMrank [7] with a standard binary SVMs. As the results were comparable, we employed SVMs using the KeLP toolkit[3], which enables to combine our three subsets of features within different kernels; namely RBF for the similarity features, tree kernels for the parse trees, and either linear or RBF kernels for the ranking-based feature. We set the C parameter of SVMs to 1 in all the experiments and the parameters of the TKs and RBF kernel to default values.

We conducted three experiments with growing complexity for assessing the effectiveness of our different feature sets (see Section 4), with respect to GR. In agreement, with the SemEval challenge, we evaluate our rankings with Mean Average Precision (MAP), average Recall (AvgRec), and Mean Reciprocal Rank (MRR).

We tested the performance of each of the feature sets in isolation and pair-wise. Table 4 reports the obtained performance both on the development and test sets. The strong baseline is computed on GR —a product of the Google technology and its associated knowledge bases. We have two advantages over Google: (i) it is not tuned up on the specific Web data we use and (ii) it does not probably use syntactic structures in powerful algorithms such as TKs.

[2] In the dataset, the ten associated forum questions do not corresponds to the top-10, ranked according to GR as some of them were filtered out. We only report our findings with rank features projected into the range $[0, \ldots, 10]$ for brevity.

[3] `https://github.com/SAG-KeLP`

**Table 5: Performance of different rank features (all models include RBF kernel on similarities and tree kernel). † shows statistically different results (at 95%) with respect to GR. The best SemEval systems are included for comparison.**

| | DEV | | | TEST | | |
|---|---|---|---|---|---|---|
| Model | MAP | AvgRec | MRR | MAP | AvgRec | MRR |
| GR baseline | 71.35† | 86.11 | 76.67 | 74.75† | 88.30 | 83.79 |
| **Linear Kernel** | | | | | | |
| *pos* | 72.18 | 88.41 | 78.00 | 75.67 | 90.77 | 83.38 |
| $pos^{-1}$ | 73.28† | 88.47 | 80.00 | 76.28† | 90.72 | 84.62 |
| **RBF Kernel** | | | | | | |
| *pos* | 73.24† | 88.37 | 78.40 | 76.47† | 90.78 | 84.21 |
| $pos^{-1}$ | 73.60† | 88.85 | 79.67 | 75.89 | 90.57 | 84.14 |
| **SemEval top-3** | | | | | | |
| UH-PRHLT [6] | – | – | – | 76.70 | 90.31 | 83.02 |
| ConvKN [2] | – | – | – | 76.02 | 90.70 | 84.64 |
| Kelp [5] | – | – | – | 75.83 | 91.02 | 82.71 |

Our results support the hypotheses above. Indeed, the MAP of the models derived by Similarities, TK, and their combinations is below GR: without accessing to the Google resources, our models can just approach the search engine's performance. However, when using the Position feature, our best model outperforms the MAP of GR by 2.30 and 1.66 absolute percent points on the development and test sets, respectively. The RBF kernel on the Position feature produces a larger improvement as it can more effectively express higher similarity values when the positions of questions are close. This cannot be done with a linear kernel.

To better study the results above, Table 5 reports the combinations between the kernel function (Linear or RBF) and the representation of the Position feature (the position itself or its inverse). While all combinations improve above the baseline, there is no clear indication on the best choice between pos or $\text{pos}^{-1}$. However, the use of the RBF kernel results in the highest performance.

Our best configuration obtains a MAP of 76.47, which is not statistically different from the best system of the competition, i.e., UH-PRHLT [6]. Still the latter makes heavy use of knowledge bases, such as BabelNet and FrameNet.

# 6. CONCLUSIONS

Establishing question–question similarity is a key component of real-world cQA systems. In this paper, we showed that the combination of similarity features, syntactic structures based on tree kernels and features based on the ranking of search engines is able to boost the reranking performance on a real-world cQA dataset. In particular, our results suggest that Google uses general models that can be on par with specific models trained on specific domains. However, if we also use advanced syntactic/semantic representations for modeling the structural relations between questions, we can achieve higher results. In particular, for the first time, we modeled and tested relational tree kernels for cQA, which are robust to noise and can thus boost Google's ranking.

In the future, we would like to better structure the representation of the questions. Indeed, as mentioned before, there are several different sections of the question text, e.g., subquestions, subject, elaborations. These could be used to improve our shallow representation, which, at the moment, merges all the question trees in a flat macro-tree.

## Acknowledgments

This work is part of the QCRI, HKBU project, Interactive sYstems for Answer Search (Iyas).

## References

[1] L. Allison and T. Dix. A bit-string longest-common-subsequence algorithm. *Inf. Process. Lett.*, 23(6):305–310, Dec. 1986.

[2] A. Barrón-Cedeño, G. Da San Martino, S. Joty, A. Moschitti, F. Al-Obaidli, S. Romeo, K. Tymoshenko, and A. Uva. ConvKN at SemEval-2016 Task 3: Answer and question selection for question answering on arabic and english fora. In *Proceedings of SemEval '16*, pages 896–903, San Diego, California, June 2016. ACL.

[3] X. Cao, G. Cong, B. Cui, C. S. Jensen, and C. Zhang. The use of categorization information in language models for question retrieval. In *CIKM*, pages 265–274, 2009.

[4] H. Duan, Y. Cao, C.-Y. Lin, and Y. Yu. Searching questions by identifying question topic and question focus. In *ACL*, pages 156–164, 2008.

[5] S. Filice, D. Croce, A. Moschitti, and R. Basili. Kelp at semeval-2016 task 3: Learning semantic relations between questions and answers. In *Proceedings of SemEval '16*, pages 1116–1123, San Diego, California, June 2016. ACL.

[6] M. Franco-Salvador, S. Kar, T. Solorio, and P. Rosso. UH-PRHLT at SemEval-2016 Task 3: Combining lexical and semantic-based features for community question answering. In *Proceedings of SemEval '16*, pages 814–821, San Diego, California, June 2016. ACL.

[7] T. Joachims. Optimizing search engines using clickthrough data. KDD, pages 133–142, 2002.

[8] C. Lyon, J. Malcolm, and B. Dickerson. Detecting short passages of similar text in large document collections. EMNLP, pages 118–125, 2001.

[9] A. Moschitti. Efficient Convolution Kernels for Dependency and Constituent Syntactic Trees. In *ECML*, pages 318–329. 2006.

[10] P. Nakov, L. Màrquez, A. Moschitti, W. Magdy, H. Mubarak, A. A. Freihat, J. Glass, and B. Randeree. SemEval-2016 task 3: Community question answering. In *Proceedings of SemEval '16*. ACL, 2016.

[11] A. Severyn and A. Moschitti. Structural relationships for large-scale learning of answer re-ranking. SIGIR, pages 741–750, 2012.

[12] K. Tymoshenko and A. Moschitti. Assessing the impact of syntactic and semantic structures for answer passages reranking. In *Proceedings of CIKM '15*, pages 1451–1460, New York, NY, USA, 2015. ACM.

[13] K. Wang, Z. Ming, and T.-S. Chua. A syntactic tree matching approach to finding similar questions in community-based qa services. In *SIGIR*, pages 187–194, 2009.

[14] M. Wise. Yap3: Improved detection of similarities in computer program and other texts. In *SIGCSE*, pages 130–134, 1996.

[15] G. Zhou, L. Cai, J. Zhao, and K. Liu. Phrase-based translation model for question retrieval in community question answer archives. In *ACL*, pages 653–662, 2011.