The Use of Dependency Relation Graph to Enhance the Term Weighting in Question Retrieval

Weinan Zhang[1*] *Zhao-yan Ming*[2†] *Yu Zhang*[1]
Liqiang Nie[2] *Ting Liu*[1] *Tat-Seng Chua*[2]
(1) School of Computer Science and Technology, Harbin Institute of Technology, Harbin, China
(2) School of Computing, National University of Singapore, Singapore
{wnzhang,zhangyu,tliu}@ir.hit.edu.cn
{mingzhaoyan,nieliq,dcscts}@nus.edu.sg

ABSTRACT
With the emergence of community-based question answering (cQA) services, question retrieval has become an integral part of information and knowledge acquisition. Though existing information retrieval (IR) technologies have been found to be successful for document retrieval, they are less effective for question retrieval due to the inherent characteristics of questions, which have shorter texts. One of the major common drawbacks for the term weighting-based question retrieval models is that they overlook the relations between term pairs when computing their weights. To tackle this problem, we propose a novel term weighting scheme by incorporating the dependency relation cues between term pairs. Given a question, we first construct a dependency graph and compute the relation strength between each term pairs. Next, based on the dependency relation scores, we refine the initial term weights estimated by conventional term weighting approaches. We demonstrate that the proposed term weighting scheme can be seamlessly integrated with popular question retrieval models. Comprehensive experiments well validate our proposed scheme and show that it achieves promising performance as compared to the state-of-the-art methods.

TITLE AND ABSTRACT IN ANOTHER LANGUAGE (CHINESE)

利用依存句法关系图改进问句检索中的词项赋权

随着社区型问答服务的出现，问句检索成为了信息及知识获取的重要途径。尽管已有的信息检索模型在文档检索方面取得成功，但是由于问句检索的短文本特性，使得已有检索模型很难适用于问句检索。对于已有的基于词项赋权的问句检索模型而言，一个主要的问题是在计算词项权重时忽略了词项之间的关系。为了解决这个问题，我们提出了一种新的利用词项间的依存句法关系作为线索的词项赋权机制。对于给定问句，我们首先构建依存关系图来计算每个词项对的关联强度，进而我们根据依存关联度来更新常规的词项权重。我们验证了所提出的词项赋权机制能够有效地整合到现有的问句检索模型中，且实验结果相比于当前最优问句检索模型有了较大提升。

KEYWORDS: cQA, Question Retrieval, Dependency Relations, Term Weighting.

KEYWORDS IN CHINESE: 社区型问答，问句检索，依存句法关系，词项赋权.

[*] This work was done when the first author was an intern in National University of Singapore.
[†] Corresponding author

Proceedings of COLING 2012: Technical Papers, pages 3105–3120,
COLING 2012, Mumbai, December 2012.

1 Introduction

With the proliferation and growth of Web 2.0, cQA services, such as Yahoo! Answers[1], Quora[2] and WikiAnswer[3], have emerged as extremely popular alternatives to acquire information online. They permit information seekers to post their specific questions on any topic and obtain answers provided by other participants. Meanwhile, the blooming social networking technologies quickly link the questions to the experts with first hand experiences and propagate well-answered questions among public who also have similar or relevant questions. Over times, a tremendous number of high quality QA pairs devoted by human intelligence has been accumulated as comprehensive knowledge bases, which greatly facilitate general users to seek information by querying in natural languages (Park and Croft, 2010; Ming et al., 2010; Park et al., 2011). As cQA services contain large scale question and answer (Q&A) archives, they offer an invaluable information resource on the Web, to provide answers to new questions posed by the users (Jeon et al., 2005b).

However, question retrieval is not a trivial task (Wang et al., 2009) due to the following problems. First, compared to other indexed documents, the archived questions in current cQA forums are usually very short, which are hard to be matched by lexicon and statistics based approaches such as okapi BM25 (Robertson et al., 1994) model etc. Similar situation happens to twitter search which also deals with short text. It was pointed out in (Teevan et al., 2011; Kwak et al., 2010) that traditional IR technologies can not be directly applied to such applications. Second, the queries are frequently depicted in natural language form that often includes various sophisticated syntactic and semantic features; they can not be easily handled by the simple key word matching models employed by current dominant web search engines.

It is worth mentioning that there already exist several efforts dedicated to research on question match. For example, Xue et al. (Xue et al., 2008) have exploited the translation-based language model (TLM) for question retrieval in large QA database and achieved significant retrieval effectiveness. A syntactic tree kernel approach to tackling the similar question matching problem was proposed by Wang et al. (Wang et al., 2009). Cui et al. (Cui et al., 2005) have tried to measure term dependencies by using different dependency parsing relation paths between the same term pairs. However, they didn't consider how the dependency parsing relation path similarities between term pairs influence the term weights. Despite their success, literature regarding question retrieval is still relatively sparse. Most of the existing work overlook the term relations by assuming that the terms in questions are independent. However, term relations, which reflect the semantic closeness between term pairs, have potentially great impacts on term weighting tasks. Table 1 shows the searching result by TLM which is the state-of-the-art question retrieval model. Though both questions are relevant to the search query, one is ranked at the top, while the other is ranked at 31st. This example demonstrates that the ability to capture term relevance among different dependency parsing relation paths is a key problem in question retrieval.

In this paper, we propose a novel term weighting scheme by exploiting the dependency relations between term pairs, which assumes that strongly dependent terms should be assigned closer weights. Given a question, we first construct a dependency graph and compute the corresponding dependency relevance matrix. Next, based on the dependency relations, a general approach

[1]http://answers.yahoo.com/
[2]http://www.quora.com/
[3]http://wiki.answers.com/

Query	How do you **charge** a farad **capacitor**?	Rank
Correct Position	How do you **charge** a 1 farad **capacitor**?	1
Wrong Position	5 farad **capacitor** for my audio system.. how to **charge** / install?.	31

Table 1: An example of question retrieval result which shows the relevant questions in both correct and wrong ranking position.

is employed to recover the "true weights" from the initial "basic" ones estimated using the traditional methods, such as maximum-likelihood (Xue et al., 2008). Finally, we integrate our term weighting scheme with classic IR model and the state-of-the-art TLM for question retrieval.

The contributions of this work are two-fold:

- First, we propose a novel term weighting scheme that models the closeness in term weights between word pairs in a sentence based on its overall grammatical dependency graph. To the best of our knowledge, this is the first work that tries to enhance term weighting based on dependency relation.
- Second, we seamlessly integrate the novel dependency graph based term weights as an orthogonal factor into the state-of-the-art retrieval models, and produce promising results on real-world data.

The remainder of this paper is organized as follows. Section 2 introduces our term weighting scheme. Sections 3 and 4 present the improved question retrieval model and our experimental results, respectively. Related works are briefly reviewed in Section 5, followed by the conclusions and future work in the last Section.

2 Proposed Term Weighting Scheme

As a key component in question retrieval models, we will first introduce our proposed term weighting scheme based on dependency relation modeling, before we proceed to integrate the model into a unified question retrieval.

2.1 Dependency Relation Detection

As mentioned earlier, dependency relations in the grammatical sense may exist between term pairs and may have certain effects on quantifying term importance. To further study the dependency strength given a question, we first perform dependency parsing utilizing the popular Stanford parser tool (de Marneffe et al., 2006). An illustrating example of parsing result for the question "How do you charge a farad capacitor?" is shown in Figure 1(a). The labels in red font represent dependency relations between term pairs. We note that dependency relations only exist between two terms which are syntactic related. It is also observed that the result of dependency parsing for a sentence is usually represented as a tree. We next remove the pseudo root node from the generated tree and ignore the directions of arcs as well as the labels, we then obtain the undirected dependency graph $G = (V, E)$, for $V = w_1, w_2, ..., w_n$, $E = e_1, e_2, ..., e_m$. where w_i represents the term in query, and e_j represents the undirect relation between terms.

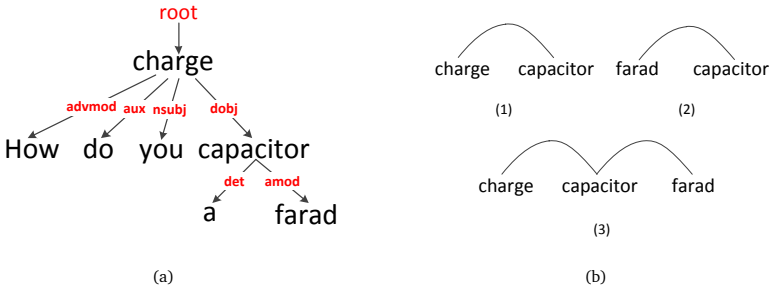


Figure 1: Illustration of the dependency relations for the question "How do you charge a farad capacitor?": (a) dependency parsing tree and (b) dependency relation path.

The undirected graph ensures that every term pair in a given question has a dependency relation path, with shorter paths reflect stronger relations. Figure 1(b) shows the representative length of the dependency relation paths from dependency relation graph. From Figure 1(b), it is obvious that the term "charge" is the direct neighbour of term "capacitor", hence, the dependency relation path length dr_path_len equals to 1 as shown in Figure 1(b) (1). However, "charge" is a bit farther away from the term "farad" as the dr_path_len between term "charge" and "farad" equals to 2 as shown in Figure 1(b) (3). This implies that "charge" should be weighted more closely with "capacitor" than with "farad".

2.2 Dependency based Closeness Estimation for Pairwise Terms

Several existing methods can be employed to compute the closeness between pairwise terms, such as pointwise mutual information (pmi), Chi and mutual information (Gao et al., 2004; Terra and Clarke, 2003). However, few of them take the syntactic dependency into consideration. Instead our approach estimates the dependency relevance of term pairs by linearly integrating multi-faceted cues, i.e., dependency relation path analysis as well as probabilistic analysis.

First, from the perspective of dependency relation path, we denote $dr\_path\_len(t_i, t_j)$ as the length of dependency relation path between term t_i and t_j. The dependency relevance can be defined as:

$$Dep(t_i, t_j) = \frac{1}{b^{dr\_path\_len(t_i,t_j)}} \quad (1)$$

where b is a constant larger than 1, which is selected based on a development set comprising 28 questions, which are randomly sampled from our querying collection. We tune b to the value that optimize the MAP. We name this metric as term dependency metric.

Second, we perform statistical analysis to capture the closeness of term pairs by pmi (Terra and Clarke, 2003) which directly capture the statistical relevance or independence between two terms, share many characteristics as mutual information. It can be formally formulated as:

$$Close_{pmi}(t_i, t_j) = \log \frac{p(t_i, t_j)}{p(t_i)p(t_j)} \quad (2)$$

where $p(t_i, t_j) = \frac{N_d(t_i,t_j)}{N_D}$ represents the probability of co-occurrence between terms t_i and t_j. $p(t_i) = \frac{N_d(t_i)}{N_D}$ and $p(t_j) = \frac{N_d(t_j)}{N_D}$ are respectively the probability of t_i and t_j occur in the whole data collection, where $N_d(t_i, t_j)$ represents the number of documents that contain both t_i and t_j. N_D represents the total number of documents. Meanwhile, $N_d(t_i)$ and $N_d(t_j)$ represent the number of documents that contain terms t_i and t_j respectively. This metric is referred to as term closeness metric.

Finally, we linearly combine the above two metrics by metric combination as:

$$w_{rel(i,j)} = \lambda Dep(t_i, t_j) + (1 - \lambda) Close_{pmi}(t_i, t_j) \tag{3}$$

where λ is a trade-off parameter.

2.3 Reallocation of Relation-based Term Weights

In the above section, we introduce the dependency relevance between two terms. Through capturing the strength of relevance, we expect to optimize the term weights. In this section, we will introduce the method which we use to reallocate term weights using dependency relevance.

First, for a given question q, we compute the dependency relevance among terms in q. If there are n terms in question q, we can construct a $n \times n$ matrix M, which we call the dependency relevance matrix. The element in M, m_{ij}, represents the dependency relevance between terms t_i and t_j computed using Equation 3. Note that as the dependency relevance graph is undirected, M is a symmetric matrix.

Second, we use orthogonal transformation to transform matrix M into a random matrix E so that we can ensure that there must be an analytical solution for the equations of which coefficient matrix equals E, where each elements in E is in range $[0, 1)$, and the sum of elements in the same row equals to 1. Hence, E has an eigenvalue that equals to 1. The solution vector of E exists and the vector with eigenvalue 1 corresponds to the solution vector. In addition, $E = D^{-1}M$, where D^{-1} is the orthogonal matrix which is used to transform the matrix M to E. Moreover, when matrix E and B are written as $E = D^{-\frac{1}{2}}BD^{\frac{1}{2}}$ and $B = D^{-\frac{1}{2}}MD^{-\frac{1}{2}}$, we can see that E is similar in structure with matrix B. Therefore, E and B will have the same eigenvalues. In fact, after solving the eigenvalues of matrix B, we get the eigenvalues of E.

Third, once we transform the dependency relevance matrix into a random matrix, we can obtain the analytical solution as term weights. However, we can also see that the analytical solution is not dependent on the initial term weight vector W_q^0. In our method, the initial term weight vector is estimated using the traditional IR models, such as VSM, BM25 and LM, and the translation-based language model (TLM) (Xue et al., 2008). Although, these models are term independent models, their term weighting schemes can also reflect the relevance between the query and documents. Hence, we linearly combine the initial term weight vector into our term weighting scheme and the analytical solution is the final optimized term weight vector W_q^*, which is derived as shown in Algorithm 1.

Term weight reassignment can be regarded as recovering the "true" weights from the initial one by using dependency relation information. The initial term weights provide a baseline for the "true" weights. Though noisy, they still reflect partial facts of the "true" weights and thus need to be preserved to some extent. Therefore, we introduce the trade-off parameter α. A small α means that the initial term weights play important role. When $\alpha = 0$, the new term weights will be the same as the initial weights.

Algorithm 1: The term weighting reallocation algorithm

Input: W_q^0, M
Output: W_q^*
Compute: $E = D^{-1}M$
Given: α
$W_q = \alpha E W_q + (1-\alpha)W_q^0$
Solution:
$W_q^* = (1-\alpha)(1-\alpha E)^{-1}W_q^0$

3 Unified Question Retrieval Model

To demonstrate that our term weighting scheme can be seamlessly integrated with the current popular question retrieval models without any underlying change, we first introduce the classic IR models and describe ways that our proposed term weighting scheme can be integrated.

3.1 Classic IR Models (VSM, BM25, LM)

The VSM model has been widely used in question retrieval. We consider a popular variation of this model, given query q, the ranking score S_{q,q^c} of the question q^c can be computed as follows:

$$S_{q,q^c} = \frac{\sum_{t \in q \bigcap q^c} w_{t,q} w_{t,q^c}}{\sqrt{\sum_t w_{t,q}^2}\sqrt{\sum_t w_{t,q^c}^2}}, \tag{4}$$
$$\text{where } w_{t,q} = \ln(1 + \frac{N}{f_t}), w_{t,q^c} = 1 + \ln(tf_{t,q^c}).$$

Here, given the query question q, S_{q,q^c} represents the ranking score of candidate question q^c. N is the number of questions in the collection, f_t is the number of questions that contain term t, and tf_{t,q^c} is the frequency of term t in q^c.

While the VSM model favors short questions, the BM25 model takes into account the question length to overcome this problem. Given a query q, the ranking score S_{q,q^c} of the question q^c can be computed as follows:

$$S_{q,q^c} = \sum_{t \in q \bigcap q^c} w_{t,q} w_{t,q^c}, \tag{5}$$
$$\text{where } w_{t,q} = \ln(\frac{N + f_t + 0.5}{f_t + 0.5}),$$
$$w_{t,q^c} = \frac{(k+1)tf_{t,q^c}}{k(1-\mathbb{b}) + \mathbb{b}\frac{W_{q^c}}{W_A} + tf_{t,q^c}}.$$

Here, k and $\mathbb{b}$ are two empirical parameters. W_{q^c} is the question length of q^c and W_A is the average question length in the whole question set.

The LM model is widely used in information retrieval, and also in question retrieval. The basic idea of the LM model is to estimate a language model for each question, and then rank questions by the likelihood of the query according to the estimated model for questions. Here, we use

Dirichlet smoothing for LM model. Given a query q, the ranking score S_{q,q^c} of the question q^c can be computed as follows:

$$\begin{aligned} S_{q,q^c} &= \prod_{t \in q} P(t|q^c) \\ &= \sum_{t \in q} P(t|M_q) \times \log P(t|M_{q^c}) \\ &= \sum_{t \in q \bigcap q^c} w_{t,q} w_{t,q^c}, \\ &\text{where } P(t|M_q) = tf_{t,q^c}, \\ P(t|M_{q^c}) &= \frac{|q^c|}{|q^c|+\delta} \times \frac{tf_{t,q^c}}{|q^c|} + \frac{\delta}{|q^c|+\delta} \times \frac{tf_{t,C}}{|C|} \end{aligned} \tag{6}$$

Here, C is the collection which contains about 20 millions question and answer pairs. $tf_{t,C}$ is the frequency of term t in C and δ is a smoothing parameter. Dirichlet smoothing is used in language model.

Integrating New Term Weights with Classic IR Models

From the aforementioned classic IR models, we find that they can be generalized to the following format:

$$S_{q,q^c} = \sum_{t \in q \bigcap q^c} w^0_{t,q} w_{t,q^c} \tag{7}$$

where t is term in question query q and $w^0_{t,q}$ represents the weight of term t in q. Note that language model can be transformed into the general form by logarithmic transformation. To replace the original term weights into dependency relevance term weight, we derive the updated form of IR models as follows:

$$S_{q,q^c} = \sum_{t \in q \bigcap q^c} w^{dr}_{t,q} w_{t,q^c} \tag{8}$$

where $w^{dr}_{t,q}$ is the updated weights explored by our proposed term weighting scheme.

3.2 Translation-based Language Model (TLM)

The TLM model, which is the state-of-the-art model in question retrieval, can be formally stated as:

$$p(q|q^c) = \prod_{w \in q} p(w|q^c) \tag{9}$$

where $p(w|q^c)$ is written as:

$$p(w|q^c) = \beta p_{ml}(w|q^c) + (1-\beta) \sum_{t \in q^c} p(w|t) p(t|q^c) \tag{10}$$

Here, given query question q, q^c indicates the candidate question for retrieval. $p(w|q^c)$ and $p(w|t)$ denote the language model and translation model, respectively; and β is the parameter to balance the two models.

Integrating New Term Weights with TLM

It is worth emphasizing that $p_{ml}(w|q^c)$ is the term weighting component in TLM. We further accomplish the unified question retrieval by simply replacing it with our term weighting scheme, and restate it as:

$$p(w|q^c) = \gamma p_{dr}(w|q^c) + (1-\gamma)\sum_{t \in q^c} p(w|t)p(t|q^c) \quad (11)$$

where dr indicates the dependency relevance.

4 Experiments

4.1 Experimental Settings

We collected a large real-world data set from Yahoo! Answers, that contains $1,123,034$ questions as our searching corpora, covering a wide range of topics, including health, internet, etc. From this dataset, we randomly selected 140 questions as our searching queries and 28 as the development set to tune all the involved parameters.

To obtain the relevance ground truth of each question query, we pool the top 20 results from various methods, such as vector space model, okapi BM25 model, language model and our proposed methods. We then asked two annotators, who are not involved in the design of the proposed methods, to independently annotate whether the candidate question is relevant (score 1) with the query question or not (score 0). When conflicts occur, a third annotator was involved in making the final decision.

For evaluation, we use precision at position n ($p@n$)($n = 1, 5, 10$), mean average precision (MAP) and mean reciprocal rank (MRR).

4.2 On Performance Comparison

We evaluate the effectiveness of our proposed term weighting scheme, with several question retrieval approaches.

First, we introduce several baselines, which includes three classic IR models, namely the Vector Space Model (VSM), okapi BM25 model (BM25) and Language model (LM).

- **VSM**: The Vector Space Model is used for question retrieval as baseline-1.
- **BM25**: The okapi BM25 model is used for question retrieval as baseline-2.
- **LM**: The language model based IR model is used for question retrieval as baseline-3.

The reason we use the classic IR models as baselines is that the classical IR models are easy to develop and tractable to operate. They capture the evidences from the whole corpus and perform well in tradition IR (Robertson et al., 1994) and cQA question and answer retrieval (Jeon et al., 2005b) tasks. The parameters in the above methods are tuned using development queries. The smoothing parameter δ of language model is set to 600; the k in BM25 model is set to 1.2 and $\mathbb{b}$ is set to 0.75 by following (Robertson et al., 1994).

Correspondingly, we derive three models which are integrated with our dependency relevance term weighting scheme, namely, **drVSM**, **drBM25** and **drLM**, respectively.

	p@1	p@5	p@10	MAP	MRR
VSM	0.1714	0.1691	0.1297	0.1980	0.1598
%chg	+18.4%	+18.2%	+17.5%	+4.5%	+16.9%
drVSM	**0.2029**	**0.1999***	**0.1523***	**0.2069***	**0.1868***
BM25	0.1857	0.1866	0.1418	0.2133	0.1716
%chg	+15.4%	+14.2%	+15.1%	+3.5%	+15.9%
drBM25	**0.2143**	**0.2131***	**0.1632***	**0.2208***	**0.1989***
LM	0.2071	0.2064	0.1603	0.2635	0.1929
%chg	+13.6%	+13.1%	+13.4%	+3.0%	+10.1%
drLM	**0.2353**	**0.2334***	**0.1818***	**0.2714***	**0.2124***

Table 2: Experiment results of classic IR models and the corresponding enhanced models that are integrated with the proposed dependency relevance term weights. $*$ indicates statistical significance over the respective baselines at 0.95 confidence interval using the t-test. %chg denotes the performance improvement in percent of dependency relevance based term weighting scheme enhanced model over the corresponding baseline.

Table 2 summarizes the experimental results in the five evaluation metrics using the three retrieval models and the three baselines. We can observe that the performances of all the three retrieval models are enhanced by dependency relevance-based term weighting scheme. Meanwhile, they obtain significant improvements over their baselines respectively. It indicates that, on the one hand, the proposed term weighting scheme is effective in question retrieval task. On the other hand, the proposed term weighting scheme provides orthogonal information about term weights when combined with the three classic IR models.

Through the experimental results, we can also see that the three classic IR models benefit differently from the dependency graph-based term weights. We conjecture the reason may be that, first, the LM has the collection smoothing scheme, so the original term weighting scheme is more rational than VSM and BM25. Second, the BM25 term weighting scheme considers question length feature so that it is unbiased in questions with different length, while the VSM favors short questions. Therefore, VSM benefits most from the term weighting scheme, followed by BM25 and LM.

Next, we introduce the state-of-the-art dependency relation-based question retrieval models as follow.

- **PRM**: Dependency relation-based passage retrieval model (PRM), which is proposed by Cui et al. (Cui et al., 2005) for question retrieval as baseline-4.

We use PRM[4] as baseline to check that whether our method is more effective than the previous dependency based IR model in question retrieval.

Next, we introduce the state-of-the-art question retrieval model TLM as another baseline, as we use two metrics to capture the term relevances, we introduce tcmTLM and tdmTLM to check the performances of each metric in question retrieval. Finally, we combine the above two metrics in

[4]We ran PRM under the setting of baseline-5 as described in (Cui et al., 2005)

form of Equation 3 and get the dependency relevance-based question retrieval model (**drTLM**). We describe the above four models as follows.

- **TLM**: Translation-based language model (TLM) which is proposed by Xue et al. (Xue et al., 2008), we implement it as baseline-5.
- **tcmTLM**: TLM integrated with our term weighting scheme where only term closeness metric is utilized to estimate the term relations.
- **tdmTLM**: TLM integrated with our term weighting scheme where only term dependency metric is utilized to estimate the term relations.
- **drTLM**: TLM integrated with our term weighting scheme where both term closeness metric and term dependency metric are combined to estimate the term relations.

For each method mentioned above, the involved parameters are carefully tuned, and the parameters with the best performances are used to report the final comparison results.

	p@1	p@5	p@10	MAP	MRR
PRM	0.2429	0.2397	0.1974	0.3595	0.2174
%chg	+14.1%	+10.6%	+7.4%	+16.0%	+18.8%
TLM	0.1928	0.1976	0.1759	0.2889	0.1889
%chg	+37.5%	+34.2%	+20.5%	+44.3%	+36.7%
tcmTLM	0.2084	0.2036	0.1903	0.3123	0.2145
%chg	+33.0%	+30.2%	+11.4%	+33.5%	+20.4%
tdmTLM	0.2675	0.2590	0.2086	0.4014	0.2495
%chg	+3.6%	+2.4%	+1.6%	+3.9%	+3.5%
drTLM	$\mathbf{0.2771}^{*}_{\dagger}$	$\mathbf{0.2651}^{*}_{\dagger}$	$\mathbf{0.2120}^{*}_{\dagger}$	$\mathbf{0.4170}^{*}_{\dagger}$	$\mathbf{0.2583}^{*}_{\dagger}$

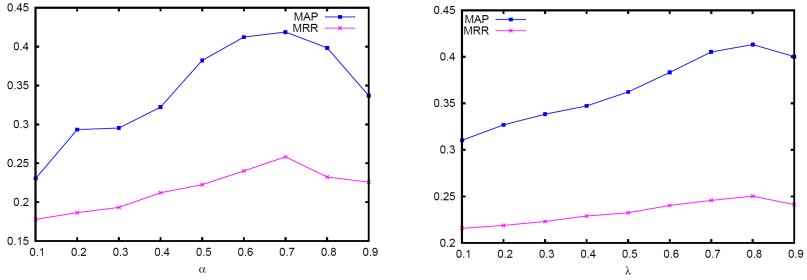
Table 3: Performance comparison among different question retrieval methods. $*$ and † indicate that the statistical significance over baseline and tcmTLM respectively is distributed within 0.95 confidence interval using the t-test. %chg denotes the boosted performance by **drTLM** in percentage. The results of our method are in bold font.

It can be observed that the performance of TLM can be enhanced by our term weighting scheme. This is due to the fact that our term weighting scheme captures the relations between term pairs and get better term weighting allocation. Compared with the existing dependency relation-based question answering passage retrieval model PRM, our method outperforms PRM in the above five evaluation methods. In particular, we only use dependency path length as a bridge to capture the relevance between two terms, which is a simple, stable and efficient way to use deep parsing, and get better performance. From this table, we can also observe that TLM integrated with tdm outperforms itself integrated with tcm. This is because tdm characterize more intrinsic dependency relations rather than the simple co-occurrences captured by tcm. Furthermore, the TLM incorporated with both tcm and tdm achieves the best performance. It worth noting that the performance of TLM is lower than that in the original paper, it is because that we use different data set and the answer evidence is not considered here.

4.3 On Parameter Sensitivity

In this section, we present how the parameters influence on question retrieval performance. In our experiments, grid search is performed to obtain the optimal values for parameters on the development set under the results of **drTLM** model.

As discussed before, smaller α in Algorithm 1 means initial term weighting scores dominate the term reassignment task, and ignore the dependency relations at all when α trends to zero. While a larger α means that our term weighting scheme will play a major role. The curve of MAP and MRR with different α value is presented in Figure 2(a) with other parameters fixed. We can see that the MAP and MRR increase with α growing and arrive at the peak when $\alpha = 0.7$ on our real dataset (development set); the performance then decrease sharply after that.



(a) (b)

Figure 2: The performance of **drTLM** with different α (a) and λ (b), when other parameters are fixed.

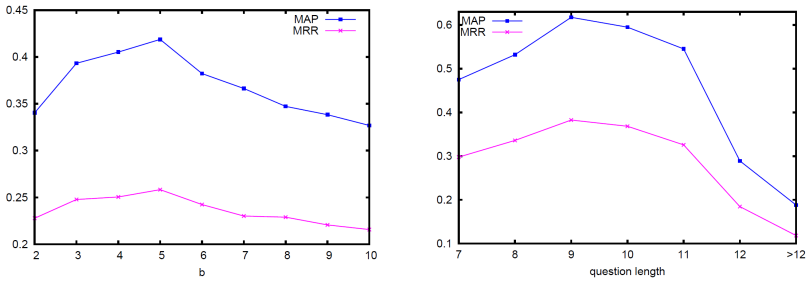
From Figure 2(a), we can infer that in our term weighting scheme, the improvement is mainly depended on term dependency relevance weighting scheme, meanwhile, the original term weights can somewhat influence the performance.

In our term weighting scheme, we introduce two relevance metrics between term pairs for term weighting reallocation as represents in Equation 3, the parameter λ is used to balance the two relevance metrics. Figure 2(b) shows the variation of MAP and MRR when λ is changing from 0.1 to 0.9.

From Figure 2(b), we can see that the MAP and MRR increase with λ growing and arrive at peak when $\lambda = 0.8$. It indicates that comparing with term closeness metric, the term dependency metric play the dominant role in the proposed term weighting scheme. Furthermore, it also illustrates that the term dependency relevance metric can effectively capture the strength of relations between two terms and influence the reallocation of term weights.

To further check the influence of parameter b in Equation 1 on the performance of question retrieval, Figure 3(a) presents the curve of MAP and MRR with different b value. From Figure 3(a), we can see that the performance arrive at the peak when $b = 5$.

In addition, we also consider the influence of question length, which indicates the number of words in one question, on the performance of question retrieval. Figure 3(b) shows the curve



(a) (b)

Figure 3: The performance of **drTLM** with different b (a) and question length (b), when other parameters are fixed.

of MAP and MRR with different question length. Through Figure 3(b), we actually check the ability of our proposed method on handling queries in different length as well as in different complexities. From Figure 3(b), we can see that the proposed method can adapt queries in wide range of length, which is from 7 to 11, and get well performance on question retrieval. Meanwhile, we also see that in question retrieval, neither shorter nor longer queries, can get better performance. It also reveals that natural language question queries can better represent users' searching intent than key words queries as they contain plentiful lexical information, as well they may introduce more noise. The parameter γ in Equation 11 equals 0.8, which also illustrates that in the dependency relevance-based TLM (**drTLM**) model, the proposed term weighting scheme contributes more in question retrieval. The improvement of searching results mainly depend on the reallocated term weights.

4.4 On Efficiency Analysis

For our proposed approach, we can see that the computational cost mainly comes from two parts: (1) question dependency parsing; (2) graph-based term weighting. Assume the question length is n, it can be analyzed that the computational cost scales as $O(n^3)$. In our data collection, n is averaged as 10.8, which leads to very low computational cost. In our experiments, we compare the time of process for each search round between the proposed **drTLM** model and PRM model which is also a dependency relation-based model that we use as baseline-5. The average search rounds that **drTLM** complete at one second is 14, while PRM is 17 (with a pc of 72G memory and Intel(R) Xeon(R) CPU E5620@2.40GHz). It means that the above two methods are comparable in efficiency. However, our proposed method doesn't need to training models, which leads to more efficient. Meanwhile, for further efficiency, we can also use iterative methods to get the numerical solution instead of analytical solution in graph-based term weighting.

4.5 Case Study

Table 4 representatively illustrates the top 5 search results for the query "How do you charge a farad capacitor?" by TLM and **drTLM** which is our dependency graph enhanced TLM. Clearly, our proposed model returns more relevant questions at top positions, mainly due to the adjusted weights for term "charge", "farad", and "capacitor".

Rank No.	TLM	**drTLM**
1	**How do you charge a 1 farad capacitor**?	**How to charge a farad capacitor**?
2	**How do you charge a 5 farad capacitor**?	**How do you charge a 1 farad capacitor**?
3	What resistor do you use to charge a 1 farad capacitor?	**How do you charge a 5 farad capacitor**?
4	How do you install a farad amp capacitor?	How do you install a 1 farad capacitor?
5	How do you hook up a 3 farad capacitor to two amps?	**5 farad capacitor for my audio system.. how to charge / install**?

Table 4: Search results comparison between TLM and **drTLM** for query "How do you charge a farad capacitor?".Questions in bold font are relevant ones.

5 Related Work

The existing IR technologies are frequently based on Bag-of-Words models and regard both the query and documents in collections as composition of individual and independent words. For example, Ponte et al. (Ponte and Croft, 1998) utilized unigram language model for information retrieval. Jones et al. (Jones et al., 2000) proposed the binary independent retrieval (BIR) model to capture the relevance between queries and documents. Duan et al. (Duan et al., 2008) proposed a new language model to capture the relation between question topic and focus. They may not be directly applicable in the question retrieval domain due to at least two reasons. First, compared to the simple keywords based search, the querying questions are usually represented in natural language and depict some concepts linked by intrinsic semantic relationships. Second, the to be searched documents are also questions, which are far shorter than the verbose documents in traditional search approaches.

Jeon et al. (Jeon et al., 2005a,b), moving forward one step, provided comparison of four different retrieval models, i.e., vector space model, okapi, language model and translation model for question retrieval in archived cQA data, experimental results revealed that the translation model outperforms the other models. Later, Xue et al. (Xue et al., 2008) combined the language model and translation model to a translation-based language model and observed better performance in question retrieval. Following that, Ming et al. (Ming et al., 2010) utilized three domain specific metrics to explore term weights and integrated them into existing IR models. However, most of these term weighting based retrieval models ignore the dependency relations between term pairs.

Researchers never stop to capture the term dependencies for IR models. For instance, (Song and Croft, 1999; Srikanth and Srihari, 2002) replaced the unigram to bigram and bi-term in language model. Gao et al. (Gao et al., 2004) proposed a dependency language model

to capture term dependencies through dependency parsing relations. Park et al. (Park and Croft, 2010) explore dependency features for term ranking in verbose query. Moreover, they proposed a quasi-synchronous IR model (Park et al., 2011) to integrate dependency information. Cui et al. (Cui et al., 2005) have tried to measure the terms dependencies by using different dependency parsing relation paths between same term pairs. Sun et al. (Sun et al., 2006, 2005) explored dependency relations for query expansion and answer extraction in question passage retrieval and answering retrieval. However, they only estimated term dependencies or syntactics between adjacent term and overlooked the nonadjacent cases. To tackle this issue, in this paper, we proposed a term weighting approach by incorporating global dependency relevance.

Conclusions

In this paper, we explored the dependency relations between question terms to enhance the question retrieval in cQA. Given a question, we first automatically constructed a dependency graph, and then estimated the relation strength between vertex pairs. Based on the quantified dependency relations, we proposed a novel term weighting scheme to refine the initial term weights estimated by traditional technologies. Further, we demonstrated that our term weighting approach can be unified with the state-of-the-art question retrieval models. By conducting experiments on real-world data, we demonstrated that our proposed scheme yields significant gains in retrieval effectiveness.

This work begins a new research direction for weighting question terms by incorporating dependency relation cues. In future work, we will further study the dependency relation based term weights by differentiating the importance of relation types and assigning relation-aware weights.

Acknowledgments

This work was supported by the Natural Science Foundation of China (Grant No. 61073129, 61073126), 863 State Key Project (Grant No. 2011AA01A207) and partially supported by NExT Search Centre, which is supported by the Singapore National Research Foundation & Interactive Digital Media R&D Program Office, MDA under research grant (WBS:R-252-300-001-490).

References

Cui, H., Sun, R., Li, K., Kan, M.-Y., and Chua, T.-S. (2005). Question answering passage retrieval using dependency relations. In *Proceedings 28th annual international ACM SIGIR conference*, pages 400–407.

de Marneffe, M.-C., MacCartney, B., and Manning, C. D. (2006). Generating typed dependency parses from phrase structure trees. In *LREC*.

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

Gao, J., Nie, J.-Y., Wu, G., and Cao, G. (2004). Dependence language model for information retrieval. In *SIGIR*, pages 170–177.

Jeon, J., Croft, W. B., and Lee, J. H. (2005a). Finding semantically similar questions based on their answers. In *SIGIR*, pages 617–618.

Jeon, J., Croft, W. B., and Lee, J. H. (2005b). Finding similar questions in large question and answer archives. In *CIKM*, pages 84–90.

Jones, K. S., Walker, S., and Robertson, S. E. (2000). A probabilistic model of information retrieval: development and comparative experiments. In *Information Processing and Management*, pages 779–840.

Kwak, H., Lee, C., Park, H., and Moon, S. (2010). What is twitter, a social network or a news media? In *WWW'10: Proceedings of the 19th International World Wide Web Conference*.

Ming, Z., Chua, T.-S., and Cong, G. (2010). Exploring domain-specific term weight in archived question search. In *CIKM*, pages 1605–1608.

Park, J. H. and Croft, W. B. (2010). Query term ranking based on dependency parsing of verbose queries. In *SIGIR*, pages 829–830.

Park, J. H., Croft, W. B., and Smith, D. A. (2011). A quasi-synchronous dependence model for information retrieval. In *Proceedings of the 20th ACM international conference on Information and knowledge management*, CIKM '11, pages 17–26.

Ponte, J. M. and Croft, W. B. (1998). A language modeling approach to information retrieval. In *SIGIR '98: Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval*, pages 275–281.

Robertson, S. E., Walker, S., Jones, S., Hancock-Beaulieu, M., and Gatford, M. (1994). Okapi at trec-3. In *TREC*.

Song, F. and Croft, W. B. (1999). A general language model for information retrieval. In *CIKM '99: Proceedings of the eighth international conference on Information and knowledge management*, pages 316–321.

Srikanth, M. and Srihari, R. K. (2002). Biterm language models for document retrieval. In *SIGIR*, pages 425–426.

Sun, R., Cui, H., Li, K., Kan, M.-Y., and Chua, T.-S. (2005). Dependency relation matching for answer selection. In *Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '05, pages 651–652.

Sun, R., Ong, C.-H., and Chua, T.-S. (2006). Mining dependency relations for query expansion in passage retrieval. In *Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '06, pages 382–389.

Teevan, J., Ramage, D., and Morris, M. R. (2011). # twittersearch : a comparison of microblog search and web search. In *WSDM*, pages 35–44.

Terra, E. and Clarke, C. L. A. (2003). Frequency estimates for statistical word similarity measures. In *NAACL '03: Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology*, pages 165–172.

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

Xue, X., Jeon, J., and Croft, W. B. (2008). Retrieval models for question and answer archives. In *SIGIR*, pages 475–482.