Exploring Domain-specific Term Weight in Archived Question Search

Zhao-Yan Ming[1,2], Tat-Seng Chua[2] and Gao Cong[3]
[1]NUS Graduate School for Integrative Sciences and Engineering
[2]School of Computing, National University of Singapore
[3]School of Computer Engineering, Nanyang Technological University
{mingzy,chuats}@comp.nus.edu.sg,gaocong@ntu.edu.sg

ABSTRACT

Community Question Answering services, e.g., Yahoo! Answers, have accumulated large archives of question answer (QA) pairs for information and answer retrieval. An effective question retrieval model is essential to increase the accessibility of the QA archives. QA archives are usually organized into categories and question search can be performed within the whole collection or within a certain category.

In this paper, we explore domain-specific term weight for archived question search. Specifically, we propose a novel light-weighted term weighting scheme that exploits multiple aspects of the domain information. We also introduce a framework to seamlessly integrate domain-specific term weight into the existing retrieval models. Extensive experiments conducted on real Archived QA data demonstrate the utility of the proposed techniques.

Categories and Subject Descriptors

H.4 [**Information Systems Applications**]: Miscellaneous; H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval

General Terms

Algorithms, Experimentation, Performance

Keywords

CQA question search, Domain-specific term weighting

1. INTRODUCTION

Community-based Question Answering (CQA) sites have become increasingly popular and they have accumulated very large archives of user generated question answer pairs that form valuable knowledge bases for information seekers. To effectively share the knowledge in Archived Question Answer repository, it is essential to develop effective question search models that are capable of returning relevant questions and answers for a user query.

The existing widely used information retrieval models, including TFIDF, Okapi BM25, and Language Model capture document-level and collection-level evidences in their term weighting schemes. Going beyond the traditional bag-of-word approaches, recently Xue et al [9] exploits semantic relations between terms using translation model and Wang et al [8] utilizes the syntactic relations between sentences.

In this paper, we aim to explore domain-level evidence of questions to complement the existing retrieval models. As a specific application of IR, Question Search in CQA repository is distinct from the search of web pages or news articles in that questions are organized into categories. This makes it possible to extract domain-specific information to enhance question search. Intuitively, a word may be more important in some domain than other domains. For example, "shutter" is an important word in domain *Consumer Electronics*, especially the sub-domain *camera*, but not in domain *Health*. To utilize domain-level evidence to enhance the existing retrieval models, first we need derive domain-level evidence, which is non-trivial. It is expected that the domain-level evidence would reflect the domain topic.

To achieve this, we compute the domain-level evidence for each term by taking into account three aspects, namely, *General Collection Based Evidence*, *Sub-domain Based Evidence*, and *Entropy Based Evidence*, as complements to measure term importance together with document level evidence and collection level evidence. We proceed to present them in the following section.

2. PROPOSED APPROACH

2.1 General Collection Based Evidence

Divergence Feature

The divergence of term distribution in a specific collection from a general collection reveals the significance of terms globally in its domain. We employ *Jensen-Shannon*(JS) divergence to capture the difference of term distribution in two collections. It is defined as the mean of the relative entropy of each distribution to the mean distribution. We examine the point-wise function for each individual term as follows:

$$d_{JS}(t, S||G) = \frac{p_s(t)\log\frac{2p_s(t)}{p_s(t)+p_g(t)} + p_g(t)\log\frac{2p_g(t)}{p_s(t)+p_g(t)}}{2} \quad (1)$$

where S and G denote the domain-specific and general vocabularies, with their probability distributions $p_s(t_i)$ and

CIKM'10, October 26–29, 2010, Toronto, Ontario, Canada.
Copyright 2010 ACM 978-1-4503-0099-5/10/10 ...$10.00.

$p_g(t_i)$ obtained by the Maximum Likelihood Estimator on the specific and the general vocabularies, respectively.

Estimating Term Saliency from Divergence Feature We define a mapping function $f_n : d_{JS} \rightarrow \omega_{s_1}$, which produces as output an estimation ω_{s_1} (denotes the term saliency score generated by aspect 1 general collection based evidence) given d_{JS} as input. This function has a normalization effect on the raw score of aspect 1. We propose a heuristic evaluation function based on logistic function $L(x)$ as $f_n(x) = 1 + \tau L(x + \alpha)$. Therefore the final form of the aspect 1 term saliency score ω_{s_1} is:

$$\omega_{s_1} = 1 + \tau \frac{1}{1 + e^{-(d_{JS}+\alpha)}} \tag{2}$$

2.2 Sub-domain Based Evidence

Term distribution in a domain is likely to vary from one sub-domain to another. To capture the specificity of terms with regard to a sub-domain, we propose to measure the specificity of terms within a domain by comparing the term distribution in each sub-domain to that in the domain collection. The terms that have different distributions in a sub-domain and the whole domain would be important to characterize the sub-domain from the general domain.

Aspect 2 term saliency score ω_{s_2} is calculated the similar way as in Aspect 1. Let S denote a domain collection and S_s denote a sub-domain of S. Given $p_s(t)$ and $p_{S_s}(t)$ (the probability of t in a sub-domain S_s), we compute the difference of term distributions between domain S and subdomain S_s by invoking Equation 1 so as to get $d_{JS}(t, S||S_s)$. Similarly, we apply the Equation 2 on $d_{JS}(t, S||S_s)$ to do the saliency estimation (normalization) to obtain ω_{s_2}.

2.3 Term Entropy Based Evidence

To capture the specificity of terms with regard to all the sub-domains of a domain, we compute the entropy for terms across the sub-domains in a domain. Intuitively, a term of high entropy is more likely to occur in many sub-domains while a term of low entropy tends to occur only a few sub-domains. We value the terms with low entropy since they are more distinctive.

$$Entropy(t, S) = -\sum_{C \in \{S_s\}} p(C|t) log(p(C|t)) \tag{3}$$

where $p(C|t) = tf(C, t)/\sum_{Z \in \{S_s\}} tf(Z, t)$, and $tf(Z, t)$ is the frequency of the term t within a sub-domain Z.

Since low entropy terms are deemed as important, the term saliency score from Aspect 3 ω_{s_3} is thus defined as the inverse of the entropy:

$$\omega_{s_3} = 1/(Entropy(t, S) + \epsilon) \tag{4}$$

where ϵ is a smoothing parameter we set as 0.001.

We use a linear interpolation of the three aspects to derive the domain-level evidence $\omega_{de}(t) = \pi_1\omega_{s_1} + \pi_2\omega_{s_2} + pi3\omega_{s_3}$.

3. INTEGRATION WITH EXISTING IR MODELS

3.1 Preliminaries

Vector Space Model

The Vector Space Model has been used widely in question retrieval [5, 6]. We consider a popular variation of this model [11]: given a query $\mathbf{q}$, the ranking score $S_{\mathbf{q},\mathbf{d}}$ of the question $\mathbf{d}$ can be computed as follows:

$$S_{\mathbf{q},\mathbf{d}} = \frac{\sum_{t \in \mathbf{q} \bigcap \mathbf{d}} w_{\mathbf{q},t} w_{\mathbf{d},t}}{\sqrt{\sum_t w_{\mathbf{q},t}^2}\sqrt{\sum_t w_{\mathbf{d},t}^2}}, \text{ where}$$
$$w_{\mathbf{q},t} = \ln(1 + \frac{N}{f_t}),\ w_{\mathbf{d},t} = 1 + \ln(tf_{t,\mathbf{d}}) \tag{5}$$

Here N is the number of questions in the collection, f_t is the number of questions containing the term t, and $tf_{t,\mathbf{d}}$ is the frequency of term t in $\mathbf{d}$.

BM25 Model

While the Vector Space Model favors short questions, the Okapi BM25 Model [7] takes into account the question length to overcome this problem. The Okapi Model is used for question retrieval by Jeon et al. [5]. Given a query $\mathbf{q}$ and a question $\mathbf{d}$, the ranking score $S_{\mathbf{q},\mathbf{d}}$ is computed as follows:

$$\begin{aligned} S_{\mathbf{q},\mathbf{d}} &= \sum_{t \in \mathbf{q} \bigcap \mathbf{d}} w_{\mathbf{q},t} w_{\mathbf{d},t}, \text{ where} \\ w_{\mathbf{q},t} &= \ln(\frac{N - f_t + 0.5}{f_t + 0.5}), \\ w_{\mathbf{d},t} &= \frac{(k+1)tf_{t,\mathbf{d}}}{k(1-b) + b\frac{W_\mathbf{d}}{W_A} + tf_{t,\mathbf{d}}} \end{aligned} \tag{6}$$

Here N is the number of questions in the collection; f_t is the number of questions containing the term t; $tf_{t,\mathbf{d}}$ is the frequency of term t in $\mathbf{d}$; k, and b are set to 1.2 and 0.75, respectively, by following Robertson et al. [7] ; and $W_\mathbf{d}$ is the question length of $\mathbf{d}$ and W_A is the average question length in the collection.

Language Model

The Language Model is used in previous work [2, 3, 5] for question retrieval. The basic idea of the Language 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. We use Dirichlet smoothing [10]. Given a query $\mathbf{q}$ and a question $\mathbf{d}$, the ranking score $S_{\mathbf{q},\mathbf{d}}$ is computed as follows:

$$\begin{aligned} S_{\mathbf{q},\mathbf{d}} &= \prod_{t \in \mathbf{q}} (P(t|\mathbf{d})), \text{ where} \\ P(t|\mathbf{d}) &= \frac{|\mathbf{d}|}{|\mathbf{d}| + \lambda} \times \frac{tf_{t,\mathbf{d}}}{|\mathbf{d}|} + \frac{\lambda}{|\mathbf{d}| + \lambda} \times \frac{tf_{t,\mathbf{C}}}{|\mathbf{C}|} \end{aligned} \tag{7}$$

Here $\mathbf{C}$ is the collection and λ is the smoothing parameter.

3.2 A General Framework for Integrating Domain level Evidence

The aforementioned Bag-of-Word retrieval functions can be generalized to the following format:

$$Score(\mathbf{q}, \mathbf{d}) = \sum_{t_i \in \mathbf{q}} \omega(t_i) \tag{8}$$

where t_i is the i^{th} query term, and $\omega(\cdot)$ is a term weighting model. Generally, $\omega(\cdot)$ is a function that takes in document level and collection level evidences of a term. Note that language model can be transformed into the general form by logarithmic transformation.

To accommodate the three-aspect domain level saliency score $\omega_{de}(t_i)$, we introduce a general framework as follows:

$$Score(\mathbf{q}, \mathbf{d}) = \sum_{t_i \in \mathbf{q}} \omega(t_i) * \omega_{de}(t_i) \tag{9}$$

4. EXPERIMENTS

4.1 Data Collection:

We collect questions from two top categories, *Consumer Electronics* (*CE*) and *Heath* of Yahoo! Answers. Each category encompasses a few subcategories. We view each category as a domain and the sub-categories its sub-domains.

For the query set, we randomly select 300 questions from either domain's archive, with the remaining 864013 and 682747 questions as the searching corpora for *CE* and *Health* respectively. As in the real question search scenario, only *subject* is used in the queries. After preprocessing, we obtained 253 and 266 questions for either domain respectively. From the remaining questions, we randomly choose 230 for testing, and the others (23 and 36 questions, respectively) are used for development. Evaluation has been performed by pooling the top 20 results from various methods.

To evaluate the proposed term weighting scheme on dataset of different document lengths, we generate two versions of searching corpora: one contains the *subject* field of questions only and the average length of question is about 10 words, the other concatenates all the three fields *subject* + *content* + *best answers*, with average length of about 124 words.

4.2 Question Search Models:

To evaluate the performance of the proposed term weighting scheme, we use three sets of question search models, each set consisting of three methods. The first set is based on the Language Model (LM):

(1) LM (baseline): Language Modeling approach.

(2) LM@OpS: This model searches in the subdomain that the query was originally assigned in Yahoo! Answers. We view it as the "optimal" subdomain information.

(3) LM+ω_{de}: This model integrates the term score from LM and the proposed ω_{de} domain-level evidence.

Similarly, the other two sets of search models are based on Vector Space Model (VSM) and Okapi BM25 ($BM25$), respectively. Hence, we have another six models, VSM(the baseline of the set), $VSM@OpS$, VSM+ω_{de}; and $BM25$(the baseline of the set), $BM25@OpS$, and $BM25 + \omega_{de}$.

The parameters in the above systems are tuned using the development queries. For the proposed method in Equation 2, we fix τ to be 1.0 as it does not affect the shape of the function. α is set as 2.0 as the optimal range is $[2.0, 3.0]$, and $\pi_{1,2,3} = \frac{1}{3}$ The smoothing parameter λ of LM is set to be 600, and for Okapi BM25 $k = 1.2$, $b = 0.75$.

4.3 Experimental Results and Discussion

Table 1 summarizes the experimental results in Mean Average Precision (MAP) using the three set of retrieval models on the two types of data, *subj* and *long* on two domains *CE* and *Health*. We make the following observations:

(1) All the three ω_{de} (domain-level evidence) enhanced retrieval models achieve significant improvement over their respective baseline. This shows that the proposed term weighting scheme combines well with the existing retrieval models. Among the three set of retrieval models, VSM benefits the most, followed by BM25 and LM. The possible reason, we conjecture, would be that the smoothing effect of LM distinguishes between the salient and trivial terms in the collection of a domain, and therefore reflects the domain specificity of the terms to some extent. The *idf* component in VSM and BM25 has limited effect on reflecting the term domain specificity, and thus benefit more from a complementary domain-specific term weighting factor.

Table 1: † indicates statistical significance over the respective baselines at 0.95 confidence interval using the t-test. %chg denotes the performance improvement in percent of each domain-level evidence enhanced model over the corresponding baseline.

Retrieval Models	CE		Health	
	subj	long	subj	long
LM	0.2857	0.3085	0.3119	0.3282
LM@OpS	0.2912	0.3014	0.2826	0.3156
$LM + \omega_{de}$	$0.346^\dagger$	$0.3687^\dagger$	$0.3735^\dagger$	0.3913
%chg	21.1	19.5	19.7	19.2
VSM	0.2551	0.2708	0.2731	0.2898
VSM@OpS	0.2465	0.2662	0.2833	0.3014
$VSM + \omega_{de}$	$0.3165^\dagger$	$0.3412^\dagger$	$0.3265^\dagger$	$0.3447^\dagger$
%chg	24.0	25.9	19.5	18.9
BM25	0.2684	0.2872	0.2852	0.3016
BM25@OpS	0.2507	0.2707	0.2925	0.3135
$BM25 + \omega_{de}$	$0.3212^\dagger$	$0.3496^\dagger$	$0.3315^\dagger$	$0.3557^\dagger$
%chg	19.6	21.7	16.2	17.9

(2) The baseline models searching in the optimal subdomain do not consistently improve the respective baseline searching in the whole domain. The results show that the querys' sub-domain information does not help the retrieval performance when used directly. This may be because relevant questions are also contained in other sub-domains but not only in the sub-domain of query questions, and some of the questions are not correctly assigned to subdomains in the Yahoo! Answers. However, when applying ω_{de} to enhance retrieval models, the sub-domain of a question is implicitly obtained by the sub-domain based evidence ω_{s_2}. Sub-domains that have the higher accumulated ω_{s_2} of the terms in the question are more likely the true sub-domain of the question. This suggests that ω_{s_2} does an implicit classification of the questions searched.

(3) The proposed term weighting scheme works well on both short (only subject field) and long questions (each consisting of subject, content and best answer). The improvement on *long* is slightly lower than that on *subj* field. By comparing *subj* and *long*, we observe that longer documents have higher baseline performances. The reason might be that questions with only *subj* contain one single sentence that both the salient and trivial terms appear only once. Given the domain-specific information, the *subj* questions retrieval can be even better enhanced.

To get a better understanding of the comparing models, Table 2 gives part of the results of an example query question. As can be seen, the top 2 returned results of LM are both irrelevant though they share a large portion of words with the query. The problem is that the salient terms like "food", "coloring", and "eye" are treated the same as less important terms like "what", "would", "happen", *etc.*. Though it is possible to remove "what" and "would" using a well con-

Table 2: Search results for "What would happen if u put food coloring in your eye?".

Models	Rank	Questions	SubDomain
LM	1st	What would happen if your eye's pupil was your entire eye?	Optical
	2nd	What would happen if you got poision ivy in your eye?	Optical
	5th	**Is it dangerous to put food coloring in your eyes?**	Medicine
LM@OpS	1st	What would happen if your eye's pupil was your entire eye?	Optical
	2nd	What would happen if you got poision ivy in your eye?	Optical
	3rd	What would happen to an eye kept shut for a long period of time?	Optical
$LM + \omega_{de}$	1st	**Is it dangerous to put food coloring in your eyes?**	Medicine
	2nd	**Can i put Food Coloring in my eye?**	Other
	3rd	What would happen if you got poision ivy in your eye?	Optical

structed stopword list, the list may be hard to adapt to all cases. And terms like "happen" may be arguable.

The top 3 results using LM@OpS are quite similar to the query but do not capture the main content. It is consistent with our earlier observation that searching in the optimal subdomain is not a good way to use sub-domain information. We see that the proposed model $LM+\omega_{de}$ returns questions that share mainly content words with the query. It indicates that the proposed model recognizes the domain specificity of the terms and assigns weights properly.

5. RELATED WORK

Question retrieval has recently been investigated for CQA data. Jeon et al. [4,5] compare four different retrieval methods, i.e., the vector space model, the Okapi model, the language model, and the translation model, for question retrieval on CQA data, and the experimental results show that the translation model outperforms the other models. In subsequent work [9], they propose a translation-based language model that combines the translation model and the language model for question retrieval. Duan et al. [3] propose a solution that makes use of question structures for retrieval by building a structure tree for questions in a category of Yahoo! Answers to discover question topic and question focus. Recently, Wang et al. [8] employ a parser to build syntactic trees of questions, and questions are ranked based on the similarity between their syntactic trees and the syntactic tree of the query question.

We are also aware of the recent work by Cao et al. [1, 2] that exploits the question categories in CQA data for question retrieval. However, the proposed category-based smoothing approach [2] is tightly coupled with the langauge model and is difficult to apply to other retrieval models. Cao et al. [1] propose to compute the ranking score of a question by a linear combination of the relevance score of a query to the question and the relevance score of a query to the category containing the question. While the method [1] aims to improve question search in the whole collection, our proposed techniques aim to improve question search within a certain domain, which is offered as an option for question search by many CQA services.

6. CONCLUSIONS

In this paper, we propose a novel term weighting scheme that exploits multiple aspects of the domain information to generate domain-level evidence to enhance the existing bag-of-words information retrieval models. Extensive experiments conducted on real Archived QA data from Yahoo! Answer demonstrate that the proposed techniques significantly improve the performance of the existing retrieval models.

This work opens to several interesting directions for future work. First, it is of relevance to evaluate the performance of the proposed domain-specific term weighting scheme on longer documents like web pages. Second, it would be interesting to build a domain-specific stopword list according to the proposed methods of computing domain-level evidences for each term.

7. REFERENCES

[1] X. Cao, G. Cong, B. Cui, and C. S. Jensen. A generalized framework of exploring category information for question retrieval in community question answer archives. In *WWW*, pages 201–210, 2010.

[2] 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.

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

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

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

[6] V. Jijkoun and M. de Rijke. Retrieving answers from frequently asked questions pages on the web. In *CIKM*, pages 76–83, 2005.

[7] S. Robertson, S. Walker, S. Jones, M. Hancock-Beaulieu, and M. Gatford. Okapi at trec-3. In *TREC*, pages 109–126, 1994.

[8] 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.

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

[10] C. Zhai and J. Lafferty. A study of smoothing methods for language models applied to information retrieval. *ACM Transactions on Information Systems (TOIS)*, 22(2):179–214, 2004.

[11] J. Zobel and A. Moffat. Inverted files for text search engines. *ACM Comput. Surv.*, 38(2):6, 2006.