Recommending Questions Using the MDL-based Tree Cut Model

Yunbo Cao[1,2], Huizhong Duan[1], Chin-Yew Lin[2], Yong Yu[1], and Hsiao-Wuen Hon[2]

[1]Shanghai Jiao Tong University,
Shanghai, China, 200240
{summer, yyu}@apex.sjtu.edu.cn

[2]Microsoft Research Asia,
Beijing, China, 100080
{yunbo.cao, cyl, hon}@microsoft.com

ABSTRACT

The paper is concerned with the problem of *question recommendation*. Specifically, given a question as query, we are to retrieve and rank other questions according to their likelihood of being good recommendations of the queried question. A good recommendation provides alternative aspects around users' interest. We tackle the problem of question recommendation in two steps: first represent questions as graphs of topic terms, and then rank recommendations on the basis of the graphs. We formalize both steps as the *tree-cutting* problems and then employ the *MDL* (Minimum Description Length) for selecting the best cuts. Experiments have been conducted with the real questions posted at Yahoo! Answers. The questions are about two domains, 'travel' and 'computers & internet'. Experimental results indicate that the use of the *MDL-based tree cut model* can significantly outperform the baseline methods of word-based VSM or phrase-based VSM. The results also show that the use of the MDL-based tree cut model is essential to our approach.

Categories and Subject Descriptors

H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval – *search process*; H.4.m [**Information Systems and Applications**]: Miscellaneous - *BSP*; I.7.m [**Document and Text Processing**]: Miscellaneous

General Terms

Algorithms, Experimentation, Human Factors

Keywords

Question Recommendation, Query Suggestion, Tree Cut Model, Minimum Description Length

1. INTRODUCTION

Community-based Q&A service (referred to as cQA) is a kind of web service where people can post questions and answer other people's questions. The growth in cQA has been one of the emerging trends at the age of Web 2.0. The typical examples provided by the commercial search engines include Yahoo! Answers[1], Live QnA[2], and Baidu Zhidao[3]. The huge number of retail and business sites that provide FAQ services can be viewed as the same type of system.

Over time, cQA services build up very large archives of previous questions and their answers. In order to avoid the lag time involved with waiting for a personal response, a cQA service will typically automatically search this archive to see if the same question has previously been asked. If the question is found, then a previous answer can be provided with very little delay. That is what we called question search. For example, given the question in the first row of Table 1, question search is to return the question in the second row, which are semantically equivalent to the input question and expected to have the same answer. Many methods have been proposed for tackling the problem [13-15].

Table 1: Question search vs. question recommendation

Queried question:
Any cool clubs in Berlin or Hamburg?
Question search:
What are the best/most fun clubs in Berlin?
Question recommendation
How far is it from Berlin to Hamburg?
Where to see between Hamburg and Berlin?
Hong long does it take to get to Hamburg from Berlin on the train?
Cheap hotel in Hamburg?

In this paper, to complement question search, we are to explore a novel application which we call *question recommendation*. We consider a question as a combination of *question topic* and *question focus*. Question topic usually presents the major context/constraint of a question (e.g., Berlin, Hamburg) which characterizes users' interest. Question focus (e.g., cool club) presents certain aspect (or descriptive features) of the question topic. When users ask questions, they usually are pretty clear about their question topics. However, they might not be aware that there exist several aspects around the question topics that are also worth exploring. Thus, it is desirable that an automatic system can suggest alternative aspects of the queried question topic and recommend the related questions

WWW 2008, April 21–25, 2008, Beijing, China.
ACM 978-1-60558-085-2/08/04.

[1] http://answers.yahoo.com

[2] http://qna.live.com

[3] http://zhidao.baidu.com

accordingly. For example, if a user wants to leave for Berlin and Hamburg, it will be very useful that the system can suggest the question "*where to see between Hamburg and Berlin*?" (or others in the third row of Table 1) although her or his question is "*any cool clubs in Berlin or Hamburg?*" To the best of our knowledge, no existing study has been presented with regard to question recommendation. Query suggestion can be considered as an analog of question recommendation in the setting of web search. However, it is not designed for long queries like questions.

We tackle the problem of question recommendation in two steps: first represent questions as graphs of topic terms, and then rank recommendation candidates on the basis of the graphs.

To represent question as graphs of topic terms, we need have a method for the *extraction of topic terms*. We consider as *topic terms* all the terms that can be used to represent the question topic or the question focus. For example, 'Hamburg', 'Berlin', and 'cool club' can be extracted as topic terms from the query in Table 1. We can make use of all the noun phrases and ngrams as the candidate topic terms. However, the number of noun phrases and ngrams is usually too large to manage and some of them are too sparse to be accurately modeled. Therefore, we propose to reduce some topic terms to their prefixes or suffixes. For example, given two topic terms 'embassy suite hotel' and 'suite hotel', we may remove 'embassy suite hotel' from the set of topic terms by considering it same as 'suite hotel'. We can call this '*reduction of topic terms*'. The *reduction* can help reduce the size of the vocabulary of topic terms and moderate the sparseness problem.

A good recommendation provides alternative aspects (question focus) around users' interest (question topic). That is to say, a good recommendation should differ from the queried question in *question focus* and stick to the queried question in *question topic*. Thus, provided that questions are represented by topic terms, the step of ranking recommendations should be able to discriminate question topic from question focus. The discrimination problem can be complex because it should be based on all the related questions as well as the queried question. For the question in Table 1, for example, the topic terms 'Hamburg' and 'Berlin' are considered more likely to be question topics than the topic term 'cool club', considering that, in the six questions in Table 1, the former two topic terms occurs much more than the latter one.

In this paper, we propose using the MDL-based (Minimum Description Length) tree cut model for handling the two issues raised above, *reduction of topic terms* and *discrimination between question topic and question focus*. MDL is a principle of data compression and statistical estimation from information theory, which enables the proposed approach to optimize balance between specificity and generality.

We empirically conduct the question recommendations with the questions about 'travel' and the questions about 'computers & internet'. Both two kinds of questions are from Yahoo! Answers. Experimental shows that our proposed approach using the MDL-based tree-cut model can significantly outperform the baseline methods of the word-based VSM (Vector Space Model) and the phrase-based VSM. The word-based VSM is just the conventional VSM [22] for information retrieval in which words are used to represent queries and documents (in our case, they are questions). The phrase-based VSM differs from the word-based VSM in that it uses the extracted topic terms to represent questions.

The contribution of this paper can be summarized as,

- To the best of our knowledge, this paper is the first effort of question recommendation. As will be illustrated in Section 2, the problem of question recommendation is different from query suggestion/substitution of the web search.
- The MDL-based tree cut model is proposed to tackle two important issues regarding question recommendations. The two issues are *reduction of topic terms* and *discrimination between question topic and question focus.* The MDL-based approach enables the question recommendation to optimize balance between specificity and generality.
- Extensive experiments have been conducted to evaluate the proposed approach using a large collection of real questions provided at Yahoo! Answers.

The rest of the paper is organized as follows: Section 2 introduces the related work. In Section 3, we formalize the problem of 'question recommendation'. In Section 4, we present our MDL-based approach to question recommendation. In Section 5, we empirically verify the effectiveness of the proposed approach. Section 6 summarizes our work and discusses the future work.

2. RELATED WORK

2.1 Question Search

As shown in Table 1, question search is the research area most related to question recommendation since it also deals with questions as targets directly. Given a queried question, question search is to find the questions that are semantically similar to the queried question. The major focus of the research is to tackle the lexical chasm problem between questions.

The research of question search is first conducted using FAQ data [3, 18, 23]. FAQ Finder [3] heuristically combines statistical similarities and semantic similarities between questions to rank FAQs. Conventional vector space models are used to calculate the statistical similarity and WordNet [8] is used to estimate the semantic similarity. Sneiders [23] proposed template based FAQ retrieval systems. Lai et al. [18] proposed an approach to automatically mine FAQs from the Web. However, they did not study the use of these FAQs after they were collected.

Recently, the research of question search has been further extended to the cQA. For example, Jeon et al. [13-15] compared four different retrieval methods, i.e. cosine similarity, Okapi, language model (LM), and statistical machine translation model (SMT), for automatically fixing the lexical chasm between questions of question search. They found that the SMT-based method performed the best.

Actually, there is a bar between search and recommendation. If we want to have close *question focus* as well as *question topic,* it is more suitable to achieve that with question search; if we want to have further *question focus* around certain *question topic,* it is better to leverage *question recommendation.*

2.2 Query Suggestion / Substitution

In the setting of web search, there are two research areas related to the problem of question recommendation, namely query suggestion and query substitution.

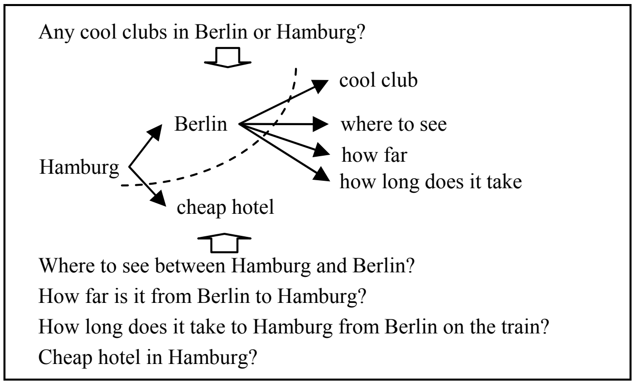
Query suggestion is a functionality to help users of a search engine to better specify their information need by suggesting related queries that has been frequently used by other users. The suggestions usually consist of synonymous queries and relevant queries. Thus, query suggestion is actually an analog of question search as shown in Table 1 in the setting of web search. In contrast to that, the suggestions provided by question recommendation may not be synonymous to the queried question.

Typical methods for query suggestion exploit query logs [7, 9, 12, 24] and document collections [11, 17], by assuming that in the same period of time, many users share the same or similar interests, which can be expressed in different manners. However, none of the methods is aimed at handling the sentence-level recommendation as question recommendation does.

Query substitution [16] replaces a user's original search query by generating a new query containing terms closely related to the original query. To compare that, as will be elaborated later, question recommendation is conducted to explore aspects which diverge from the original (question) query.

3. PROBLEM STATEMENT

Given an initial search query of a question *q*, we wish to recommend a question *q'* such that *q* and *q'* reflect different aspects of users' interests. We do this by substituting appropriate phrases as shown schematically in Figure 1.



Any cool clubs in Berlin or Hamburg?

Where to see between Hamburg and Berlin?
How far is it from Berlin to Hamburg?
How long does it take to Hamburg from Berlin on the train?
Cheap hotel in Hamburg?

Figure 1. An example on question recommendation

We consider a question as a combination of *question topic* and *question focus.* Question topic usually presents the major context/constraint of a question (e.g., Berlin, Hamburg) which characterizes users' interest. Question focus (e.g., cool club) presents certain aspect (or descriptive features) of the question topic. As for question recommendation, we are to substitute the question focus so that users can explore different aspects of their interests by reading recommendations.

In Figure 1, we assume that there exists a tree (graph) of topic terms representing the queried question and targeted questions. In the tree, the nodes representing question topic are expected to be closer to the root node than the nodes representing question focus. That is to say, the nodes representing question focus tend to appear as leaf nodes. Thus, as for question recommendation, we can then substitute the topic terms by beginning at the leaf nodes and stopping at certain level of the tree. We call the tree representing questions as "*question tree*".

In order to achieve the substitution-based recommendation as Figure 1 exemplifies, we need to solve the following two sub-problems:

- **Representing questions as trees (graphs) of topic terms**:

  Regarding this, we have two questions to answer: 1) how do we build the vocabulary of topic terms such that the vocabulary well models both the given questions and the new queried question? 2) Given the vocabulary of topic terms, how do we construct a question tree as that in Figure 1?

- **Ranking of recommendation candidates**

  Regarding this, we have also two questions to answer: 1) how do we discriminate the question focus from the question topic so that we can substitute the question focus for exploring different aspects of users' interest? Given a tree as presented in Figure 1, this question is equivalent to that of choosing a cut indicated by the dash line. 2) Based on a cut of question tree, how do we rank various choices of substitution? For example, in Figure 1, the possible substitutions include 'where to see', 'how far', 'how long does it take', and 'cheap hotel'.

In the next section, we are to explain our MDL-based approach to handling the issues raised by the questions above.

4. QUESTION RECOMMENDATION

Our approach to question recommendation consists of two steps: *represent questions as trees (graphs) of topic terms* and then *rank recommendation candidates*.

In this section, we are to explain the two steps in details. As our approach involves much use of a MDL-based tree cut model, we will first introduce what a *MDL-based tree cut model* is before the detailed explanation of the two steps of our approach.

4.1 Tree Cut Model and MDL

Formally, a tree cut model M [19] can be represented by a pair consisting of a tree cut Γ, and a probability parameter vector θ of the same length, that is,

$$M = (\Gamma, \theta) \tag{1}$$

where Γ and θ are

$$\Gamma = [C_1, C_2, \ldots C_k],\ \ \theta = [p(C_1), p(C_2), \ldots p(C_k)] \tag{2}$$

where $C_1, C_2, \ldots C_k$ are classes determined by a cut in the tree and $\sum_{i=1}^{k} p(C_i) = 1$. A 'cut' in a tree is any set of nodes in the tree that defines a partition of all the nodes, viewing each node as representing the set of child nodes as well as itself. For example, the cut indicated by the dash line in Figure 2 corresponds to three classes: $[n_0, n_{11}]$, $[n_{12}, n_{21}, n_{22}, n_{23}]$, and $[n_{13}, n_{24}]$. In the next two sub-sections, each node represents a topic term.

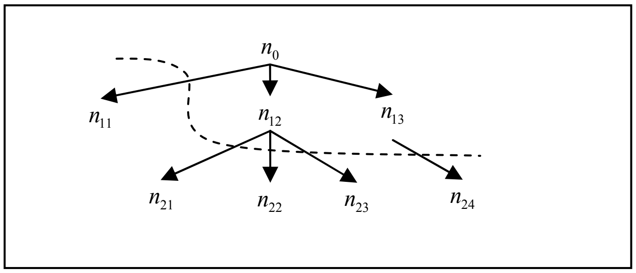


Figure 2. An example on the tree cut model

A straightforward way for determining a cut of a tree is to collapse the nodes of less frequency into its parent node. However, the method is too heuristic for it relies much on manually tuned frequency threshold. In our practice, we turn to use a theoretically well-motivated method based on the *MDL* (Minimum Description Length) principle. MDL is a principle of data compression and statistical estimation from information theory [2, 20, 21].

Given a sample S and a tree cut Γ, we employ MLE to estimate the parameters of the corresponding tree cut model $\hat{M} = (\Gamma, \hat{\theta})$, where $\hat{\theta}$ denotes the estimated parameters.

According to the MDL principle, the description length [19] $L(\hat{M}, S)$ of the tree cut model $\hat{M}$ and the sample S is the sum of the model description length $L(\Gamma)$, the parameter description length $L(\hat{\theta} \mid \Gamma)$, and the data description length $L(S \mid \Gamma, \hat{\theta})$, i.e.

$$L(\hat{M}, S) = L(\Gamma) + L(\hat{\theta} \mid \Gamma) + L(S \mid \Gamma, \hat{\theta}) \qquad (3)$$

The model description length $L(\Gamma)$ is a subjective quantity which depends on the coding scheme employed. Here, we simply assume that each tree cut model is equally likely *a priori.*

The parameter description length $L(\hat{\theta} \mid \Gamma)$ is calculated as

$$L(\hat{\theta} \mid \Gamma) = \frac{k}{2} \times \log|S| \qquad (4)$$

where $|S|$ denotes the sample size and k denotes the number of free parameters in the tree cut model, i.e. k equals the number of nodes in Γ minus one.

The data description length $L(S \mid \Gamma, \hat{\theta})$ is calculated as

$$L(S \mid \Gamma, \hat{\theta}) = -\sum_{n \in S} \log \hat{p}(n) \qquad (5)$$

where

$$\hat{p}(n) = \frac{1}{|C|} \times \frac{f(C)}{|S|} \qquad (6)$$

where C denotes the class of node *n*; $f(C)$ denotes the total frequency of nodes in class C in the sample S.

With the description length defined as (3), we wish to select a tree cut model with the minimum description length and output it as the result of reduction. Note that the model description length can be ignored because it is same for the tree cut models.

The MDL-based tree cut model was originally introduced for handling the problem of generalizing case frames using a thesaurus [19]. To the best of our knowledge, no existing work utilized it for question recommendation, or even query suggestion/substitution. This may be partially because of the unavailability of the resources (e.g., thesaurus) which can be used for embodying the queries/questions in a tree structure.

In the next two sub-sections, we will explain our methods for constructing the tree structure of topic terms and questions.

4.2 Representing Questions as Trees (Graphs) of Topic Terms

In our view, the problem of representing questions as trees of topic terms involves two issues: (a) acquiring topic terms; and (b) linking topic terms. We will elaborate our methods used to tackle the two issues in the following subsections.

4.2.1 Acquiring Topic Terms

The topic term acquisition process consists of two phases: *extraction of topic terms* and *reduction of topic terms*.

Extraction of Topic Terms

In general, there are many possible choices of linguistic units such as words, noun phrases, and n-grams, which can be used to representing topics. A good topic term should, together with other topic terms, capture the major meaning of a sentence and distinguish one topic from others. Words are usually too specific to outline the major meaning of sentences. Therefore, in our practice, the *extraction of topic terms* only considers noun phrases and n-grams as candidates of topic terms.

- BaseNP:

  A base noun phrase (BaseNP) is defined as a simple and non-recursive noun phrase [4]. In many cases Base NPs represent holistic and no-divisible concepts, and thus we extract BaseNPs (instead of noun phrases) as topic term candidates. The BaseNPs include both the multi-word terms (e.g., budget hotel, nice shopping mall) and the named entities (e.g., Berlin, Hamburg, Forbidden City). We make use of the tool introduced in [25] to extract the BaseNPs.

- WH-ngram:

  Ngram [5] of words can also be used as the topic term candidates. However, most meaningful n-grams are already covered by the BaseNPs. Thus, to complement the BaseNPs, we consider only using the WH-ngrams, which are the ngrams beginning with the WH-words. The WH-words include '*when*', '*what*', '*where*', '*which*', and '*how*'. Many ngrams (e.g., 'where to') are noisy terms. There are many methods [1, 6] for testing whether an ngram is a meaningful collocation/phrase. In our practice, we make use of the MDL-based tree cut model to eliminate the noisy WH-ngrams.

Table 2. The examples on topic terms

Type	Topic Term	Frequency
BaseNP	hotel	3983
	suite hotel	3
	embassy suite hotel	1
	nice suite hotel	2
	western hotel	40
	good western hotel	14
	inexpensive western hotel	12
	beachfront hotel	5
	good beachfront hotel	3
	great beachfront hotel	3
	nice hotel	224
	affordable hotel	48
WH-ngram	where	365
	where to learn	6
	where to learn computer	1
	where to learn Japanese	1
	where to buy	5
	where to buy ginseng	1
	where to buy insurance	23
	where to buy tea	12

Table 2 provide the example BaseNPs containing 'hotel' and the example WH-ngrams containing 'where'. Note that the table doesn't include all the topic term candidates containing 'hotel' or 'where'.

Reduction of Topic Terms

The *reduction of topic terms* is needed in order to represent the extracted topic terms more compactly as well as to judge the (degree of) reusability of the topic terms when applied to unseen data. That is to say, the reduction is to generate a vocabulary of topic terms which well models both the given questions and the new queried question.

For example, given the topic term candidates containing 'hotel' as list in Table 2, we wish to reduce the topic term "embassy suite hotel" as "suite hotel" because "embassy suite hotel" is too sparse and unlikely to be hit by the new question posted by other users. At the same time, we wish to keep "inexpensive western hotel" although "western hotel" is also one of the topic terms.

Formally, the *reduction of topic terms* is a decision-making process of, given a corpus of questions, what topic terms are more likely applicable to unseen questions. By the language of model selection, it is to select a model best fitting the given corpus and having good capability of generality. Within the model selection, each operation of *reduction of topic terms* results in a different model.

We make use of the MDL-based tree cut model for the model selection. In the following, we are to explain how a tree of topic terms is constructed such that it can model the process of reducing topic terms.

As for reduction, intuitively we want to ignore the modifier part of a topic term when reducing the topic term to another topic term. Thus, we define two types of reduction: a) removing the prefixes of BaseNPs; b) removing the suffixes of WH-ngrams.

We make use of the data structure of prefix tree (also known as trie) [10] for the representation of the BaseNPs or WH-ngrams. The two types of reduction correspond to two types of prefix tree, namely prefix tree of reversely-ordered BaseNPs and prefix tree of WH-ngrams. As for the reversely-ordered BaseNP, for example, "beachfront hotel" is rewritten as "hotel beachfront".

Figure 3 provides a prefix tree of the BaseNPs list in Table 2. Note that the orders of words are reversed before the BaseNPs are fed into the prefix tree. The numbers in the parentheses are the frequencies of the corresponding topic terms. For example, the node denoted by "beachfront (5)" means that the frequency of "beachfront hotel" is 5, which doesn't include the frequency of "good beachfront hotel" and that of "great beachfront hotel".

Figure 4 provides a prefix tree of the WH-ngrams list in Table 2. Note that the functional words such as 'to' and 'for' are skipped when the WH-ngrams are fed into the prefix tree.

For the prefix trees depicted in Figure 3 and 4, the root node (required by the definition of prefix tree) which is associated with *empty string* is ignored.

Given a prefix tree of topic terms, we can then employ the MDL principle for selecting the best cut. A prefix tree can have various cuts which correspond to different choices of topic terms. In Figure 3, the dot line and the dash line are just two of all the possible cuts. The selection given by the MDL is the cut indicated by the dash line, which results in the new tree at the bottom of Figure 3. In the new tree, for example, the frequency of "beachfront hotel" is updated as 11 to include "good beachfront hotel" and "great beachfront hotel". Similarly, the MDL can also give the best cut indicated by the dash line in Figure 4.

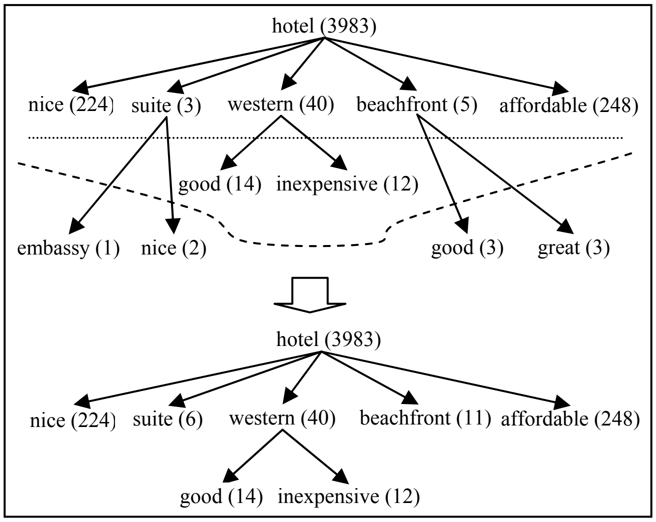


Figure 3. The prefix tree of the BaseNPs about 'hotel'

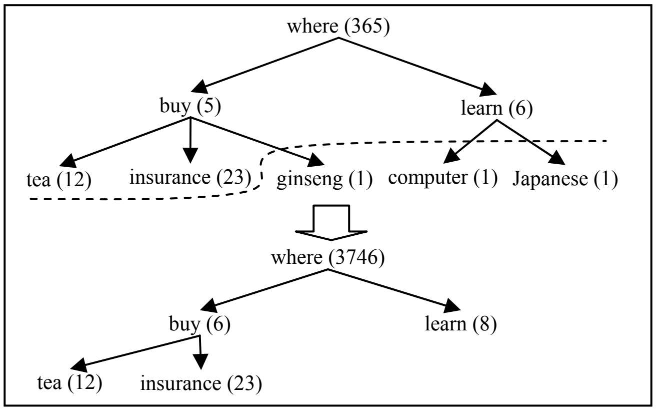


Figure 4. The prefix tree of the WH-ngrams about 'where'

4.2.2 Linking Topic Terms

Recall that our approach to question recommendation relies on a tree structure called '*question tree*' which consists of all the topic terms occurring in either the queried question or the related questions. In this subsection, with a series of definitions, we are to describe how a question tree is constructed from a collection of questions.

Definition 1 (Topic Profile) The *topic profile* θ_t of a topic term t in a categorized text collection is a probability distribution of categories $\{p(c \mid t)\}_{c \in C}$ where C is a set of category.

$$p(c \mid t) = \frac{count(c,t)}{\sum_{c \in C} count(c,t)} \qquad (7)$$

where $count(c,t)$ is the frequency of the topic term t within the category c. Clearly, we have $\sum_{c \in C} p(c \mid t) = 1$.

By 'categorized questions', we refer to the questions that are organized in a tree of taxonomy. For example, at Yahoo! Answers, the question "How do I install my wireless router" is categorized as "Computers & Internet → Computer Networking".

Definition 2 (Specificity) The *specificity* $s(t)$ of a topic term t is the inverse of the entropy of the topic profile θ_t. More specifically,

$$s(t) = 1/(-\sum_{c \in C} p(c \mid t) \log p(c \mid t) + \varepsilon) \qquad (8)$$

where ε is a smoothing parameter used to cope with the topic terms whose entropy are 0. In our practice, the value of ε is set as 0.001.

We use *specificity* to represent how specific a topic term is in characterizing information needs of users who post questions. A topic term of high specificity (e.g., Hamburg, Berlin) usually specifies the *question topic* corresponding to the main context of a question. Thus, a good question recommendation is required to keep it as much as possible so that the recommendation can be around the same context. A topic term of low specificity is usually used to represent the *question focus* (e.g., cool club, where to see) which is relatively volatile.

Actually, it is natural to think about an alternative definition of *specificity*. That is, defining *specificity* of a topic term *t* as the inverse of the number of categories in which *t* is mentioned.

$$s'(t) = 1/\sum_{c \in C} \delta(count(c,t) > 0)$$

where $\delta(\cdot)$ is an indicator function such that it equals to 1 when the condition is satisfied and 0 otherwise.

Let's see why the alternative definition is not appropriate for weighing *topic terms*. For example, in the category 'Europe' of Yahoo! Answers, you may find a question "I am from Beijing and like to have your recommendation on where to see in Berlin". The similar questions containing 'Beijing' can also be found in other categories although most occurrences of 'Beijing' are in the category "Asia Pacific → China". The alternative definition of *specificity* may mistakenly consider "Beijing" as a very general topic term.

Definition 3 (Topic Chain) A topic chain q^c of a question q is a sequence of ordered topic terms $t_1 \to t_2 \to \cdots \to t_m$ such that

1) t_i is included in q, $1 \le i \le m$;
2) $s(t_k) > s(t_l)$, $1 \le k < l \le m$.

For example, the topic chain of "any cool clubs in Berlin or Hamburg?" is "Hamburg → Berlin → cool club" because the *specificities* for 'Hamburg', 'Berlin', and 'cool club' are 0.99, 0.62, and 0.36.

Definition 4 (Question Tree) A question tree of a question set $Q = \{q_i\}_{i=1}^N$ is a prefix tree built over the topic chains $Q^c = \{q_i^c\}_{i=1}^N$ of the question set Q. Clearly, if a question set contains only one question, its question tree will be exactly same as the topic chain of the question.

Note that the root node of a question tree is associated with *empty string* as the definition of prefix tree requires [10].

Given the following topic chains with respect to the questions in Figure 1,

- Hamburg → Berlin → cool club
- Hamburg → Berlin → where to see
- Hamburg → Berlin → how far
- Hamburg → Berlin → how long does it take
- Hamburg → cheap hotel

we can have the question tree presented in the middle of Figure 1.

4.3 Ranking Recommendation Candidates

As introduced in Section 3, question recommendation defined in this paper is conducted by substituting consistent topic terms. Therefore, given a question represented as a topic chain, we should be able to 1) determine what consistent topic terms (representing *question focus*) should be substituted, and 2) score the recommendation candidates rendered by various substitutions.

We will address these two issues in the following sub-sections.

4.3.1 Determination of Substitutions

The topic terms of *low specificity* are usually used to represent the *question focus* (e.g., cool club, where to see), which are relatively volatile. *Determination of substitutions* is to discriminate these topic terms from those of *high specificity* and then suggest them as substitutions.

Recall that the topic terms in a topic chain of question are ordered according to their specificity values. Thus, a cut of a topic chain naturally gives a decision of discriminating the topic terms of low specificity (representing question focus) from the topic terms of high specificity (representing question topic).

Given a topic chain of question consisting of m topic terms, there exist $(m-1)$ possible cuts. Each possible cut gives one kind of suggestion on substitution. A straightforward method is just to take the $(m-1)$ cuts and then on the basis of them suggest $(m-1)$ kinds of substitutions. However, such a simple method complicates the problem of ranking recommendation candidates for it introduces too much uncertainty. In addition to that, the simple method cannot leverage the related questions which provide more observations of the topic terms.

Thus, we propose using the MDL-based tree cut model for the search of best cut of a topic chain. The best cut will be unique for a single topic chain.

Given a topic chain q^c of a question q, we are to construct a question tree of the related questions as follow:

1) Collect a set of topic chains $Q^c = \{q_i^c\}_{i=1}^N$ such that at least one topic term occurs in both q^c and q_i^c
2) Construct a question tree from the set of topic chains $Q^c \bigcup q^c$

Then, with the MDL-based tree cut model, we can obtain a best cut of the question tree, which also gives a cut for each topic chain (including q^c).

The uniqueness of the best cut can dramatically reduce the uncertainty of recommendation ranking, compared to the simple method. In addition to that, the best cut is obtained by observing the distribution of topic terms over all the potential recommendations (all the questions related to a queried question), instead of the queried question only.

A cut of a topic chain q^c separates the topic chain in two parts: HEAD and TAIL. HEAD (denoted as $H(q^c)$ is the subsequence of the original topic chain q^c before the cut. TAIL (denoted as

$T(q^c)$ is the subsequence of the original topic chain q^c after the cut. Thus, $q^c = H(q^c) \rightarrow T(q^c)$.

4.3.2 Scores of Recommendation Candidates

The ranking of recommendation candidates can be based on a recommendation score $r(\tilde{q} \mid q)$ defined over a queried question q and a recommendation candidate $\tilde{q}$ such that, given that $\tilde{q}_1$ and $\tilde{q}_2$ are both recommendation candidates of the query q, $\tilde{q}_1$ is the better recommendation of q than $\tilde{q}_2$ if $r(\tilde{q}_1 \mid q) > r(\tilde{q}_2 \mid q)$.

Given that the topic chain of a queried question q is separated as $q^c = H(q^c) \rightarrow T(q^c)$ by a cut and the topic chain of a recommendation candidate $\tilde{q}$ is separated as $\tilde{q}^c = H(\tilde{q}^c) \rightarrow T(\tilde{q}^c)$, we further require that the recommendation score $r(\tilde{q} \mid q)$ satisfies,

- **Specificity:** The more similar are $H(q^c)$ and $H(\tilde{q}^c)$, the greater $r(\tilde{q} \mid q)$ is;
- **Generality:** The more similar are $T(q^c)$ and $T(\tilde{q}^c)$, the less $r(\tilde{q} \mid q)$ is.

The requirements assure that the substitutions given by recommendation candidates focus on the TAIL part of the topic chain, which provides users the opportunity of exploring different question focus (e.g., cool club vs. where to see) around the same question topic (e.g., Hamburg, Berlin).

In order to define the recommendation score, we first introduce a score $sim(q_2^c \mid q_1^c)$ measuring the similarity of the topic chain q_1^c to q_2^c,

$$sim(q_2^c \mid q_1^c) = \frac{1}{|q_1^c|} \sum_{t_1 \in q_1^c} s(t_1) \cdot \max_{t_2 \in q_2^c} PMI(t_1, t_2) \qquad (9)$$

where $|q_1^c|$ represents the number of topic terms contained in q_1^c. $PMI(t_1, t_2)$ represents the *PMI* (Pointwise Mutual Information) [6] of a pair of topic terms t_1 and t_2. In our experiments, the *PMI* values are calculated over questions. According to the equation, the similarity between topic chains is basically determined by the associations between consistent topic terms. The *PMI* values of individual pair of topic terms in the equation are weighed by the *specificity* of topic terms occurring in q_1^c. Note that the similarity defined here is asymmetric.

Then, we define the recommendation score $r(\tilde{q} \mid q)$ as,

$$r(\tilde{q} \mid q) = \lambda \cdot sim(H(\tilde{q}^c) \mid H(q^c)) - (1-\lambda) \cdot sim(T(\tilde{q}^c) \mid T(q^c)) \qquad (10)$$

This equation balances between the two requirements of *specificity* and *generality* in a way of linear interpolation. The higher value of λ implies that the recommendations tend to be similar to the queried question. While, the lower value of λ encourages the recommendations to explore the question focus different from that of the queried questions.

5. EXPERIMENTAL RESULTS

We have conducted experiments to verify the effectiveness of our approach to question recommendation. Particularly, we have investigated the use of the MDL-based tree cut model.

5.1 Dataset and Evaluation Measures

Dataset

We made use of the questions crawled from Yahoo! Answers for the evaluation. More specifically, we utilize the *resolved* questions under two of the top-level categories at Yahoo! Answers, namely 'travel' and 'computers & internet'. The crawled questions include 314,616 items from the 'travel' category and 210,785 items from the 'computers & internet' category. Each resolved question consists of three fields: 'title', 'description', and 'answers'.

We developed two test sets, one for the category 'travel' denoted as 'TREVAL TST', and the other for 'computers & internet' denoted as 'COM-INT TST'. In order to create the test sets, we randomly selected 100 questions for each category.

To obtain good question recommendations given a queried question, we employed the Vector Space Model (VSM) [22] to retrieve the top 20 results and did manual judgments. Given a returned result by VSM, an assessor is asked to label it with '*relevant*' or '*irrelevant*'. If a returned result is considered as a recommendation for the queried question, the assessor will label it '*relevant*'; otherwise, the assessor will label it as '*irrelevant*'. Two assessors were involved in the manual judgments. Each of them was asked to label 50 questions from 'TRAVEL TST' and 50 from 'COM-INT TST'. In the process of manual judgment, the assessors were presented only the *title*s of the questions (for both the queried questions and the returned questions). We assume that the titles of the questions already provide enough contextual information for understanding users' information needs. Table 3 provides the statistics on the final test set.

Table 3. Statistics on the test data

	# Queries	# Returned	# Relevant
TRAVEL TST	100	2,000	405
COM-INT TST	100	2,000	386

Baseline and Evaluation Measures

We utilized two baseline methods for demonstrating the effectiveness of our approach, the word-based VSM and the phrase-based VSM. The word-based VSM is just the conventional VSM [22] for information retrieval in which words are used to represent queries and documents (in our case, they are questions). The phrase-based VSM is similar to the word-based VSM but it uses the extracted topic terms to represent questions.

We made use of three measures for evaluating the results of question recommendation methods. They are MAP, R-precision, and Top N precision (P@N).

MAP is widely used in IR and is based on the assumption that there are two categories: positive (relevant) and negative (irrelevant) in ranking of instances (questions). MAP calculates the mean of average precisions over a set of queries. Given a query q_i, its average precision ($AvgP_i$) is defined as the average of precision after each positive (relevant) instance is retrieved.

$$AvgP_i = \sum_{j=1}^{M} \frac{P(j) \times \mathrm{pos}(j)}{\text{number of positive instances}} \qquad (11)$$

where j is the rank, M is the number of instances retrieved, *pos(j)* is a binary function to indicate whether the instance in the rank j is positive (relevant), and *P(j)* is the precision at the given cut-off rank j:

$$P(j) = \frac{\text{number of positive instances in top } j \text{ positions}}{j} \quad (12)$$

R-precision is defined as

$$R - precision = \frac{\sum_{i=1}^{K} P(R_i)}{K} \quad (13)$$

where R_i is number of recommendations (labeled as 'relevant') for the query i in the ground truth. K is the total number of queries (topics) in the evaluation set.

Top N precision ($P@N$) is defined as

$$P @ N = \frac{\sum_{i=1}^{K} P(N)}{K} \quad (14)$$

Here, $N = 1, 2, .., 10$.

5.2 Recommending Questions about Travel

In the experiments, we made use of the questions about 'travel' to test the performance of the proposed MDL-based approach to question recommendation. More specifically, we used the 100 queries in the test set 'TRAVEL TST' to search for recommendations from the 314,616 questions categorized as 'travel'. Note that only the questions occurring in the test set can be evaluated.

We made use of the taxonomy of questions provided at Yahoo! Answers for the calculation of *specificity of topic terms*. The taxonomy is organized in a tree structure. In the following experiments, we only utilized as the categories of questions the leaf nodes of the taxonomy tree (regarding 'travel'), which includes 355 categories.

We utilized the question titles only for the *extraction of topic term* and the calculation of PMI (Pointwise Mutual Information) [6].

We randomly divided the test queries into five even subsets and conducted 5-fold cross-validation experiments. In each trial, we tuned the parameter λ in the equation (10) with four of the five subsets and then applied it to one remaining subset. The experimental results reported below (except that in Figure 5) are those averaged over the five trials.

Basic results:

In Table 4, our approach includes all the components based on the MDL such as the MDL-based reduction of topic terms and the MDL-based selection of substitution.

Table 4. Recommending questions about 'travel'

Methods	R-Precision	P@5	MAP
VSM	0.235	0.226	0.321
PVSM	0.276	0.234	0.291
Our approach	**0.324**	**0.290**	**0.350**

From Table 4, we see that our approach outperforms the baseline approaches VSM and PVSM in terms of all the measures. We conducted significant test (t-test) on the improvements of our approach over VSM and PVSM in terms of R-Precision and P@5. The result indicates that the improvements are statistically significant (p-value < 0.05). We also conducted t-test on the improvements of our approach over VSM and PVSM in terms of MAP. The result indicates that the improvements are not statistically significant. That is to say, compared to the baseline methods, our approach does well on the top-rank results while achieving comparable performance on all the returned results. Note that the average number of the 'relevant' results for a queried question about 'travel' is 4.05 (405/100), which means that the R-Precision is an approximation of P@4.

Table 5 provides the TOP-3 recommendations which are given by VSM, PVSM, and our approach respectively. Both VSM and PVSM return as the top-1 recommendation the question which is semantically equivalent to the queried question. The top-2/top-3 recommendations given by VSM and the top-2 given by PVSM provide the 'hotel information' around the locations other than 'downtown Chicago'. Actually, all these recommendations cannot help the user (who posted the queried question) explore other aspects of users' interest ('downtown Chicago'). The reason is that neither VSM nor PVSM is aware that the *question topic* of the queried question is the 'downtown Chicago'. In contrast, our approach can provide other aspects (*question focus*) about 'downtown Chicago' because of its awareness of the *question topic*.

Table 5. Recommendations for "What's a good but cheap hotel/motel/anything in downtown Chicago?"

Methods	Recommendations
VSM	1. What is a clean, cheap hotel near the downtown Chicago? 2. What is a good cheap hotel/motel near Disneyland? 3. What is a good cheap motel in Tapei?
PVSM	1. What is a clean, cheap hotel near the downtown Chicago? 2. Is there any cheap hotel/motel in Calgary Alberta? 3. What is there to do in downtown Chicago?
Our approach	1. What is there to do in downtown Chicago? 2. What are some fun cheap/free things to do & see in downtown Chicago? 3. What's the cost of a cab in downtown Chicago?

Effectiveness of MDL:

The major advantage of our approach is that the MDL-based tree cut model enables itself to leverage the global context for both *reduction of topic terms* and *selection of substitution.* In *reduction of topic terms*, the decision of reducing a topic term or not is made by observing the empirical distribution of all the extracted topic terms over the entire data collection. In *selection of substitution*, the decision of substituting a topic term or not is made by observing the entire related questions as well as the queried question.

To see how much the MDL-based tree cut model benefit the question recommendation, we introduce another three baseline methods for comparison. The first method (denoted as 'First') is made by removing the component '*reduction of topic terms*'. The second (denoted as 'Second') is to replace the *MDL-based selection of substitution* with the simple method of considering all the possible cuts of topic chains. In the simple method, any link within the topic chain of the queried question is considered as a cut. The third method (denoted as 'Third'), both removing the *reduction of topic terms* and replacing the *MDL-based selection of substitution,* is the combination of the first and the second.

Table 6. The advantage of the MDL-based tree cut model

Methods	R-Precision	P@5	MAP
First	0.302	0.296	0.330
Second	0.209	0.198	0.234
Third	0.153	0.156	0.172
Our approach	**0.324**	**0.290**	**0.350**

Table 6 provides the comparison. From Table 6, we see that either removing *reduction of topic terms* or replacing *the MDL-based selection of substitution* impairs the performance of question recommendation. The *t-test* also confirms that the performance drop rendered by replacing the MDL-based selection of substitution is statistically significant (p-value < 0.05). Although the improvement contributed by *reduction of topic terms* is not statistically significant, we still argue its use. This is because the improvement was achieved while the size of the vocabulary of topic terms was decreased dramatically. The size of the vocabulary is 289,251 before the *reduction* and 173,202 after the *reduction*. The reduction in size is about 40%.

The Use of Different Types of Topic Term

In this experiment, we are to see the use of the two types of topic term, namely BaseNP and WH-ngram. Table 7 illustrates that both BaseNPs and WH-ngrams are useful for characterizing the questions. In Table 7, the method '- BaseNP' denotes that of solely using WH-ngams in *extraction of topic terms* and '- WH-ngram' denotes that of solely using BaseNPs in *extraction of topic terms*. To see what roles the BaseNPs and WH-ngrams play in the topic chains, we further analyzed the 100 test queries (not including the recommendations) whose best cuts were given by the MDL-based tree cut model. The analysis results show that 63% (189/301) noun phrases occur in the HEAD of topic chains and 41% (25/61) WH-ngrams occur in the HEAD part. In other words, WH-ngrams are used as the *question focus* more often than noun phrases.

Table 7. The use of noun phrases and WH-ngrams

Methods	R-Precision	P@5	MAP
- BaseNP	0.035	0.044	0.039
- WH-ngram	0.321	0.290	0.346
Our approach	**0.324**	**0.290**	**0.350**

Aggregation Strategy:

The equation (9) provides one strategy, *max*, for aggregating the individual association of topic terms from two topic chains. Alternatively we can also make use of another strategy, *average*, as given by the equation (15).

$$sim(q_2^c \mid q_1^c) = \frac{1}{\left|q_1^c\right| \cdot \left|q_2^c\right|} \sum_{t_1 \in q_1^c, t_2 \in q_2^c} s(t_1) \cdot PMI(t_1, t_2) \qquad (15)$$

With *max* strategy, the similarity of two topic chains is determined by the similarity of the pair of topic terms which are most close to each other in terms of *PMI*. With the *average* strategy, the similarity of two topic chains is determined by the average of similarities of the pairs of topic terms.

Table 8. Aggregation strategy: max vs. average

Methods	R-Precision	MAP	P@5
average	0.197	0.237	0.188
max	**0.324**	**0.290**	**0.350**

From Table 8, we see that the *max* strategy outperforms the *average* strategy significantly (*p*-value < 0.05). This suggests that the similarity of two topic chains is better basing on the similarity of the closest topic terms (in terms of *PMI*) from the two topic chains.

Balance between Specificity and Generality

In equation (10), we use the parameter λ to trade-off the *specificity* and *generality* of recommendations. The higher value of λ implies that the recommendations tend to be similar to the queried question. While, the lower value of λ biases the recommendations to explore the question focus different from that of the queried questions. In this experiment, we are to explore how influential the value of λ is on the performance of question recommendation.

Figure 5 provides the change of R-Precision with respect to λ . The result was obtained with the 100 queries directly, instead of the five-fold cross-validation. From Figure 5, we see that the proposed approach performs best when λ is around 0.7. Therefore, at the point of 0.7, the question recommendation can best balance the *specificity* and *generality*.

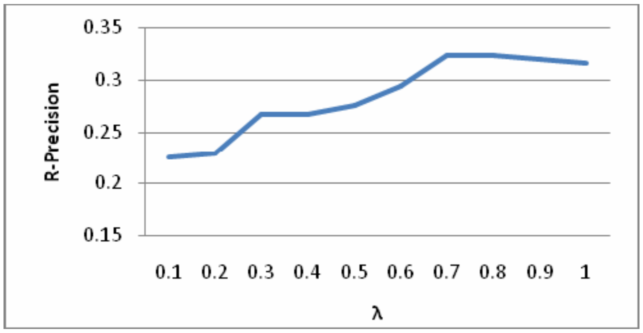


Figure 5. Balancing between specificity and generality.

However, one might expect that the best λ is around 0.5. The reason for the existence of the controversy (0.7 vs. 0.5) is that the similarity between two topic terms in the HEAD of a topic chain is not of the same scale as that between topic terms in the TAIL. For example, it is unusual to see the co-occurrence of 'where to go' and 'where to see' within a single question, which implies almost 100% negative association. In contrast, we can observe the co-occurrence of 'Hamburg' and 'Berlin' within a single question, which gives a moderate positive association. Thus, the question recommendation tends to use a bigger value of λ for the *specificity* (the HEAD part of a topic chain) as a re-scaling mechanism as well.

5.3 Recommending Questions about Computers & Internet

In the experiments, we made use of the questions about 'computers & internet' to test the performance of the proposed MDL-based approach to question recommendation. More specifically, we used the 100 queries in the test set 'COM-INT TST'' to search for recommendations from the 210,785 questions categorized as 'computers & internet'.

We utilized as the categories of questions the leaf nodes of the taxonomy tree regarding 'computers & Internet', which include 23 categories. We made use of only the question titles for the *extraction of topic term* and the calculation of *PMI*.

Again we conducted 5-fold cross-validation for the tuning of the parameter λ. The experimental results reported in Table 9 are averaged over the five trials. In Table 9, our approach made use of the same technique as presented in Table 4.

Table 9. Recommending questions about 'computers & Internet'

Methods	R-Precision	P@5	MAP
VSM	0.216	0.200	0.307
PVSM	0.242	0.214	0.257
Our approach	**0.316**	**0.248**	**0.316**

Again, we see that our approach outperforms the baseline approaches VSM and PVSM in terms of all the measures. We conducted significant test (t-test) on the improvements of our approach over VSM and PVSM. The result indicates that the improvements in terms of R-Precision and P@5 are statistically significant (p-value < 0.05). Thus, we can draw the same conclusion with the 'computers & internet' data as that with the 'travel' data. That is, compared to the baseline methods, our approach does well on the top-rank results while achieving comparable performance on all the returned results. Note that the average number of the 'relevant' results for a queried question about 'computers & internet' is 3.86 (386/100), which means that the R-Precision is an approximation of P@4.

6. CONCLUSIONS

In this paper, we have proposed to address the problem of question recommendation as a complement of question search.

The question recommendation was defined as substituting *question focus*. Under the setting, we have developed a new approach which consists of two steps: *representing questions as trees (graphs) of topic terms* and *ranking recommendation candidates*. We have proposed using the MDL-based tree cut model for tackling the issues involved in these two steps.

Experimental results indicate that our approach performs significantly better than the baseline methods of the word-based VSM and the phrase-based VSM. The results also show that the use of the MDL-based tree cut model is essential to our approach.

Though only utilizing the data from community-based question answering service (cQA), we also can find the categorized questions at forum sites or FAQ sites. Thus, as one of our future work, we will try to investigate the effectiveness of the proposed approach for other kinds of web services.

The proposed approach is not limited to question recommendation. Long queries of web search usually provide enough context information such as 'topic' and 'focus' as questions do. Thus, as the other future work, we will try to apply the proposed approach to query suggestion of long queries.

ACKNOWLEDGMENTS

We thank the anonymous reviewers for their valuable comments and suggestions to this paper.

REFERENCES

[1] Banerjee, S. and Pedersen, T. The design, implementation, and use of the ngram statistics package. In *Proc. of the 4th CICLing'03*.

[2] Barron, A., Rissanen, J., and Yu, B. The minimum description length principle in coding and modeling. *IEEE Trans. Information Theory*, vol. 44 (1998), pp. 2743-2760.

[3] Burke, R. D., Hammond, K. J., Kulyukin, V. A., Lytinen, S. L., Tomuro, N., and Schoenberg, S. Question answering from frequently asked question files: Experiences with the faq finder system. Technical report, 1997.

[4] Cao, Y. and Li, H. Base noun phrase translation using web data and the EM algorithm. In *Proc. of COLING'02*.

[5] Christopher, M. D. and Hinrich, S. Foundations of Statistical Natural Language Processing. MIT Press: 1999.

[6] Church, K. W. and Hanks, P. Word association norms, mutual information, and lexicography. In *Proc. of ACL'89*.

[7] Cuerzan, S. and White, R. W. Query Suggestion based on landing pages. In *Proc. of SIGIR'07*.

[8] Fellbaum, C. WordNet: An Electronic Lexical Database. MIT Press, 1998.

[9] Fonseca, B. M., Golgher, P. B., Moura, E. S., Possas, B., and Ziviani, N. Discovering search engine related queries using association rules. *Journal of Web Engineering*, 2003.

[10] Fredkin, E. Trie Memory. Communications of the ACM, D. 3(9):490-499, 1960.

[11] Gleich, D. and Zhukov, L. SVD based term suggestion and ranking system. In *Proc. of ICDM'04*.

[12] Jensen, E. C., Beitzel, S. M., Chowdhury, A., and Frieder, O. Query phrase suggestion from topically tagged session logs. In *Proc. of FQAS'06*.

[13] Jeon, J. and Croft, W. B. Learning translation-based language models using Q&A archives. Technical Report, University of Massachusetts.

[14] Jeon, J., Croft, W. B., and Lee, J. Finding similar questions in large question and answer archives. In *Proc. of CIKM'05*.

[15] Jeon, J., Croft, W. B., and Lee, J. H. Finding semantically similar questions based on their answers. In *Proc. of SIGIR'05*.

[16] Jones, R., Rey, B., Madani, O., and Greiner, W. Generating query substitutions. In *Proc. of WWW'06*.

[17] Kawamae, N., Suzuki, H., and Mizuno, O. Query and content suggestion based on latent interest and topic class. In *Proc. of WWW'04*.

[18] Lai, Y.-S., Fung, K.-A., and Wu, C.-H. Faq mining via list detection. In *Proc.* of the Workshop on Multilingual Summarization and Question Answering, 2002.

[19] Li, H. and Abe, N. Generalizing Case Frames using a thesaurus and the MDL principle. *Computational Linguistics*, 24(2), pp.217-244, 1998.

[20] Rissanen, J. Modeling by shortest data description. *Automatica*, vol. 14 (1978), pp. 465-471.

[21] Rissanen, J. Universal coding information, prediction and estimation. *IEEE Transaction on Information Theory*, vol. 30(4): 629-636.

[22] Salton, G., Wong, A., and Yang, C. S. A vector space model for automatic indexing. Communications of the ACM, vol. 18, nr. 11, pages 613–620.

[23] Sneiders, E. Automated question answering using question templates that cover the conceptual model of the database. In *Proc. of the 6th NLDB'02*.

[24] Wen, J. R., Nie, J.-Y., and Zhang, H. J. Query clustering using user logs. *ACM Trans. Information Systems*, 20(1):59-81, 2002.

[25] Xun, E. D., Huang, C.N., and Zhou, M. A unified statistical model for the identification of English BaseNP. *In Proc. of ACL'00*.