Searching Questions by Identifying Question Topic and Question Focus

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

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

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

Abstract

This paper is concerned with the problem of question search. In question search, given a question as query, we are to return questions semantically equivalent or close to the queried question. In this paper, we propose to conduct question search by identifying question topic and question focus. More specifically, we first summarize questions in a data structure consisting of question topic and question focus. Then we model question topic and question focus in a language modeling framework for search. We also propose to use the MDL-based tree cut model for identifying question topic and question focus automatically. Experimental results indicate that our approach of identifying question topic and question focus for search significantly outperforms the baseline methods such as Vector Space Model (VSM) and Language Model for Information Retrieval (LMIR).

1 Introduction

Over the past few years, online services have been building up very large archives of questions and their answers, for example, traditional FAQ services and emerging community-based Q&A services (e.g., Yahoo! Answers[1], Live QnA[2], and Baidu Zhidao[3]).

To make use of the large archives of questions and their answers, it is critical to have functionality facilitating users to search previous answers. Typically, such functionality is achieved by first retrieving questions expected to have the same answers as a queried question and then returning the related answers to users. For example, given question *Q1* in Table 1, question *Q2* can be returned and its answer will then be used to answer *Q1* because the answer of *Q2* is expected to partially satisfy the queried question *Q1*. This is what we called *question search.* In question search, returned questions are *semantically equivalent or close* to the queried question.

Query: *Q1: Any cool clubs in Berlin or Hamburg?*
Expected: *Q2: What are the best/most fun clubs in Berlin?*
Not Expected: *Q3: Any nice hotels in Berlin or Hamburg?* *Q4: How long does it take to Hamburg from Berlin?* *Q5: Cheap hotels in Berlin?*

Table 1. An Example on Question Search

Many methods have been investigated for tackling the problem of question search. For example, Jeon et al. have compared the uses of four different retrieval methods, i.e. vector space model, Okapi, language model, and translation-based model, within the setting of question search (Jeon et al., 2005b). However, all the existing methods treat questions just as plain texts (without considering question structure). For example, obviously, *Q2* can be considered semantically closer to *Q1* than *Q3-Q5* although all questions (*Q2-Q5*) are related to *Q1.* The existing methods are not able to tell the difference between question *Q2* and questions *Q3*, *Q4,* and *Q5* in terms of their relevance to question *Q1*. We will clarify this in the following.

In this paper, we propose to conduct question search by identifying question topic and question focus.

The question topic usually represents the major context/constraint of a question (e.g., Berlin, Hamburg) which characterizes users' interests. In contrast, question focus (e.g., cool club, cheap hotel) presents certain aspect (or descriptive features) of the question topic. For the aim of retrieving semantically equivalent (or close) questions, we need to

[1] http://answers.yahoo.com
[2] http://qna.live.com
[3] http://zhidao.baidu.com

Proceedings of ACL-08: HLT, pages 156–164,
Columbus, Ohio, USA, June 2008. ©2008 Association for Computational Linguistics

assure that returned questions are related to the queried question with respect to both question topic and question focus. For example, in Table 1, *Q2* preserves certain useful information of *Q1* in the aspects of both question topic (Berlin) and question focus (fun club) although it loses some useful information in question topic (Hamburg). In contrast, questions *Q3-Q5* are not related to *Q1* in question focus (although being related in question topic, e.g. Hamburg, Berlin), which makes them unsuitable as the results of question search.

We also propose to use the MDL-based (Minimum Description Length) tree cut model for automatically identifying question topic and question focus. Given a question as query, a structure called *question tree* is constructed over the question collection including the queried question and all the related questions, and then the MDL principle is applied to find a *cut* of the question tree specifying the question topic and the question focus of each question.

In a summary, we summarize questions in a data structure consisting of *question topic* and *question focus*. On the basis of this, we then propose to model question topic and question focus in a language modeling framework for search. To the best of our knowledge, none of the existing studies addressed question search by modeling both question topic and question focus.

We empirically conduct the question search with questions about 'travel' and 'computers & internet'. Both kinds of questions are from Yahoo! Answers. Experimental results show that our approach can significantly improve traditional methods (e.g. VSM, LMIR) in retrieving relevant questions.

The rest of the paper is organized as follow. In Section 2, we present our approach to question search which is based on identifying question topic and question focus. In Section 3, we empirically verify the effectiveness of our approach to question search. In Section 4, we employ a translation-based retrieval framework for extending our approach to fix the issue called 'lexical chasm'. Section 5 surveys the related work. Section 6 concludes the paper by summarizing our work and discussing the future directions.

2 Our Approach to Question Search

Our approach to question search consists of two steps: (a) summarize questions in a data structure consisting of question topic and question focus; (b) model question topic and question focus in a language modeling framework for search.

In the step (a), we employ the MDL-based (Minimum Description Length) tree cut model for automatically identifying question topic and question focus. Thus, this section will begin with a brief review of the MDL-based tree cut model and then follow that by an explanation of steps (a) and (b).

2.1 The MDL-based tree cut model

Formally, a tree cut model M (Li and Abe, 1998) can be represented by a pair consisting of a tree cut Γ, and a probability parameter vector θ of the same length, that is,

$$M = (\Gamma, \theta) \quad (1)$$

where Γ and θ are

$$\Gamma = [C_1, C_2, .. C_k], \\ \theta = [p(C_1), p(C_2), \dots, p(C_k)] \quad (2)$$

where $C_1, C_2, \dots C_k$ 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 1 corresponds to three classes: $[n_0, n_{11}]$, $[n_{13}, n_{24}]$, and $[n_{12}, n_{21}, n_{22}, n_{23}]$.

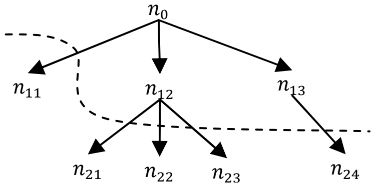

Figure 1. An Example on the Tree Cut Model

A straightforward way for determining a cut of a tree is to collapse the nodes of less frequency into their parent nodes. 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 principle. MDL is a principle of data compression and statistical estimation from information theory (Rissanen, 1978).

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 (Li and Abe, 1998) $L(\hat{M}, S)$ 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 $L(\hat{\theta}|\Gamma)$, and the data description length $L(S|\Gamma,\hat{\theta})$, i.e.

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

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 $L(\hat{\theta}|\Gamma)$ is calculated as

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

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 $L(S|\Gamma,\hat{\theta})$ is calculated as

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

where

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

where C is the class that n belongs to and $f(C)$ denotes the total frequency of instances 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. Note that the model description length $L(\Gamma)$ can be ignored because it is the same for all tree cut models.

The MDL-based tree cut model was originally introduced for handling the problem of generalizing case frames using a thesaurus (Li and Abe, 1998). To the best of our knowledge, no existing work utilizes it for question search. This may be partially because of the unavailability of the resources (e.g., thesaurus) which can be used for embodying the questions in a tree structure. In Section 2.2, we will introduce a tree structure called *question tree* for representing questions.

2.2 Identifying question topic and question focus

In principle, it is possible to identify *question topic* and *question focus* of a question by only parsing the question itself (for example, utilizing a syntactic parser). However, such a method requires accurate parsing results which cannot be obtained from the noisy data from online services.

Instead, we propose using the MDL-based tree cut model which identifies question topics and question foci for a set of questions together. More specifically, the method consists of two phases:

1) Constructing a *question tree*: represent the queried question and all the related questions in a tree structure called *question tree*;
2) Determining a *tree cut*: apply the MDL principle to the *question tree*, which yields the cut specifying *question topic* and *question focus.*

2.2.1 Constructing a question tree

In the following, with a series of definitions, we will describe how a *question tree* is constructed from a collection of questions.

Let's begin with explaining the representation of a question. A straightforward method is to represent a question as a bag-of-words (possibly ignoring stop words). However, this method cannot discern 'the hotels in Paris' from 'the Paris hotel'. Thus, we turn to use the linguistic units carrying on more semantic information. Specifically, we make use of two kinds of units: BaseNP (Base Noun Phrase) and WH-ngram. A BaseNP is defined as a simple and non-recursive noun phrase (Cao and Li, 2002). A WH-ngram is an ngram beginning with WH-words. The WH-words that we consider include '*when*', '*what*', '*where*', '*which*', and '*how*'. We refer to these two kinds of units as '*topic terms'.* With 'topic terms', we represent a question as a *topic chain* and a set of questions as a *question tree.*

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

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

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

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". Actually, we can find categorized questions at other online services such as FAQ sites, too.

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,

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

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

We use the term *specificity* to denote 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 because it tends to occur only in a few categories. 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 and might occur in many categories.

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

1) t_i is included in q, $1 \leq i \leq m$;

2) $s(t_k) > s(t_l)$, $1 \leq k < l \leq m$.

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 (Fredkin, 1960).

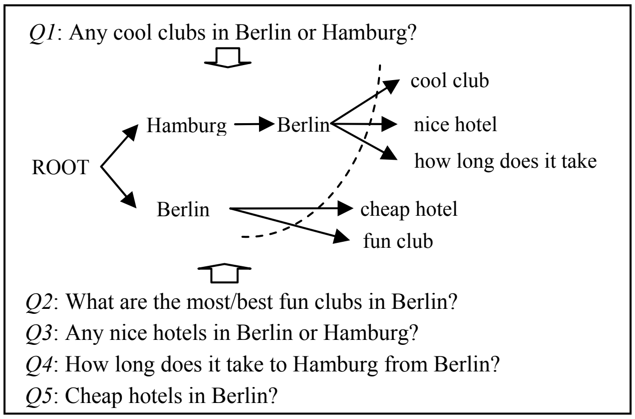
Q1: Any cool clubs in Berlin or Hamburg?
ROOT
Hamburg
Berlin
cool club
nice hotel
how long does it take
Berlin
cheap hotel
fun club
Q2: What are the most/best fun clubs in Berlin?
Q3: Any nice hotels in Berlin or Hamburg?
Q4: How long does it take to Hamburg from Berlin?
Q5: Cheap hotels in Berlin?

Figure 2. An Example of a Question Tree

Given the topic chains with respect to the questions in Table 1 as follow,

- *Q1*: Hamburg → Berlin → cool club
- *Q2*: Berlin → fun club
- *Q3*: Hamburg → Berlin → nice hotel
- *Q4*: Hamburg → Berlin → how long does it take
- *Q5*: Berlin → cheap hotel

we can have the question tree presented in Figure 2.

2.2.2 Determining the tree cut

According to the definition of a *topic chain,* the topic terms in a topic chain of a question are ordered by their specificity values. Thus, a cut of a topic chain naturally separates the topic terms of low specificity (representing question focus) from the topic terms of high specificity (representing question topic). Given a topic chain of a question consisting of m topic terms, there exist $(m-1)$ possible cuts. The question is: which cut is the best?

We propose using the MDL-based tree cut model for the search of the best cut in a topic chain. Instead of dealing with each topic chain individually, the proposed method handles a set of questions together. Specifically, given a queried question, we construct a question tree consisting of both the queried question and the related questions, and then apply the MDL principle to select the best cut of the question tree. For example, in Figure 2, we hope to get the cut indicated by the dashed line. The topic terms on the left of the dashed line represent the question topic and those on the right of the dashed line represent the question focus. Note that the tree cut yields a cut for each individual topic chain (each path) within the question tree accordingly.

A cut of a topic chain q^c of a question q 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 q^c after the cut. Thus, $q^c = H(q^c) \rightarrow T(q^c)$. For instance, given the tree cut specified in Figure 2, for the topic chain of *Q1* "Hamburg → Berlin → cool club", the HEAD and TAIL are "Hamburg → Berlin" and "cool club" respectively.

2.3 Modeling question topic and question focus for search

We employ the framework of language modeling (for information retrieval) to develop our approach to question search.

In the language modeling approach to information retrieval, the relevance of a targeted question $\tilde{q}$ to a queried question q is given by the probability $p(q|\tilde{q})$ of generating the queried question q

from the language model formed by the targeted question $\tilde{q}$. The targeted question $\tilde{q}$ is from a collection C of questions.

Following the framework, we propose a mixture model for modeling question structure (namely, question topic and question focus) within the process of searching questions:

$$p(q|\tilde{q}) = \lambda \cdot p(H(q)|H(\tilde{q})) + (1-\lambda) \cdot p(T(q)|T(\tilde{q})) \quad (9)$$

In the mixture model, it is assumed that the process of generating question topics and the process of generating question foci are independent from each other.

In traditional language modeling, a single multinomial model $p(t|\tilde{q})$ over terms is estimated for each targeted question $\tilde{q}$. In our case, two multinomial models $p\big(t\big|H(\tilde{q})\big)$ and $p\big(t\big|T(\tilde{q})\big)$ need to be estimated for each targeted question $\tilde{q}$.

If unigram document language models are used, the equation (9) can then be re-written as,

$$p(q|\tilde{q}) = \lambda \cdot \prod_{t \in H(q)} p\big(t\big|H(\tilde{q})\big)^{count(q,t)} + (1-\lambda) \cdot \prod_{t \in T(q)} p\big(t\big|T(\tilde{q})\big)^{count(q,t)} \quad (10)$$

where $count(q,t)$ is the frequency of t within q.

To avoid zero probabilities and estimate more accurate language models, the HEAD and TAIL of questions are smoothed using background collection,

$$p\big(t\big|H(\tilde{q})\big) = \alpha \cdot \hat{p}\big(t\big|H(\tilde{q})\big) + (1-\alpha) \cdot \hat{p}(t|C) \quad (11)$$

$$p\big(t\big|T(\tilde{q})\big) = \beta \cdot \hat{p}\big(t\big|T(\tilde{q})\big) + (1-\beta) \cdot \hat{p}(t|C) \quad (12)$$

where $\hat{p}(t|H(\tilde{q}))$, $\hat{p}(t|T(\tilde{q}))$, and $\hat{p}(t|C)$ are the MLE estimators with respect to the HEAD of $\tilde{q}$, the TAIL of $\tilde{q}$, and the collection C.

3 Experimental Results

We have conducted experiments to verify the effectiveness of our approach to question search. Particularly, we have investigated the use of identifying question topic and question focus for search.

3.1 Dataset and evaluation measures

We made use of the questions obtained from Yahoo! Answers for the evaluation. More specifically, we utilized the *resolved* questions under two of the top-level categories at Yahoo! Answers, namely 'travel' and 'computers & internet'. The 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'. For search we use only the 'title' field. It is assumed that the titles of the questions already provide enough semantic information for understanding users' information needs.

We developed two test sets, one for the category 'travel' denoted as 'TRL-TST', and the other for 'computers & internet' denoted as 'CI-TST'. In order to create the test sets, we randomly selected 200 questions for each category.

To obtain the ground-truth of question search, we employed the Vector Space Model (VSM) (Salton et al., 1975) to retrieve the top 20 results and obtained manual judgments. The top 20 results don't include the queried question itself. Given a returned result by VSM, an assessor is asked to label it with '*relevant*' or '*irrelevant*'. If a returned result is considered semantically equivalent (or close) to the queried question, the assessor will label it as '*relevant*'; otherwise, the assessor will label it as '*irrelevant*'. Two assessors were involved in the manual judgments. Each of them was asked to label 100 questions from 'TRL-TST' and 100 from 'CI-TST'. In the process of manually judging questions, the assessors were presented only the *title*s of the questions (for both the queried questions and the returned questions). Table 2 provides the statistics on the final test set.

	# Queries	# Returned	# Relevant
TRL-TST	200	4,000	256
CI-TST	200	4,000	510

Table 2. Statistics on the Test Data

We utilized two baseline methods for demonstrating the effectiveness of our approach, the VSM and the LMIR (language modeling method for information retrieval) (Ponte and Croft, 1998).

We made use of three measures for evaluating the results of question search methods. They are MAP, R-precision, and MRR.

3.2 Searching questions about 'travel'

In the experiments, we made use of the questions about 'travel' to test the performance of our approach to question search. More specifically, we used the 200 queries in the test set 'TRL-TST' to search for 'relevant' questions 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 randomly divided the test queries into five even subsets and conducted 5-fold cross-validation experiments. In each trial, we tuned the parameters λ, α, and β in the equation (10)-(12) with four of the five subsets and then applied it to one remaining subset. The experimental results reported below are those averaged over the five trials.

Methods	MAP	R-Precision	MRR
VSM	0.198	0.138	0.228
LMIR	0.203	0.154	0.248
LMIR-CUT	**0.236**	**0.192**	**0.279**

Table 3. Searching Questions about ‘Travel’

In Table 3, our approach denoted by LMIR-CUT is implemented exactly as equation (10). Neither VSM nor LMIR uses the data structure composed of question topic and question focus.

From Table 3, we see that our approach outperforms the baseline approaches VSM and LMIR in terms of all the measures. We conducted a significance test (t-test) on the improvements of our approach over VSM and LMIR. The result indicates that the improvements are statistically significant (p-value < 0.05) in terms of all the evaluation measures.

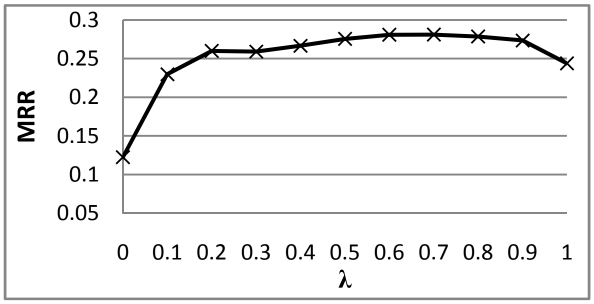

Figure 3. Balancing between Question Topic and Question Focus

In equation (9), we use the parameter λ to balance the contribution of question topic and the contribution of question focus. Figure 3 illustrates how influential the value of λ is on the performance of question search in terms of MRR. The result was obtained with the 200 queries directly, instead of 5-fold cross-validation. From Figure 3, we see that our approach performs best when λ is around 0.7. That is, our approach tends to emphasize question topic more than question focus.

We also examined the correctness of question topics and question foci of the 200 queried questions. The question topics and question foci were obtained with the MDL-based tree cut model automatically. In the result, 69 questions have incorrect question topics or question foci. Further analysis shows that the errors came from two categories: (a) 59 questions have only the HEAD parts (that is, none of the topic terms fall within the TAIL part), and (b) 10 have incorrect orders of topic terms because the specificities of topic terms were estimated inaccurately. For questions only having the HEAD parts, our approach (equation (9)) reduces to traditional language modeling approach. Thus, even when the errors of category (a) occur, our approach can still work not worse than the traditional language modeling approach. This also explains why our approach performs best when λ is around 0.7. The error category (a) pushes our model to emphasize more in question topic.

Methods	Results
VSM	1. How cold does it usually get in Charlotte, NC during winters? 2. How long and cold are the winters in Rochester, NY? **3. How cold is it in Alaska?**
LMIR	**1. How cold is it in Alaska?** 2. How cold does it get really in Toronto in the winter? 3. How cold does the Mojave Desert get in the winter?
LMIR-CUT	**1. How cold is it in Alaska?** **2. How cold is Alaska in March and outdoor activities?** 3. How cold does it get in Nova Scotia in the winter?

Table 4. Search Results for
“How cold does it get in winters in Alaska?”

Table 4 provides the TOP-3 search results which are given by VSM, LMIR, and LMIR-CUT (our approach) respectively. The questions in bold are labeled as ‘relevant’ in the evaluation set. The queried question seeks for the ‘weather’ information about ‘Alaska’. Both VSM and LMIR rank certain ‘irrelevant’ questions higher than ‘relevant’ questions. The ‘irrelevant’ questions are not about ‘Alaska’ although they are about ‘weather’. The reason is that neither VSM nor PVSM is aware that the query consists of the two aspects ‘weather’ (how cold, winter) and ‘Alaska’. In contrast, our approach assures that both aspects are matched. Note that the HEAD part of the topic chain of the queried question given by our approach is “Alaska” and the TAIL part is “winter → how cold”.

3.3 Searching questions about ‘computers & internet’

In the experiments, we made use of the questions about ‘computers & internet’ to test the performance of our proposed approach to question search. More specifically, we used the 200 queries in the test set ‘CI-TST’’ to search for ‘relevant’ questions from the 210,785 questions categorized as ‘computers & internet’. For the calculation of *specificity of topic terms*, we utilized as the categories of questions the leaf nodes of the taxonomy tree regarding ‘computers & Internet’, which include 23 categories.

We conducted 5-fold cross-validation for the parameter tuning. The experimental results reported in Table 5 are averaged over the five trials.

Methods	MAP	R-Precision	MRR
VSM	0.236	0.175	0.289
LMIR	0.248	0.191	0.304
LMIR-CUT	**0.279**	**0.230**	**0.341**

Table 5. Searching Questions about ‘Computers & Internet’

Again, we see that our approach outperforms the baseline approaches VSM and LMIR in terms of all the measures. We conducted a significance test (t-test) on the improvements of our approach over VSM and LMIR. The result indicates that the improvements are statistically significant (p-value < 0.05) in terms of all the evaluation measures.

We also conducted the experiment similar to that in Figure 3. Figure 4 provides the result. The trend is consistent with that in Figure 3.

We examined the correctness of (automatically identified) question topics and question foci of the 200 queried questions, too. In the result, 65 questions have incorrect question topics or question foci. Among them, 47 fall in the error category (a) and 18 in the error category (b). The distribution of errors is also similar to that in Section 3.2, which also justifies the trend presented in Figure 4.

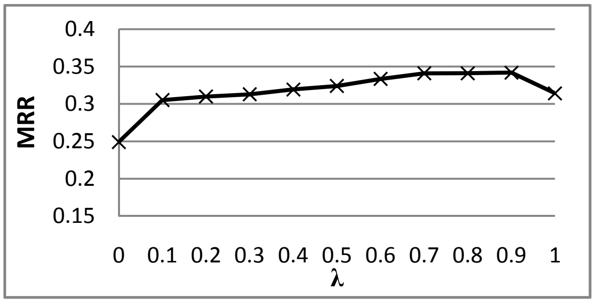

Figure 4. Balancing between Question Topic and Question Focus

4 Using Translation Probability

In the setting of question search, besides the topic what we address in the previous sections, another research topic is to fix lexical chasm between questions.

Sometimes, two questions that have the same meaning use very different wording. For example, the questions “where to stay in Hamburg?” and “the best hotel in Hamburg?” have almost the same meaning but are lexically different in question focus (where to stay vs. best hotel). This is the so-called ‘lexical chasm’.

Jeon and Bruce (2007) proposed a mixture model for fixing the lexical chasm between questions. The model is a combination of the language modeling approach (for information retrieval) and translation-based approach (for information retrieval). Our idea of modeling question structure for search can naturally extend to Jeon et al.’s model. More specifically, by using translation probabilities, we can rewrite equation (11) and (12) as follow:

$$\begin{aligned} p\big(t\big|H(\tilde{q})\big) = {} & \alpha_1 \cdot \hat{p}\big(t\big|H(\tilde{q})\big) \\ & +\alpha_2 \cdot \textstyle\sum_{t' \in H(\tilde{q})} Tr(t|t') \cdot \hat{p}\big(t'\big|H(\tilde{q})\big) \\ & +(1-\alpha_1-\alpha_2)\cdot \hat{p}(t|C) \end{aligned} \qquad (13)$$

$$\begin{aligned} p\big(t\big|T(\tilde{q})\big) = {} & \beta_1 \cdot \hat{p}\big(t\big|T(\tilde{q})\big) \\ & +\beta_2 \cdot \textstyle\sum_{t' \in T(\tilde{q})} Tr(t|t') \cdot \hat{p}\big(t'\big|T(\tilde{q})\big) \\ & +(1-\beta_1-\beta_2)\cdot \hat{p}(t|C) \end{aligned} \qquad (14)$$

where $Tr(t|t')$ denotes the probability that topic term t is the translation of t'. In our experiments, to estimate the probability $Tr(t|t')$, we used the collections of question titles and question descriptions as the parallel corpus and the IBM model 1 (Brown et al., 1993) as the alignment model.

Usually, users reiterate or paraphrase their questions (already described in question titles) in question descriptions.

We utilized the new model elaborated by equation (13) and (14) for searching questions about ‘travel’ and ‘computers & internet’. The new model is denoted as ‘SMT-CUT’. Table 6 provides the evaluation results. The evaluation was conducted with exactly the same setting as in Section 3. From Table 6, we see that the performance of our approach can be further boosted by using translation probability.

Data	Methods	MAP	R-Precision	MRR
TRL-TST	LMIR-CUT	0.236	0.192	0.279
	SMT-CUT	**0.266**	**0.225**	**0.308**
CI-TST	LMIR-CUT	0.279	0.230	**0.341**
	SMT-CUT	**0.282**	**0.236**	0.337

Table 6. Using Translation Probability

5 Related Work

The major focus of previous research efforts on question search is to tackle the lexical chasm problem between questions.

The research of question search is first conducted using FAQ data. FAQ Finder (Burke et al., 1997) 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 (Fellbaum, 1998) is used to estimate the semantic similarity. Sneiders (2002) proposed template based FAQ retrieval systems. Lai et al. (2002) proposed an approach to automatically mine FAQs from the Web. Jijkoun and Rijke (2005) used supervised learning methods to extend heuristic extraction of Q/A pairs from FAQ pages, and treated Q/A pair retrieval as a fielded search task.

Harabagiu et al. (2005) used a Question Answer Database (known as QUAB) to support interactive question answering. They compared seven different similarity metrics for selecting related questions from QUAB and found that the concept-based metric performed best.

Recently, the research of question search has been further extended to the community-based Q&A data. For example, Jeon et al. (Jeon et al., 2005a; Jeon et al., 2005b) compared four different retrieval methods, i.e. vector space model, Okapi, language model (LM), and translation-based model, for automatically fixing the lexical chasm between questions of question search. They found that the translation-based model performed best.

However, all the existing methods treat questions just as plain texts (without considering question structure). In this paper, we proposed to conduct question search by identifying question topic and question focus. To the best of our knowledge, none of the existing studies addressed question search by modeling both question topic and question focus.

Question answering (e.g., Pasca and Harabagiu, 2001; Echihabi and Marcu, 2003; Voorhees, 2004; Metzler and Croft, 2005) relates to question search. Question answering automatically extracts short answers for a relatively limited class of question types from document collections. In contrast to that, question search retrieves answers for an unlimited range of questions by focusing on finding semantically similar questions in an archive.

6 Conclusions and Future Work

In this paper, we have proposed an approach to question search which models question topic and question focus in a language modeling framework.

The contribution of this paper can be summarized in 4-fold: (1) A data structure consisting of *question topic* and *question focus* was proposed for summarizing questions; (2) The MDL-based tree cut model was employed to identify question topic and question focus automatically; (3) A new form of language modeling using question topic and question focus was developed for question search; (4) Extensive experiments have been conducted to evaluate the proposed approach using a large collection of real questions obtained from Yahoo! Answers.

Though we only utilize data from community-based question answering service in our experiments, we could also use categorized questions from forum sites and FAQ sites. Thus, as future work, we will try to investigate the use of the proposed approach for other kinds of web services.

Acknowledgement

We would like to thank Xinying Song, Shasha Li, and Shilin Ding for their efforts on developing the evaluation data. We would also like to thank Stephan H. Stiller for his proof-reading of the paper.

References

A. Echihabi and D. Marcu. 2003. A Noisy-Channel Approach to Question Answering. In *Proc. of ACL'03*.

C. Fellbaum. 1998. WordNet: An electronic lexical database. *MIT Press*.

D. Metzler and W. B. Croft. 2005. Analysis of statistical question classification for fact-based questions. *Information Retrieval*, 8(3), pages 481−504.

E. Fredkin. 1960. Trie memory. *Communications of the ACM*, D. 3(9):490−499.

E. M. Voorhees. 2004. Overview of the TREC 2004 question answering track. In *Proc. of TREC'04*.

E. Sneiders. 2002. Automated question answering using question templates that cover the conceptual model of the database. In *Proc. of the 6th International Conference on Applications of Natural Language to Information Systems,* pages 235−239.

G. Salton, A. Wong, and C. S. Yang 1975. A vector space model for automatic indexing. *Communications of the ACM*, vol. 18, nr. 11, pages 613−620.

H. Li and N. Abe. 1998. Generalizing case frames using a thesaurus and the MDL principle. *Computational Linguistics*, 24(2), pages 217−244.

J. Jeon and W.B. Croft. 2007. Learning translation-based language models using Q&A archives. Technical report, University of Massachusetts.

J. Jeon, W. B. Croft, and J. Lee. 2005a. Finding semantically similar questions based on their answers. In *Proc. of SIGIR'05.*

J. Jeon, W. B. Croft, and J. Lee. 2005b. Finding similar questions in large question and answer archives. In *Proc. of CIKM '05,* pages 84−90.

J. Rissanen. 1978. Modeling by shortest data description. *Automatica*, vol. 14, pages. 465−471

J.M. Ponte, W.B. Croft. 1998. A language modeling approach to information retrieval. In *Proc. of SIGIR'98*.

M. A. Pasca and S. M. Harabagiu. 2001. High performance question/answering. In *Proc. of SIGIR'01*, pages 366−374.

P. F. Brown, V. J. D. Pietra, S. A. D. Pietra, and R. L. Mercer. 1993. The mathematics of statistical machine translation: parameter estimation. *Computational Linguistics*, 19(2):263−311.

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

S. Harabagiu, A. Hickl, J. Lehmann and D. Moldovan. 2005. Experiments with Interactive Question-Answering. In *Proc. of ACL'05*.

V. Jijkoun, M. D. Rijke. 2005. Retrieving Answers from Frequently Asked Questions Pages on the Web. In *Proc. of CIKM'05*.

Y. Cao and H. Li. 2002. Base noun phrase translation using web data and the EM algorithm. In *Proc. of COLING'02*.

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