Large-Scale Question Classification in cQA by Leveraging Wikipedia Semantic Knowledge

Li Cai, Guangyou Zhou, Kang Liu, and Jun Zhao
National Laboratory of Pattern Recognition,
Institute of Automation, Chinese Academy of Sciences
95 Zhongguancun East Road, Beijing 100190, China
{lcai, gyzhou, kliu, jzhao}@nlpr.ia.ac.cn

ABSTRACT

With the flourishing of community-based question answering (cQA) services like Yahoo! Answers, more and more web users seek their information need from these sites. Understanding user's information need expressed through their search questions is crucial to information providers. Question classification in cQA is studied for this purpose. However, there are two main difficulties in applying traditional methods (question classification in TREC QA and text classification) to cQA: (1) Traditional methods confine themselves to classify a text or question into two or a few predefined categories. While in cQA, the number of categories is much larger, such as Yahoo! Answers, there contains 1,263 categories. Our empirical results show that with the increasing of the number of categories to moderate size, the performance of the classification accuracy dramatically decreases. (2) Unlike the normal texts, questions in cQA are very short, which cannot provide sufficient word co-occurrence or shared information for a good similarity measure due to the data sparseness.

In this paper, we propose a two-stage approach for question classification in cQA that can tackle the difficulties of the traditional methods. In the first stage, we preform a search process to prune the large-scale categories to focus our classification effort on a small subset. In the second stage, we enrich questions by leveraging Wikipedia semantic knowledge to tackle the data sparseness. As a result, the classification model is trained on the enriched small subset. We demonstrate the performance of our proposed method on Yahoo! Answers with 1,263 categories. The experimental results show that our proposed method significantly outperforms the baseline method (with error reductions of 23.21%).

Categories and Subject Descriptors

H.3.3 [**Information Systems**]: Information Storage and Retrieval

General Terms

Algorithms, Experimentation, Performance

CIKM'11, October 24–28, 2011, Glasgow, Scotland, UK.

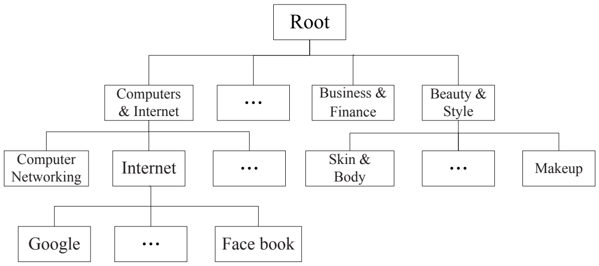

Figure 1: An example of Yahoo! Answers taxonomy.

Keywords

Question Retrieval, translation model, Wikipedia, Large-Scale Classification

1. INTRODUCTION

Community-based Question Answering (cQA) service is a particular form of online service for leveraging user-generated content, such as Yahoo! Answers [1] and Live QnA [2], which has attracted great attention from both academia and industry. Recently, efforts have been put to search similar questions [4, 22, 27] and recommend questions [3] in cQA. As a result, understanding the search and recommendation intent behind the questions issued by askers has become an important research problem. *Question Classification* (or *Question categorization*) is studied for this purpose by classifying user queried questions into a predefined target categories. Such category information can be used to help find the relevant questions in cQA archives [2]. An example of Yahoo! Answers taxonomy is shown in Figure 1. Unlike the traditional large-scale text taxonomy (e.g., Open Directory Project (ODP) and Yahoo! Directory), question classification in cQA is the task of classifying user queried questions into **predefined target categories at the leaf level**.[3]

Question classification in cQA is dramatically different from traditional text classification and factoid question classification from TREC QA. Directly applying these methods for question classification in cQA may lead to the following difficulties:

- Traditional methods confine themselves to classify a text [19]

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

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

[3] Classifying a questions into the top level (e.g., Computer & Internet) is too coarse to contain a category like "Face book" (it is in "Computer & Internet\Internet") whose subtopics might be of interest to many users. These suggest the necessity of automatic fine-grained question classification in cQA.

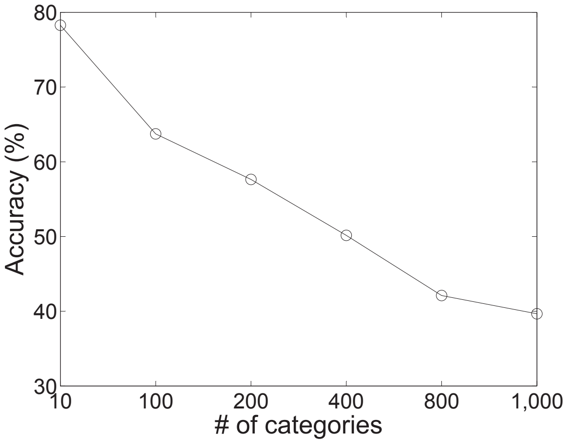

Figure 2: Number of categories vs. classification accuracy on validation data.

(benchmark collections, e.g., 20Newsgroups) or question [28] (e.g., TREC QA) into two or a few predefined categories. While in cQA, the number of categories is much larger than that in benchmark collections and TREC QA. For example in Yahoo! Answers, there contains 1,263 categories at the leaf level. Our empirical results show that with the increasing of the number of categories to a moderate size, the performance of the classification accuracy dramatically decreases. Figure 2 shows the classification accuracy vs. different number of categories on validation data using traditional methods.[4]

- Unlike the normal texts or documents, questions in cQA are usually much shorter, that is, they consist of only a few words for most questions. Because of the short lengths, they do not provide enough word co-occurrence or shared context for a good similarity measure. Therefore, traditional "bag of words" models usually fail to achieve satisfactory classification accuracy due to the data sparseness.

To address the above challenges, we propose a novel method that overcomes those difficulties and consequently improve the performance of question classification in cQA. As a first attempt to tackle the problem of large-scale category classification and data sparseness in cQA, we intend to answer the following questions: (1) For a given question, how to prune the large-scale categories into a much smaller subset of target category candidates? Therefore, we can build a local classification model on the small training data to optimize the performance for the target category candidates. (2) How to enrich questions by leveraging Wikipedia semantic knowledge in order to reduce the data sparseness? (3) How much improvement can we achieve using our proposed method? (4) Would our proposed method add too much computational burden and would it be possible to extend the idea for real world online services?

More specifically, our contributions are as follows.

- *To tackle the large-scale category classification problem*: We attempt to find a subset of categories related to a given question from the large-scale categories . Thus, the large-scale categories are pruned into a much smaller subset of target category candidates. It is intuitive that the classification performance on a smaller subset of target categories will be better than that on a larger set of categories, as shown in the experiments. For easy description, we call the above strategy as **search stage** in the rest of the paper (in Section 4).
- *To reduce the data sparseness*: We enrich questions by leveraging Wikipedia semantic knowledge due to the large coverage of concepts, rich semantic information and up-to-date content. However, Wikipedia is not a structural thesaurus like WordNet [12]. In this paper, we first build a *concept thesaurus* based on the semantic knowledge (synonym, hypernym, and associative concepts) extracted from Wikipedia. Then, we add the concepts into the questions for classification. For easy description, we call this strategy as **enrichment-based classification stage** in the rest of the paper (in Section 5).
- *To validate our proposed method*: We conduct experiments on Yahoo! Answers with 1,263 categories. Experimental results show that pruning the large-scale categories into a much smaller subset of target categories in the search stage can significantly improve the classification accuracy (i.e., eliminating more than 14.47% of errors) (see subsection 6.3.1). Moreover, enriching questions with Wikipedia concepts can further improve the accuracy (i.e., eliminating more than 12.31% of errors) (see subsection 6.3.2). Overall, training the classification model on the entire training data, the accuracy of our proposed method can achieve 52.20% on the test data, which is 23.21% error reductions over the baseline (see subsection 6.3.1). We also demonstrate that our proposed method does not add too much computational burden and it would be possible to extend for real world online services (see subsection 6.5).

The rest of this paper is organized as follows. In Section 2, we give a brief overview of related work. Section 3 presents our general framework. Section 4 describes the search stage to prune the large-scale target categories into a much smaller subset of target category candidates. Section 5 describes the enrichment-based classification stage with Wikipedia semantic knowledge. Experimental results are presented in Section 6. Finally, we conclude with ideas for future work.

2. RELATED WORK

Question classification in TREC QA has been intensively studied during the past decade. Many researchers have employed machine learning methods (e.g., maximum entropy and support vector machine) by using different features, such as syntactic features [28, 16] and semantic features [15]. However, these methods mainly focused on factoid questions and confined themselves to classify a question into two or a few predefined categories (e.g., "what", "how", "why", "when", "where" and so on). However, question classification in cQA is dramatically different from factoid question classification, as discussed in Section 1. Therefore, traditional methods may fail to achieve the satisfactory results.

Recently, Xue et al. [26] proposed a two-stage approach for large-scale text classification. In the search stage, a category-search algorithm is used to obtain category candidates for each document. Then in the classification stage, the classification model is trained on the small subset of the original taxonomy. Our approach stems from a similar motivation; however, we target the question classification in cQA. There are mainly two differences between our proposed methods and Xue et al. [26]. First, Xue et al. [26] obtain more training data by leveraging the structure information of ODP.

[4] The experiment is trained on the training data (2,000k questions) using traditional "bag of words" model (maximum entropy classifier) and tested on the validation data (10,000 questions)

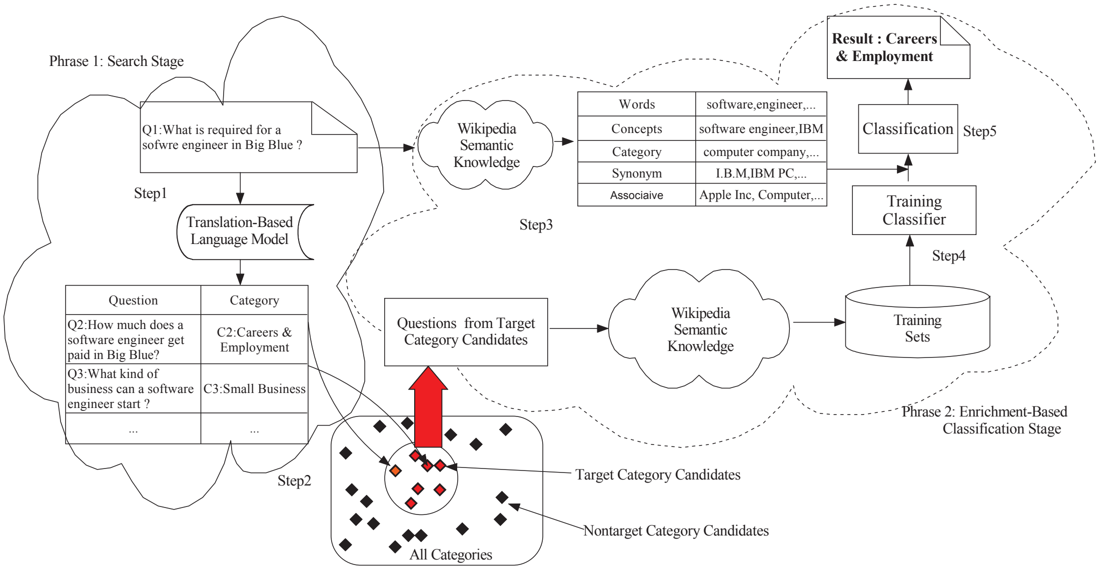
Phrase 1: Search Stage
Q1:What is required for a sofwre engineer in Big Blue ?
Step1
Translation-Based Language Model
Question
Category
Q2:How much does a software engineer get paid in Big Blue?
C2:Careers & Employment
Q3:What kind of business can a software engineer start ?
C3:Small Business
...
...
Step2
Wikipedia Semantic Knowledge
Words
software,engineer,...
Concepts
software engineer,IBM
Category
computer company,...
Synonym
I.B.M,IBM PC,...
Associaive
Apple Inc, Computer,...
Step3
Result : Careers & Employment
Classification
Step5
Training Classifier
Step4
Questions from Target Category Candidates
Wikipedia Semantic Knowledge
Training Sets
Phrase 2: Enrichment-Based Classification Stage
Target Category Candidates
Nontarget Category Candidates
All Categories

Figure 3: The general framework of our proposed approach.

In contrast, the taxonomy of Yahoo! Answers is not as deep as ODP, it is very difficult for us to utilize the structure information to collect more training data. Second, compared to the normal texts or documents in ODP, questions in cQA are very short, which leads to data sparseness. Therefore, data sparseness poses a great challenge in cQA.

To tackle the data sparseness, a variety of methods have been proposed in the literature. In general, these approaches can be divided into two categories. The first category is the basic representation of texts by exploiting phrases in the original texts from different aspects to preserve the contextual information [10, 20]. However, NLP technologies, such as parsing, are not employed as it is time assuming to apply such techniques to analyze the structure of the normal texts in detail. As a result, the methods fail to perform deep understanding of the original text.

The second category is to reduce the data sparseness by using the background knowledge. Hotho et al. [6] adopted various strategies to enrich text representation with synonyms and hypernyms from WordNet [12]. However, WordNet has limited coverage. Gabrilovich et al. [5] proposed a method to enrich documents with new features which are the concepts (Wikipedia titles) represented by the relevant Wikipedia articles. However, they do not make full use of the rich relations in Wikipedia such as hypernyms, synonyms and associated terms. Wang et al. [24] proposed to extract enrichment relations from Wikipedia and utilize the extracted relations to improve text classification. However, they treat the hypernyms and associative concepts equal with words in document.

3. THE GENERAL FRAMEWORK

In this section, we present the proposed framework for question classification in cQA. Our method works as follows. For a given question, the entire categories can be divided into two kinds: target category candidates and nontarget category candidates. For large-scale categories, the number of target category candidates for a question is much less than that of nontarget category candidates. Traditional classification methods only focus on building a global classification model to optimize the performance for all categories despite the fact that most of the categories may not be related to a given question. Our proposed approach can utilize such a property and thus focus on the categories related to the question. To this end, we extract a small subset of target category candidates from the large-scale categories by retrieving the semantically similar questions in the entire training data.

Then, we perform classification of the given question on these extracted small subset. However, unlike the normal texts or documents, questions in cQA are usually much shorter, that is, they consist of only a few words for most questions. Because of the short lengths, they do not provide enough word co-occurrence or shared context for a good similarity measure. Therefore, traditional "bag of words" model may fail to achieve satisfactory results due to the data sparseness. To reduce the data sparseness, we propose a novel method to enrich questions by leveraging Wikipedia semantic knowledge.

In order to illustrate the above ideas clearly, we give an example shown in Figure 3.

Step 1: Give question Q_1, we employ the state-of-the-art question retrieval model to get a list of semantically similar questions using, such as Q_2, Q_3, $\cdots$;

Step 2: The top-N most semantically similar questions are selected as related questions. Then we rank the target category candidates by counting the corresponding categories of these questions. The ranking is decided by the count of the categories. Finally, we can get a list of ranked target category candidates. Compared to the entire categories, this narrowing procedure helps reduce the number of target category candidates. In Figure 3, the target category candidates for question Q_1 are "Careers & Employment", "Small Business", "Software", $\cdots$;

Step 3: We enrich questions by leveraging Wikipedia semantic knowledge. Traditional "bag of words" representation is "software", "engineer", "big", "blue", $\cdots$. After mapping the given question into Wikipedia concept sets, Wikipedia concepts such as "IBM", "software engineer" can be identified from question Q_1. Moreover, semantic relations (synonym, hypernym, and associative relation) in Wikipedia are also used to expand the identified concepts (e.g., "software engineer", "IBM", "computer company", $\cdots$).

Step 4: We add the Wikipedia concepts into questions and build a local classification model for each given question.

Step 5: We add the Wikipedia concepts into the given question and classify the question using the trained local classification model.

4. SEARCH STAGE: RETRIEVAL THE TARGET CATEGORY CANDIDATES

In the search stage, we extract a small subset of target category candidates using the state-of-the-art question retrieval models.

For question retrieval, both translation-based language model (TRLM) [27] and syntactic tree matching (STM) [22] have gained state-of-the-art performance, while Ming et al. [14] compared the two methods and demonstrated that TRLM worked better than STM on the Yahoo! Answers data set. Therefore, we employ TRLM as state-of-the-art retrieval model to search the related questions.

4.1 Translation-Based Language Model

Translation-based language model (TRLM) is originally proposed to solve the lexical gap problem in question retrieval. The monolingual translation probabilities capture the lexical semantic relatedness between mismatched terms in the queried question and the historical questions in the training data. We employ TRLM to measure the semantic similarity between two questions q_1 and q_2. The similarity score function is similar to the retrieval function proposed by Xue et al. [27]:

$$P_{TRLM}(q_1|q_2) = \prod_{w \in q_1} P(w|q_2) \tag{1}$$

$$P(w|q_2)) = (1-\lambda)P_{mx}(w|q_2) + \lambda P_{ml}(w|Q) \tag{2}$$

$$P_{mx}(w|q_2) = \alpha \sum_{t \in q_2} P_{tr}(w|t)P_{ml}(t|q_2) + (1-\alpha)P_{ml}(w|q_2) \tag{3}$$

where $P(w|q_2)$, the probability that w is generated from question q_2 and smoothed using $P_{ml}(w|Q)$. $P_{ml}(w|Q)$, the prior probability that w is generated from the question collection Q. λ is the smoothing parameter. $P_{mx}(w|q_2)$ is the interpolated probabilities of $P_{ml}(w|q_2)$ and the sum of the probabilities $P_{tr}(w|t)$, weighted by $P_{ml}(t|q_2)$. P_{ml} is computed using the maximum likelihood estimator. $P(w|t)$ is the translation probability from word t to word w.

However, $P_{TRLM}(q_1|q_2)$ cannot well capture the semantic similarity between two questions q_1 and q_2 because the monolingual translation in cQA is not as strong as in the bilingual translation of SMT. We thus define a more effective method by taking the average of the two scores that switch the role of q_1 and q_2:

$$Score(q_1, q_2) = \frac{1}{2}(P_{TRLM}(q_1|q_2) + P_{TRLM}(q_2|q_1)) \tag{4}$$

We rank the semantically similar questions based on the score of $Score(q_1, q_2)$. For a given question, top-N most similar questions are selected as related questions. Then we rank the target category candidates by counting the corresponding categories of these questions. The ranking is decided by the count of the category. Finally, we can get a list of ranked target category candidates.

4.2 Learning Word Translation Probabilities

The performance of the TRLM will rely on the quality of the word-to-word translation probabilities. We follow the approach of Xue et al. [27] to the learn the word translation probabilities. In our experiments, question-answer pairs are used for training, and the GIZA++[5] toolkit is used to learn the IBM translation model 1. The training process can be accomplished through either as the source and the other as the target.

We employ $P(a|q)$ to denote the word-to-word translation probabilities with question q as the source and answer a as the target, and $P(q|a)$ denotes the opposite configuration. A simple method is to linearly combine the two translation probabilities for a source word and a target word as the final translation probability. Xue et al. [27] find that a better method is to combine the question-answer pairs used for training $P(a|q)$ with the answer-question pairs used for training $P(q|a)$, and to then use this combined set of pairs for learning the word-to-word translation probabilities.

5. ENRICHMENT-BASED CLASSIFICATION STAGE: ENRICHING QUESTIONS WITH WIKIPEDIA SEMANTIC KNOWLEDGE

In the search stage, we reduce the large-scale categories into a much smaller subset of target category candidates. In this Section, we propose a novel method to enrich questions using Wikipedia semantic knowledge to alleviate the data sparseness.

5.1 Wikipedia Thesaurus

Wikipedia is today the largest encyclopedia in the world and surpasses other knowledge bases in its coverage of concepts, rich semantic knowledge and up-to-date content. Recently, Wikipedia has gained a wide interest in IR community and has been used for many problems ranging from document classification [5, 23, 24] to text clustering [7, 8, 9]. Each article in Wikipedia describes a single topic: its title is a succinct, well-formed phrase that resembles a term in a conventional thesaurus [13]. Each article belongs to at least one category, and hyperlinks between articles capture their semantic relations as defined in international standard for thesauri [6]. These semantic relations include: equivalence (synonym), hierarchical relations (hypernym) and associative relation. However, Wikipedia is an open data resource built for human use, so it inevitable includes much noise and the semantic knowledge within it is not suitable for direct use in question classification in cQA. To make it clean and easy-to-use as a thesaurus, we first preprocess the Wikipedia data to collect Wikipedia concepts, and then explicitly derive relationships between Wikipedia based on the structural knowledge of Wikipedia.

5.1.1 Wikipeida Concept

Each article of Wikipedia describes a single topic and its title can be used to represent the concept, e.g., "United States". However, some articles are meaningless – it is only used for Wikipedia management and administration, such as "1980s", "List of newspapers", etc. Following the literature [7], we filter Wikipedia titles according to the rules describing below (titles satisfy one of below will be filtered):

- The article belongs to categories related to chronology, e.g., "Years", "Decades" and "Centuries".
- The first letter is not a capital one.
- The title is a single stopword.

5.1.2 Semantic Relations in Wikipedia

Wikipedia contains rich relation structures, such as synonym (redirect link pages), polysemy (disambiguation page), hypernym (hierarchical relation) and associative relation (internal page link). All these semantic relations express in the form of hyperlinks between Wikipedia articles [13].

Synonym:

[5] http://code.google.com/p/giza-pp/

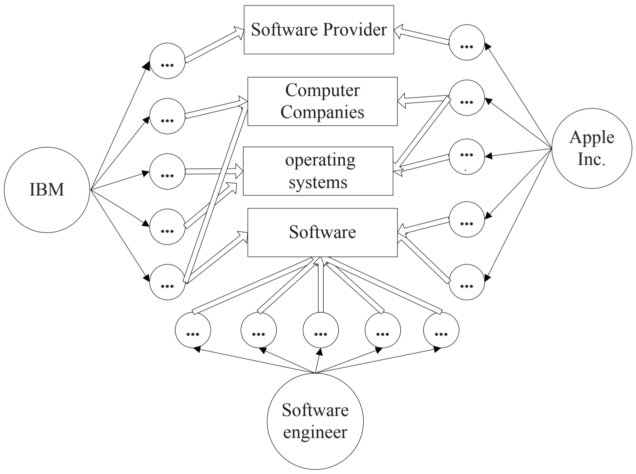

Figure 4: Out-link categories of the concepts "IBM", "Apple Inc." and "Software engineer".

Wikipedia contains only one article for any given concept by using redirect hyperlinks to group equivalent concepts to the preferred one. These redirect links cope with capitalization and spelling variations, abbreviations, synonyms, and colloquialisms. Synonym in Wikipedia mainly comes from these redirect links. For example, "IBM" is an entry with a large number of redirect pages: synonyms (I.B.M, Big blue, IBM Corporation). In addition, Wikipedia articles often mention other concepts, which already have corresponding articles in Wikipedia. The anchor text on each hyperlink may be different with the title of the linked article. Thus, anchor texts can be used as another source of synonym.

Polysemy:

In Wikipedia, disambiguation pages are provided for a polysemous concept. A disambiguation page lists all possible meanings associated with the corresponding concept, where each meaning is discussed in an article. For example, the disambiguation page of the term "IBM" lists 3 associated concepts, including "Inclusion body myositis", "Injection blow molding", and "International Business Machine".

Hypernym:

In Wikipedia, both articles (concepts) and categories belong to at least one category, and categories are nested in a hierarchical organization. The resulting hierarchy is a directed acyclic graph, in which multiple categorization schemes co-exist simultaneously [13]. To extract the real hierarchical relations from Wikipedia categories, we utilize the methods proposed in [18] to derive generic hierarchical relation from category links. Thus, we can get hypernym for each Wikipedia concept.

Associative Relation:

Each Wikipedia article contains a lot of hyperlinks, which express relatedness between them. As Milne et al. [13] mentioned that, links between articles are only tenuously related. For example, comparing the following two links: one from the article "IBM" to the article "Apple Inc.", the other from the article "IBM" to the article "Software engineer". It is clear that the former two articles are more related than the later pair. So how to measure the relatedness of hyperlinks within articles in Wikipedia is an important issue. In this paper, three measures have been introduced [24]: *Content-based*, *Out-link category-based* and *Distance-based*.

Content-based measure is based on the "bag-of-words" representation of Wikipedia articles. Clearly, this measure (denoted as S_{BOW}) has the same limitations of the "bag-of-words" approach since it only considers the words appeared in text documents.

Out-link category-based measure compares the out-link categories of two associative articles. The out-link category of a given article are the categories to which out-link articles from the original one belong. Figure 4 (a fraction of) the out-link categories of the associative concepts "IBM", "Apple Inc", and "Software engineer". The concepts "IBM" and "Apple Inc." share 37 out-link categories; "IBM" and "Software engineer" share 14 out-link categories; "Apple Inc." and "Software engineer" share 12 out-link categories. The larger the number of shared categories, the stronger the associative relation between the articles. To capture the similarity, articles are represented as vectors of out-link categories, where each component corresponds to a category, and the value of the ith component is the number of out-link articles which belong to the ith category. The cosine similarity is then computed between the resulting vectors and denoted as S_{OLC}. The computation of S_{OLC} for the concepts illustrated in Figure 4 gives the following values, which indeed reflect the actual semantic of the corresponding terms: $S_{OLC}(\textit{IBM}, \textit{Apple Inc.}) = 0.517$, $S_{OLC}(\textit{IBM}, \textit{Software engineer}) = 0.236$, $S_{OLC}(\textit{Apple Inc.}, \textit{Software engineer}) = 0.185$.

Distance-based measure captures the length of the shortest path connecting the two categories they belong to, in the acyclic graph of the category taxonomy. This measure is normalized by taking into account the depth of the taxonomy and denoted as D_{cat}.

Following [23], the overall strength of an associative relation between concepts can be written as:

$$S_{overall} = \lambda_1 S_{BOW} + \lambda_2 S_{OLC} + (1 - \lambda_1 - \lambda_2)(1 - D_{cat}) \quad (5)$$

where λ_1 and λ_2 reflect the relative importance of the individual measure. Using equation (5), we rank all the out-linked concepts for each given concept. Then we denote the out-link concepts with relatedness above certain threshold as associative ones for each given concept.

5.2 Mapping Questions into Wikipedia Concept Sets

To use the Wikipedia thesaurus to enrich questions, one of the key issues is how to map words in questions to Wikipedia concepts. Considering frequently occurred synonym, polysemy and hypernym in questions, accurate allocation of words in Wikipedia is really critical in the whole classification process.

Following Hu et al. [7], we build a phrase index which includes the phrases of Wikipedia concepts, their synonym, and polysemy in Wikipedia thesaurus. Based on the generated Wikipedia phrases index, all candidate phrases can be recognized in the web page. We use the Forward Maximum Matching algorithm [25] to search candidate phrases, which is a dictionary-based word segmentation approach. By performing this process, it is necessary to do word sense disambiguation to find its most proper meaning mentioned in questions if a candidate concept is a polysemous one. Wang et al. [24] proposed a disambiguation method based on document similarity and context information, and the implemented method show high disambiguation accuracy. We adopt Wang et al. [24]'s method to do word sense disambiguation for the polysemous concepts in the question.

Figure 5 shows an example of the identified Wikipedia concepts for question Q_1 using the above method. The phrase "software engineer" in Q_1 is mapped into Wikipedia concept "Software engineer", "Big Blue" in Q_1 is mapped into Wikipedia concept "IBM".

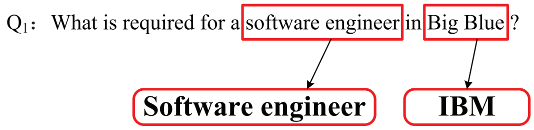

Figure 5: An example of the identified Wikipedia concepts for question Q_1.

5.3 Enriching Questions with Hypernyms

In Wikipedia, each concept belongs to one or more categories. Moreover, these categories are further belongs to more higher level categories, forming an acyclic category graph. The set of categories contained in the category graph of a given concept c is represented as $Cate(c) = \{cate_{c_1}, \cdots, cate_{c_m}\}$. In the category graph, a category may have several paths link to a concept. We calculate the distance $dis(c, cate_i)$ by the length of the shortest path from the concept c to the category $cate_i$.

As noted by Hu et al. [7], the higher level categories have less influence than those lower level categories since the lower level categories are more specific and therefore can depict the articles more accurate. In this paper, we present the influence of categories of γth layer on concept c as $Inf_\gamma(c)$ and define $lnf_1(c) = 1$. For higher levels of categories, we introduce a decay factor $\mu \in [0, 1]$. Thus, we have $lnf_\gamma(c) = \mu lnf_{\gamma-1}(c) = \mu^{\gamma-1} lnf_1(c)$. As each Wikipedia concept has more than one categories, and each category has more than one parent categories, a big γ will introduce too many categories. Therefore, we set $\gamma \leq 3$ in our experiments. Thus, for each concept c we can build a category vector $\mathbf{cate}_c = \{lnf(c, cate_{c_1}), \cdots, lnf(c, cate_{c_m})\}$, where $lnf(c, cate_{c_1}) = lnf_{dis}(c, cate_{c_i}(c))$ which indicates the influence of category $cate_{c_i}$ on concept c. For the collection C which contains all the concepts in question q, the corresponding category vector can be represented as $\mathbf{Cate}_q = \bigcup_{c \in C} \mathbf{cate}_c$.

Figure 6 shows an example of the first three level hypernyms for Wikipedia concept "IBM". For example in level 1, Wikipedia concept "IBM" has parent categories {"Computer hardware companies", "Multinatinal companies", "Cloud computing vendors"}. Thus, for concept c ="IBM", we can build a category vector $\mathbf{cate}_{\text{IBM}}$ = { ("Computer hardware companies", 1), ("Multinatinal companies", 1), $\cdots$, ("Cloud computing vendors", 1), ("Computer companies", 0.5), ("International business", 0.5), $\cdots$, ("Cloud computing", 0.5), ("Technology companies", 0.25), ("International economics", 0.25), $\cdots$, ("Centralized computing", 0.25), $\cdots$ }. Here, we set $\mu = 0.5$, as we tune the parameter on validation data in the experiment. For concept collection C = {"Software engineer", "IBM"} in Q_1, the corresponding category vector is represented as $\mathbf{Cate}_{Q_1} = \mathbf{cate}_{\text{Software engineer}} \bigcup \mathbf{cate}_{\text{IBM}}$.

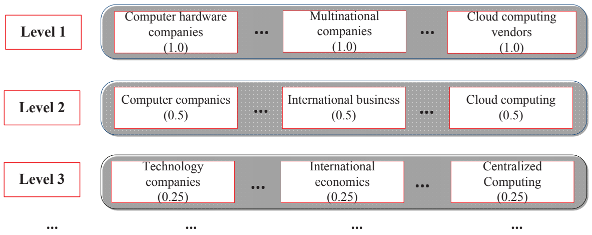
Level 1
Computer hardware companies (1.0)
...
Multinational companies (1.0)
...
Cloud computing vendors (1.0)
Level 2
Computer companies (0.5)
...
International business (0.5)
...
Cloud computing (0.5)
Level 3
Technology companies (0.25)
...
International economics (0.25)
...
Centralized Computing (0.25)
...
...
...
...

Figure 6: An example of the first three level hypernyms for Wikipedia concept "IBM".

5.4 Enriching Questions with Synonymies and Associative Concepts

To better relieve "bag of words" shortcomings, synonyms and associative concepts in Wikipedia can be used to include more related concepts to overcome the data sparseness. For each concept c in Wikipedia, a set of related concepts $\mathbf{rela}_c = \{(c_1, w(c_1, c)), (c_2, w(c_2, c)), \cdots, (c_k, w(c_k, c))\}$ are selected from its synonyms and associative concepts, in which c_k is the kth related concepts of c and $w(c_k, c)$ is the relatedness between c_k and c. The relatedness is defined as follows:

$$w(c_k, c) = \begin{cases} 1 & \text{if } c_k \text{ and } c \text{ are synonyms;} \\ S_{overall} & \text{if } c_k \text{ and } c \text{ are associative relations} \end{cases}$$

where $S_{overall}$ is defined by equation (5). For the collection C which contain all the concepts in question q, the corresponding synonym and associative vector can be represented as $\mathbf{SA}_q = \bigcup_{c \in C} \mathbf{rela}_c$.

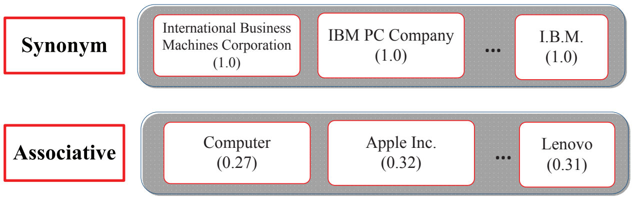

Figure 7: An example of the synonyms and associative concepts for Wikipedia concept "IBM".

Figure 7 gives an example of the synonyms and associative concepts for Wikipedia concept "IBM". For concept "IBM", a set of related concepts $\mathbf{rela}_{\text{IBM}}$ = {("International Business Machines Corporation", 1.0), ("IBM PC Company", 1.0), $\cdots$, ("I.B.M.", 1.0), ("Computer", 0.27), ("Apple Inc.", 0.32), $\cdots$, ("Lenovo", 0.31) }. For concept collection C = {"Software engineer", "IBM"} in Q_1, the corresponding synonym and associative vector can be represented as $\mathbf{SA}_q = \mathbf{rela}_{\text{Software engineer}} \bigcup \mathbf{rela}_{\text{IBM}}$.

5.5 Choosing Machine Learning Method

Many traditional classification methods, such as kNN, Naive Bayes, and more recent advanced models like maximum entropy (MaxEnt), SVMs can be used in our framework. Among them, we choose MaxEnt [1] because of the two main reasons [17]: (1) MaxEnt is robust and has been applied successfully to a wide range of NLP tasks, such as POS tagging, NER and parsing etc. It even performs better than SVMs and others in some special cases, such as classifying sparse data. (2) It is very fast in both training and testing. SVMs is also a good choice because it is powerful. However, the training and testing speed of SVMs are still a challenge to apply to almost real-time applications, especially for multi-classification problem. We train the MaxEnt classifier on the standard integrated data by using limited memory optimization (L-BFGS) [11]. As shown in recent studies, training using L-BFGS gives high performance in terms of speed and classification accuracy. For a given question q, we include the following feature vectors: "Words", "Hypernyms" $\mathbf{Cate}_q$, "Synonyms" and "Associative Concepts" $\mathbf{SA}_q$. An example of the feature vectors for question Q_1 in Figure 5 are shown in Table 1.

Table 1: An example of the feature vectors for question Q_1 in Figure 5.

Words	require(0.019), software(0.031), engineer(0.027), big(0.018), blue(0.022)
Hypernyms	Software engineering(1.0), Software engineers(1.0), Computer hardware companies(1.0), Cloud computing(0.5) International business(0.5), computer companies(0.5) Multinational companies(1.0), Technology companies(0.25) . . .
Synonyms	International Business Machines Corporation(1.0), International Business Machines(1.0), IBM computer(1.0), IBM Corporation(1.0), I.B.M.(1.0), . . .
Associative Concepts	Apple Inc.(0.32), IBM Personal Computer(0.60), Corporation(0.36), Computer science(0.47), software architecture(0.72), . . .

6. EXPERIMENTS

6.1 Data Set

We collect questions from all categories at Yahoo! Answers. We use the *getByCategory* function in Yahoo! Answers API [6] to obtain QA threads from the Yahoo! site. The resulting question repository that we use for question classification contains 2,020,000 questions. Each question consists of three fields: “question", “description" and “answers". For question classification task, we use only the “question" field. There are 26 categories at the first level and 1, 262 categories at the leaf level. Each question belongs to a unique leaf category. Our task is to classify each question into a target leaf category. Table 2 shows the data distribution. Since the whole data set is too large, we select 10,000 questions as testing data. Furthermore, in order to tune the parameters, 10,000 additional questions are also selected as validation data, and the rest (2,000k) are selected as training data. Here, we select the test data and validation data in proportion to the number of questions and categories against the whole distribution to have a better control over a possible imbalance. Especially, the training and testing data are totally exclusive to make sure that the testing data are really difficult to classify. Before the experiments, we make some preprocessing: all the questions are converted into lower case. Each question is tokenized with a stop-word remover[7] and Porter stemming.[8] Besides, we also use one million question-answer pairs from another data set [9] for training the word translation probabilities. We perform a significant test, i.e., a t-test with a default significant level of 0.05 and measure the question classification performance by the accuracy values defined as follows:

$$Accuracy = \frac{\#\text{Correctly classified questions}}{\#\text{Total number of questions}} \tag{6}$$

6.2 Parameter Setting

The experiments use six parameters. The smoothing parameter λ in equation (2); α controls the self-translation impact in the translation-based language model in equation (3); the number of the category candidates returned in the search stage; λ_1 and λ_2 in equation (5); the decay factor μ; Following the literature, we set λ to 0.8 and α to 0.5 [14].

Table 2: Number of questions in each first-level category

Category	#Size	Category	# Size
Arts & Humanities	86,744	Home & Garden	35,029
Business & Finance	86,846	Beauty & Style	37,350
Cars & Transportation	145,515	Pet	54,158
Education & Reference	80,782	Travel	205,283
Entertainment & Music	102,769	Health	132,716
Family & Relationships	34,743	Sports	214,317
Politics & Government	59,787	Social Science	46,415
Pregnancy & Parenting	43,103	Ding out	46,933
Science & Mathematics	89,856	Food & Drink	45,055
Computers & Internet	90,546	News & Events	20,300
Games & Recreation	53,458	Environment	11,276
Consumer Electronics	90,553	Local Businesses	51,551
Society & Culture	94,470	Yahoo! Products	150,445

For the number of the category candidates returned in the search stage, we tune the parameter on the validation data and will show in the experiments. For decay factor μ, we perform an exhaustive grid search of step size 0.1 on [0, 1] to find the parameter on the validation data. To tune the parameters λ_1 and λ_2 in in equation (5), we conduct a method similar to Wang et al. [24]. First, we select 10 Wikipedia concepts randomly, and then extract all the out-linked concepts in the Wikipedia articles corresponding to the 10 concepts. To obtain the ground-truth, three annotators are asked to label all the linked concepts in the 10 articles to three levels (relevant:3, neutral:2, and not relevant:1). The annotating process is carried out independently among annotators. No one among the three annotators could access the annotating results of others. After annotating, each out-linked concept in the 10 articles is labeled with 3 relevance score, and we use the average value as the final value. For example, if one annotator labels two linked concepts as neutral and the other two label them as relevant, the the final score of the two linked concepts is 1.67 ($(1 + 2 + 2)/3$). Based on the labeled data, we can tune the parameters by performing an exhaustive grid search of step size 0.1 on [0, 1] to find the best results. As a result, we set $\lambda_1 = 0.4$ and $\lambda_2 = 0.5$ in the experiment.

6.3 Experimental Results

In this subsection, we conduct several experiments to demonstrate the effectiveness of our proposed method.

6.3.1 The Effect of Search Stage

To demonstrate the effect of search stage, we compare the following two methods:

- *BOW* (baseline): Traditional "bag of words" model with the binary weighting schema. The training process is performed on the entire training data.
- *Search_BOW*: In the search stage, top 8 categories are taken as target category candidates. The parameters are tuned on the validation data shown in subsection 6.4. Then we train the model on the much smaller training data using words as features.

Table 3 shows the experimental results. From the table, we can see that using the search stage can significantly improve the classification accuracy, that is, increasing from an accuracy of 37.75% of the baseline method (*BOW*) to 46.90% (*Search_BOW*) (i.e., eliminating more than 14.47% of error, row 1 vs. row 2). After performing a search stage, we can build a local model on the much smaller training data, thus optimizing the performance for the subset of target categories.

[6] http://developer.yahoo.com/answers/

[7] http://truereader.com/manuals/onix/stopwords1.html

[8] http://www.ling.gu.se/l̃ager/mogul/porter-stemmer/index.html

[9] The Yahoo! Webscope dataset Yahoo answers comprehensive questions and answers version 1.0.2, available at http://reseach.yahoo.com/Academic_Relations.

Table 3: The effect of search stage

#	Methods	Accuracy (%)
1	*BOW*	37.75
2	*Search_BOW*	46.90

6.3.2 The Effect of Enrichment-Based Classification Stage

To demonstrate the effect of enrichment-based classification stage, we compare the following the methods:

- *Search_BOW*: This method is used as baseline, which has been discussed in subsection 6.3.1.
- *Search_BOW_HR*: We train the model on the much smaller training data using words and hypernym concepts as features.
- *Search_BOW_SA*: We train the model on the much smaller training data using words and synonyms and associative concepts as features.
- *Search_BOW_COB*: We train the model on the much smaller training data using words, hypernyms, synonyms and associative concepts as features.[10]

As described in Section 5, in order to reduce the data sparseness, we enrich questions with Wikipedia semantic knowledge. When enriching questions, we first identify the Wikipedia concepts in a question, and then enrich questions with new concepts introduced by the identified concepts. We have considered different strategies: adding hypernyms, adding synonyms and associative concepts. Here we demonstrate the effect of classification performance with questions of adding different kinds of concepts.

Table 4: The effect of adding hypernyms

#	Methods	Strategy	Accuracy (%)
1	*Search_BOW*	-	46.90
2	*Search_BOW_HR*	H_1	49.48
		H_2	48.64
		H_3	46.91
		H_4	45.83
		H_5	44.27

Table 4 demonstrates the performance of question representation with hypernyms. We first add the direct hypernyms (which are category names a candidate concept directly belongs to) for each identified concepts, and then hypernyms of both first and second level (which are parent category names of the direct category a identified concept belongs to), until hypernyms within 5 levels. In Table 4, "H_1" means adding direct hypernyms (first level) into questions; and "H_2" means adding hypernyms of both first and second level, so does for "H_3" to "H_4". Then we find that adding direct hypernyms achieves the best result on question classification, and adding more hypernyms of further levels even deteriorates the classification accuracy. The reason is that the higher level categories have less influence than the lower level categories since lower level categories are more specific and therefore can depict the articles more accurate.

Table 5 shows the result of enriching questions with synonyms and associative concepts. In Table 5, "Synonyms" means adding synonyms into questions, "A_5" means adding 5 most associative concepts into questions, so does for "A_{10}" to "A_{20}". However, out of our expectation, adding synonyms fails to improve the classification accuracy. Since we cannot rank synonyms of a given mapped concepts, we just add all its synonyms into questions, which inevitably brings some noise into questions. We also find that adding 5 most associative concepts brings best performance, whereas adding more associative concepts even decreases the performance.

Table 5: The effect of adding synonyms and associative concepts

#	Methods	Strategy	Accuracy (%)
1	*Search_BOW*	-	46.90
2	*Search_BOW_SA*	Synonyms	43.88
		A_5	50.12
		A_{10}	49.91
		A_{15}	47.11
		A_{20}	46.35

Finally, we try to add both hypernyms and associative concepts together into questions and find that, when adding into questions direct hyponyms and 5 most associative concepts for each mapped concept, this strategy achieve a significant improvement (i.e., eliminating more than 9.98% of error, row 1 vs. row 2), as shown in Table 6.

Table 6: The effect of adding hypernyms and associative concepts

#	Methods	Strategy	Accuracy (%)
1	*Search_BOW*	-	46.90
2	*Search_BOW_COB*	H_1, A_5	52.20

Moreover, in order to demonstrate that question enrichment can effectively alleviate the data sparseness, we do another important experiment by training different classifiers on different sizes of training data ranging from 100k to 2,000k without using the search stage, and measure the accuracy on the test data. We make sure that the training and testing data are totally exclusive so that these data provide very limited context shared information. This makes the test data really difficult to be classified correctly if using traditional "bag of words" model. The results of this experiments are shown in Figure 8, where *BOW_COB* means adding direct hypernyms and 5 most associative concepts into questions without using the search stage, this method also built a global model on the entire training data. There are some clear trends in the result of Figure 8.

First, question enrichment with Wikipedia semantic knowledge can achieve an impressive improvement of accuracy, that is, increasing from an accuracy of 37.75% of the traditional "bag of words" method (*BOW*) to 45.41% (*BOW_COB*) (i.e., eliminating more than 12.31% of errors). This means that question enrichment can greatly reduce the data sparseness. Second, we can achieve a high accuracy even with a small amount of training data. When the sizes of training data changes from 100k to 2,000k, the accuracy with question enrichment changes slightly from 42.11% to 45.41%, while *BOW* accuracy increases nearly 7.33%, from 30.42% to 37.75%.

[10] Following Hu et al. [9], document representation using Wikipedia alone does not perform well, so we do not provide the method in which the model only uses the Wikipedia concepts alone as features for comparison.

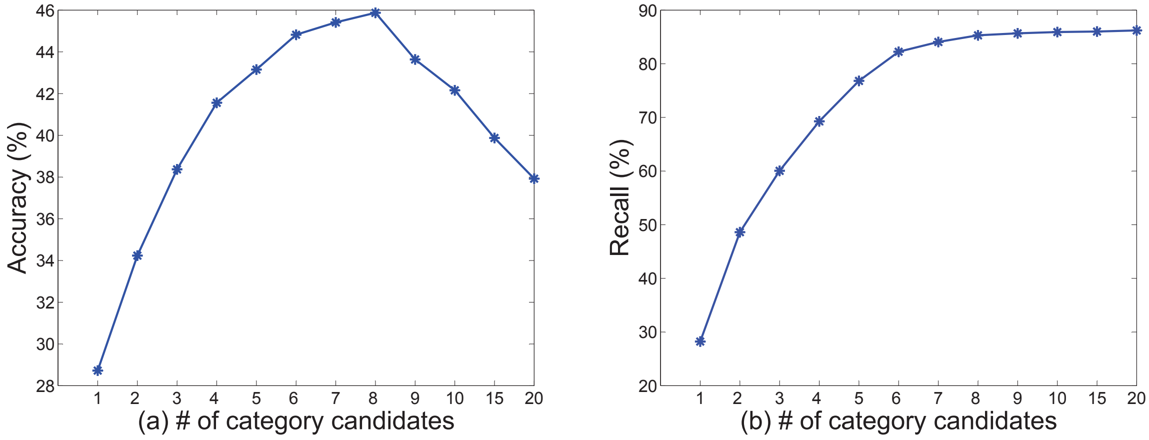

Figure 9: Classification performance and Recall on different number of category candidates on the leaf level.

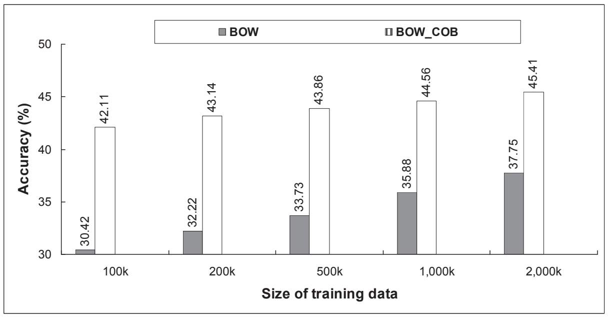

Figure 8: Classification accuracy on test data with different sizes of training data.

6.4 Category Candidate Number Selection

In the search stage, we use the state-of-the-art retrieval model TRLM to return different numbers of category candidates. we try to decide how many top ranked categories to be used to the category candidates are adequate. If we only choose one category, the two-stage method is reduced to the search stage only.

We perform the evaluation on the validation data. Our experimental result is presented in Figure 9. From Figure 9(a), we can see that too many categories also lead to involve in too many training data. As a result, large amount of training data may cause the data to be unbalanced and degrade the performance. From Figure 9(b), the more categories chosen by the search stage, the more likely we can find the correct target category. Here, the recall refers to the ratio of average correct target categories returned for each question using the search stage.

In summary, the performance increases significantly and obtain the best performance about 8 categories. Thus, the number of category candidates is set to 8 considering the trade-off between the recall and the performance. In this paper, we use the top 8 categories as the number of category candidates.

6.5 Running-Time Analysis

In this subsection, we analyze the time complexity of our approach. The learning translation-based language model (TRLM) and question indexing are conducted offline. The time complexity of online computation includes the following four steps: (1) use TRLM to retrieve the top-N most similar questions; (2) process the returned questions to get the category candidates; (3) train the classification model on the much smaller training data using words and Wikipeida concepts as features. (4) employ the classification model to classify the given question using words and Wikipeida concepts as features.

Table 7: Average running-time of each step for testing a given question (seconds)

Retrieval	training the model	classify a given question
0.105s	0.201s	0.017s

Table 7 shows the running-time of each step (averaged over the 10,000 questions in the testing data) on a PC with 8G of memory and a 2.5Ghz CPU of 8 core. We do not show the time for Step 2, since its running-time is negligable as compared to other steps. For example, when processing the returned questions to get the category candidates, the time for each question is about 10^{-5} seconds.

In summary, we observe that the average total time is 0.323 to process each given question. Therefore, the online time complexity is acceptable, and a question can be classified in real time (in the order of 10^{-1}), which indicate that our proposed method is practical and can be handle question classification in cQA efficiently.

7. CONCLUSIONS AND FUTURE WORK

In this paper, we propose a novel method to address the question classification in cQA by leveraging Wikipedia semantic knowledge. To tackle the large-scale category classification in cQA, we extract a small subset of target category candidates related to a given question by performing a search stage, thus the number of the target categories is reduced to a much smaller one. Then we enrich questions by leveraging Wikipedia semantic knowledge in the enrichment-based classification stage. Experimental results show that the classification accuracy of our proposed method significantly outperforms the traditional "bag of words" classification method on the entire large-scale categories (with error reductions of 23.12%).

As a first attempt to address the problem of large-scale category classification and data sparseness in cQA, there are several ways in which our approach might be improved: (1) A natural avenue for further research would be the development of more effective question retrieval methods (e.g., phrase-based translation model [29]) to improve the efficiency of the search stage. (2) We should try to meliorate the effect of adding synonyms by filtering "Redirect" links. After removing useless redirect links, such as spelling variations, adding synonyms into questions will not brings as much noise as before, and its effect could be better. (3) We will try to boost the question enrichment by leveraging YAGO [21] due to its high quality compared to Wikipedia.

8. ACKNOWLEDGMENTS

This work was supported by the National Natural Science Foundation of China (No. 60875041 and No. 61070106). We thank the anonymous reviewers for their insightful comments. We also thank Dr. Maoxi Li and Dr. Jiajun Zhang for suggestion to use the alignment toolkits, Dr. Xianpei Han for suggestion to use the Wikipedia Miner toolkit.

9. REFERENCES

[1] A. Berger, A. Pietra, and J. Pietra. A maximum entropy approach to natural language processing. *Computational Linguistics*, 22(1):39–71, 1996.

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

[3] Y. Cao, H. Duan, C.-Y. Lin, Y. Yu, and H.-W. Hon. Recommending questions using the mdl-based tree cut model. In *WWW*, 2008.

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

[5] E. Gebrilovich and S. Markovitch. Overcoming the brittleness bottleneck using wikipedia: Enhancing text categorication with encyclopedia knowledge. In *IJCAI*, pages 1301–1306, 2006.

[6] A. Hotho, S. Staab, and G. Stumme. Wordnet improves text document clustering. In *SIGIR*, 2003.

[7] J. Hu, L. Fang, Y. Cao, H. Zeng, H. Li, Q. Yang, and Z. Chen. Enhancing text clustering by leveraging wikipedia semantics. In *SIGIR*, 2008.

[8] X. Hu, N. Sun, C. Zhang, and T.-S. Chua. Exploting internal and external semantics for the clustering of short texts using world knowledge. In *CIKM*, 2009.

[9] X. Hu, X. Zhang, C. Lu, E. K. Park, and X. Zhou. Exploiting wikipedia as external knowledge for document clustering. In *KDD*, 2009.

[10] G. Kumaran and J. Allan. Text classification and named entities for new event detection. In *SIGIR*, pages 297–304, 2004.

[11] D. Liu and J. Nocedal. On the limited memory bfgs method for large-scale optimization. *Mathematical Programming*, 45:503–528, 1989.

[12] G. Miller. Wordnet: a lexical database for english. *CACM*, 38:39–41, 1995.

[13] D. Milne, Q. Medelyan, and I. H. Witten. Mining domain-specific thesauri from wikipedia: a case study. In *IEEE/WIC/ACM WI*, 2006.

[14] Z.-Y. Ming, K. Wang, and T.-S. Chua. Prototype hierarchy based clustering for the categorization and navigation of web collection. In *SIGIR*, 2010.

[15] A. Moschitti, S. Quarteroni, R. Basili, and S. Manandhar. Exploiting syntactic and shallow semantic kernels for question/answer classification. In *ACL*, pages 776–783, 2007.

[16] T. Nguyen, L. Nguyen, and A. Shimazu. Using semi-supervised learning for question classification. *Journal of Natural Language Processing*, 15(1):3–21, 2008.

[17] X.-H. Phan, L.-M. Nguyen, and S. Horiguchi. Learning to classify short and sparse txt & web with hidden topics from large-scale data collections. In *WWW*, 2008.

[18] S. P. Ponzetto and M. Strube. Deriving a large scale taxonomy from wikipedia. In *AAAI*, 2007.

[19] F. Sebastiani. Machine learning in automated text categorization. *ACM Computing Surveys*, 34(1):1–47, 2002.

[20] D. Shen, J.-T. Sun, Q. Yang, and Z. Chen. Text classification improved through multigram models. In *CIKM*, 2006.

[21] F. M. Suchanek, G. Kasneci, and G. Weikum. Yago: A core semantic knowledge unifying wordnet and wikipedia. In *WWW*, 2007.

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

[23] P. Wang and C. Domeniconl. Building semantic kernels for text classification using wikipedia. In *KDD*, 2008.

[24] P. Wang, J. Hu, H.-J. Zeng, L. Chen, and Z. Chen. Improving text classification by using encyclopedia knowledge. In *ICDM*, pages 332–341, 2007.

[25] P. Wong and C. Chan. Chinese word segmentation based on maximum matching and word binding force. In *COLING*, 1996.

[26] G.-R. Xue, D. Xing, Q. Yang, and Y. Yu. Deep classification in large-scale text hierarchies. In *SIGIR*, 2008.

[27] X. Xue, J. Jeon, and W. B. Croft. Retrieval models for question and answer archives. In *SIGIR*, 2008.

[28] D. Zhang and W. S. Lee. Question classification using support vector machines. In *SIGIR*, 2003.

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