Boosting Chinese Question Answering with Two Lightweight Methods: ABSPs and SCO-QAT

CHENG-WEI LEE
National Tsing-Hua University Taiwan and Academia Sinica
and
MIN-YUH DAY, CHENG-LUNG SUNG, YI-HSUN LEE,
TIAN-JIAN JIANG, CHIA-WEI WU, CHENG-WEI SHIH,
YU-REN CHEN and WEN-LIAN HSU
Academia Sinica

Question Answering (QA) research has been conducted in many languages. Nearly all the top performing systems use heavy methods that require sophisticated techniques, such as parsers or logic provers. However, such techniques are usually unavailable or unaffordable for under-resourced languages or in resource-limited situations. In this article, we describe how a top-performing Chinese QA system can be designed by using lightweight methods effectively. We propose two lightweight methods, namely the Sum of Co-occurrences of Question and Answer Terms (SCO-QAT) and Alignment-based Surface Patterns (ABSPs). SCO-QAT is a co-occurrence-based answer-ranking method that does not need extra knowledge, word-ignoring heuristic rules, or tools. It calculates co-occurrence scores based on the passage retrieval results. ABSPs are syntactic patterns trained from question-answer pairs with a multiple alignment algorithm. They are used to capture the relations between terms and then use the relations to filter answers. We attribute the success of the ABSPs and SCO-QAT methods to the effective use of local syntactic information and global co-occurrence information.

This research was supported in part by the National Science Council of Taiwan under Center of Excellence Grant No. NSC 95-2752-E-001-001-PAE, the Research Center for Humanities and Social Sciences, Academia Sinica, and the thematic program of Academia Sinica under Grant No. AS 95ASIA02.
We wish to thank the Chinese Knowledge and Information Processing Group (CKIP) in Academia Sinica for providing us with AutoTag for Chinese word segmentation.
Authors' address: Institute of Information Science, Academia Sinica, 128 Academia Road, Section 2, Nankang, Taipei 115, Taiwan; email: {aska, myday, clsung, rog, tmjiang, cwwu, dapi, yrchen, hsu}@iis.sinica.edu.tw.
Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or direct commercial advantage and that copies show this notice on the first page or initial screen of a display along with the full citation. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credits is permitted. To copy otherwise, to republish, to post on servers, to redistribute to lists, or to use any component of this work in other works requires prior specific permission and/or a fee. Permissions may be requested from the Publications Dept., ACM, Inc., 2 Penn Plaza, Suite 701, New York, NY 10121-0701 USA, fax +1 (212) 869-0481, or permission@acm.org.
© 2008 ACM 1530-0226/2008/10-ART12 $5.00 DOI: 10.1145/1450295.1450297.
http://doi.acm.org/10.1145/1450295.1450297.

By using SCO-QAT and ABSPs, we improved the RU-Accuracy[1] of our testbed QA system, ASQA, from 0.445 to 0.535 on the NTCIR-5 dataset. It also achieved the top 0.5 RU-Accuracy[2] on the NTCIR-6 dataset. The result shows that lightweight methods are not only cheaper to implement, but also have the potential to achieve state-of-the-art performances.

Categories and Subject Descriptors: I.2.7 [**Artificial Intelligence**]: Natural Language Processing; H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval

General Terms: Performance, Experimentation

Additional Key Words and Phrases: Chinese question answering, co-occurrence, surface pattern, lightweight method, answer ranking, answer filtering

ACM Reference Format:
Lee, C.-W., Jiang, T.-J., Day, M.-Y., Sung, C.-L., Lee, Y.-H., Wu, C.-W., Shih, C.-W., Chen, Y.-R., and Hsu, W.-L. 2008. Boosting Chinese question answering with two lightweight methods: ABSPs and SCO-QAT. ACM Trans. Asian Lang. Inform. Process. 7, 4, Article 12 (November 2008), 29 pages. DOI = 10.1145/1450295.1450297. http://doi.acm.org/10.1145/1450295.1450297.

1. INTRODUCTION

In recent years, question answering (QA) has become a key research area in several major languages because of the urgent need to deal with the information overload caused by the rapid growth of the Internet. Since 1999, many international question-answering contests have been held at conferences and workshops, such as TREC[3], CLEF[4], and NTCIR[5]. Several top-performing systems have evolved from these contests, for example, the LCC system [Harabagiu et al. 2005] for English at TREC and the QRISTAL system [Laurent et al. 2006] for French at CLEF. The state-of-the-art QA systems in contests usually employ sophisticated techniques to analyze the passages and infer the answer. For example, QRISTAL and the University of Singapore [Cui et al. 2005] use parsers in their systems, and the LCC system uses a logic prover. Both parsers and logic provers are heavy techniques that are either unavailable or of unacceptable quality in some languages. As a result, state-of-the-art QA systems for them are often difficult to implement.

Questions in QA research can be categorized into several types, such as factoid questions, list questions and definition questions, and dealt with by different techniques. In this article, we focus on factoid questions. The answer to a factoid question is a noun or a short phrase, such as a person name, an organization name, a location, a number, time, or an object. For example, "Who is the president of the United States?" is a factoid question asking for a person's name, "What company is South Korea's No. 1 carmaker?" is asking for an organization's name (a company), and "How long is a cow's pregnancy?"

[1]RU-Accuracy is the accuracy of top1 answers regardless of their source documents. For details, please see Section 4.
[2]The RU-Accuracy of our system at NTCIR-6 CLQA was 0.553, of which 0.5 was contributed by ABSPs and SCO-QAT.
[3]Text REtrieval Conference (TREC), http://trec.nist.gov/
[4]Cross-Language Evaluation Forum (CLEF), http://www.clef-campaign.org/
[5]NTCIR (NII Test Collection for IR Systems) Project, http://research.nii.ac.jp/ntcir/

is asking for a period of time. A factoid QA system usually consists of several modules, such as question classification, passage retrieval, answer extraction, filtering, and ranking. We are primarily interested in the last two steps, which involve filtering and ranking answers.

In contrast to most state-of-the-art systems, we focus on lightweight techniques because they would be beneficial for under-resourced languages. Not all languages have such rich or high-quality resources as English. For example, Chinese parsers do not usually perform as well as English parsers due to the word segmentation problem. Therefore, English QA methods that are heavily dependent on parsing may not be effective when applied to Chinese. The situation is worse when we try to deal with regional languages, such as Taiwanese (Minnan) or Cantonese. Lightweight techniques could also be useful in resource-limited situations, such as in resource-restricted hand-held devices where it could be difficult to incorporate sophisticated technologies because of the limited memory, CPU power, and network bandwidth.

We propose two novel lightweight methods that do not require parsers or logic provers, but they still improve QA performance significantly. The first method is the Sum of Co-occurrences of Question and Answer Terms (SCO-QAT), which measures the closeness of an answer and the question keywords by calculating some co-occurrence scores for them. The second method is called Alignment-based Surface Patterns (ABSPs), which automatically generate syntax patterns of relations between terms from question-answer pairs.

SCO-QAT utilizes co-occurrence information. It is similar to Magnini's approach [Magnini et al. 2001], which has been successfully applied to QA as an answer validation mechanism. However, SCO-QAT differs from Magnini's approach in several ways. First, instead of using the whole corpus (or the whole Web), SCO-QAT only uses the retrieved passages to calculate a co-occurrence score, which is suitable when access to the whole corpus is restricted due to cost or bandwidth. For example, in a wireless situation with limited bandwidth, if a method needs to query a corpus several times, the system's response time might deteriorate to the extent that it would be unacceptable to users. Second, Magnini's approach needs manually created word-ignoring rules to deal with situations when the required statistics are unavailable. The rules may vary depending on the question and need to be adjusted when the domain or language changes. SCO-QAT resolves the problem by calculating all combinations of the co-occurrence scores for the answer and the question keywords.

Surface patterns are syntactic patterns that connect answers and question keywords. For example, Ravichandran and Hovy [Ravichandran and Hovy 2002] use patterns such as "<NAME> was born on <DATE>" and "<NAME> (<BIRTHDATE>-" to answer BIRTHDATE questions (When was X born?). Although these surface patterns are simple and accurate, four issues need to be addressed. First, more fine-grained question types must be defined. For example, in addition to using a DATE question type, we may need more date-related question types, such as BIRTHDATE, BUILTDATE, ... etc. However, that would increase the burden on the Question Classification module. Second, the method cannot deal with questions that have multiple keywords. Third, the method cannot handle cases where the information for validating

the answer is spread over several passages. Fourth, since it requires exact matches, the method cannot be applied when there is a high language variation. Our proposed surface pattern method, ABSP, can resolve the first three problems. ABSPs are generated from question-answer pairs regardless of the question type. In situations involving multiple question keywords and multiple passages, several ABSPs are used together to calculate a score for an answer.

We show that it is possible to use lightweight techniques to boost QA system performance. To achieve our goal, we employ two novel lightweight methods, SCO-QAT and ABSPs, in a Chinese QA system, which have significantly increased the RU-Accuracy of the testbed QA system, ASQA, from 0.445 to 0.535 on the NTCIR-5 CLQA dataset. It also achieved the top RU-Accuracy 0.5 on the NTCIR-6 dataset. The result shows that lightweight methods are not only cheaper to implement, but also have the potential to achieve state-of-the-art performances. In summary, we improve a Chinese QA system by employing two novel lightweight methods that do not require heavy techniques like parsing or logic provers.

The remainder of the article is organized as follows. We review related works in Section 2, and introduce the host QA system in Section 3. Our proposed methods, SCO-QAT and ABSPs, are presented in Section 4. We describe the datasets and evaluation metrics used in our experiments in Section 5, and detail the experiments in Section 6. Section 7 contains a discussion. Then, in Section 8, we summarize our conclusions and consider the direction of our future work.

2. RELATED WORKS

2.1 QA with Surface Patterns

Surface patterns have been successfully applied in a number of QA systems. For example, Ravichandran and Hovy [Ravichandran and Hovy 2001] proposed a surface text pattern for extracting answers, while Muslea [Muslea 1999] employed three different linguistic patterns to extract relevant information. Soubbotin and Soubbotin [Soubbotin and Soubbotin 2001] used richer patterns (including predefined string sequences, unordered combinations of strings, and definition patterns) to answer questions and won the TREC-2001 competition. However, since none of the above patterns include semantic information, they are called "poor-knowledge approaches" [Saiz-Noeda et al. 2001].

There has been some progress in adding semantic representations to surface patterns to improve the coverage of questions that the surface patterns can be applied to. Saiz-Noeda et al. [Saiz-Noeda et al. 2001] proposed a type of semantic pattern that uses EuroWordNet as a lexical database, but it cannot represent the constraints on a specific part of a sentence. Staab et al. [Staab et al. 2001] proposed another kind of semantic pattern that can be used for communications between semantic Web developers, as well as for mapping and reusing different target languages. However, since it is designed primarily for professional use, it is difficult to implement without domain knowledge.

A substantial amount of research has focused on paraphrasing [Barzilay and Lee 2003]. Bouma et al. [2005] use syntactic information about paraphrases to solve QA with a dependency parser. Takahashi et al. [2004] developed a system for QAC2, which uses paraphrasing to perform greedy answer seeking. However, the approach is not efficient because it is intended for structural matching-based answering, which needs large-scale paraphrase patterns.

As mentioned in the the Introduction, four issues related to Ravichandran and Hovy [2001] surface pattern method need to be solved. First, finer-grained question types have to be defined. Second, it cannot deal with questions containing multiple keywords. Third, it cannot cover cases where the information for validating the answer is spread over several passages. Fourth, it requires an exact match, so it cannot be applied when there is a high language variation. Our proposed surface pattern method, ABSPs, deals with the first three problems to a certain extent. ABSPs do not need to define additional question types because they are generated from question-answer pairs regardless of the question type. For cases involving multiple question keywords or where the information is spread over multiple passages, we use several ABSPs and combine all the information to calculate a score for an answer.

2.2 QA with Co-occurrence Information

Clarke et al. [2001] suggested that redundancy could be used as a substitute for deep analysis because critical information may be duplicated many times in high-ranking passages. Several systems [Clarke et al. 2002; Cooper and Ruger 2000; Kwok and Deng 2006; Lin et al. 2005; Zhao et al. 2005; Zheng 2005] incorporate answer frequency, which is redundant answer information, in their answer ranking components. However, using this feature alone would be insufficient for some questions. Magnini et al. [2001] consider that the number of documents in which the question terms and the answer co-occur is useful for QA. The hypothesis is similar to that of Clarke et al. [2001], who use co-occurrence methods to measure the relevance of an answer to the given question based on Web search results. As the co-occurrence information tends to be unreliable when the co-occurrence count is too small, Magnini et al. apply some word-ignoring rules to reduce the number of question keywords when the number of returned documents is less than a certain threshold.

Magnini et al.'s approach is not applicable to some QA scenarios because it requires a large number of queries for a question. For example, given a QA system with an average number of 40 answers for a question, it will require more than 40 queries to the search engine for each question. Therefore, it is difficult to respond to a question within a reasonable time. Moreover, search engines usually do not allow a large number of queries in a short period of time.

To cope with such resource limited situations, we developed a novel method called SCO-QAT, which is based on the same assumption as Magnini et al.'s hypothesis. However, instead of querying the Web multiple times, SCO-QAT

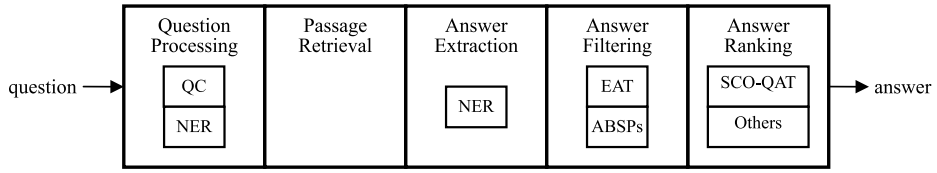

Fig. 1. System architecture of ASQA for Chinese-Chinese Factoid QA.

relies on retrieved passages solely; therefore, it does not need any word-ignoring rules.

3. THE HOST QA SYSTEM: ASQA

Experiments in this article were conducted on a host QA system, ASQA ("Academia Sinica Question Answering system"[6]), which we developed to deal with Chinese related QA tasks. The system participated in the CLQA C-C (Chinese-to-Chinese) subtasks at NTCIR-5 and NTCIR-6, and achieved state-of-the-art performances in both cases. The architecture of ASQA consists of five modules as shown in Figure 1. Questions are first analyzed by the question processing module to get keywords, named entities (NEs), and the question type. Then, queries are constructed for passage retrieval according to the question processing results. In the next phase, answer extraction is performed on the retrieved passages to obtain candidate answers, which are then filtered and ranked by the answer filtering module and answer ranking module, respectively.

3.1 Question Processing

As Chinese written texts do not contain word delimiters, we incorporate a Chinese segmentation tool to break a question into question segments comprised of words and parts-of-speech (POS). With these question segments and other information, such as HowNet[7] sense, ASQA can identify six coarse-grained question types (PERSON, LOCATION, ORGANIZATION, ARTIFACT, TIME, and NUMBER) and 62 fine-grained question types. ASQA adopts an integrated knowledge-based and machine learning approach for Chinese question classification.

We use InfoMap [Hsu et al. 2001] as the knowledge-based approach, which uses syntactic rules to model Chinese questions, and adopt SVM (Support Vector Machines) [Vapnik 1995] as the machine learning approach for a large collection of labeled Chinese questions. Each question is classified into a question type or types by InfoMap and the SVM module. Then, the integrated module selects the question type with the highest confidence score. A detailed description of our question classification scheme can be found in Day et al. [2005].

[6]ASQA demo site, http://asqa.iis.sinica.edu.tw/

[7]HowNet (http://www.keenage.com/) is a common-sense knowledge base in which the sense or meaning of a word comprises one or several sememes, the basic unit of meaning.

3.2 Passage Retrieval

ASQA splits documents into sentences and indexes them with Lucene[8], an open source information retrieval engine. Two indices are used in ASQA: one based on Chinese characters and the other on Chinese words. At runtime, ASQA utilizes the question segments and POS to form Lucene queries. Query terms are weighted according to their POS. Two Lucene queries are constructed for each question. In the initial query, quoted terms and nouns are set as *required*[9]. If this query does not return enough passages, we retry a relaxed version of the query that does not assign any query term as *required*. For each question, the top 100 passages are chosen for answer extraction.

3.3 Answer Extraction

To identify both coarse-grained and fine-grained candidate answers, ASQA uses a coarse-grained Chinese NER (Named Entity Recognition) engine [Wu et al. 2006] combined with a fine-grained taxonomy and rules. The coarse-grained NER engine, which can identify person names, organization names, and locations, ensembles several CRF (Conditional Random Fields) models with character and word features. The taxonomy and rules are compiled manually from several resources to identify other coarse-grained and fine-grained NEs.

3.4 Answer Filtering

In the next step, ASQA applies answer filters to reduce the number of candidate answers. There are two filters, the EAT (Expected Answer Type) Filter and the ABSP Filter. The EAT Filter screens candidate answers according to their types and the question type, using a mapping table containing information about question types and their corresponding expected answer types. Answers whose types are not found among the expected answer types are removed, and the remaining are the answer candidates. More information about expected answer types can be found in Day et al. [2005]. We discuss the ABSP Filter in detail in Section 4.1.

3.5 Answer Ranking

An answer ranking score is calculated for each answer by combining several features as a weighted sum:

$$RankingScore(Ans) = \sum_{i=1}^{|features|} w_i \times feature_i(Ans),$$

[8]Lucene, http://lucene.apache.org/
[9]Use Lucene's "+" operator.

where *Ans* denotes the answer; and w_i and $feature_i(Ans)$ denote, respectively, the weight and score of the ith feature. All the weights are determined by a genetic algorithm with a training dataset. We tested the answer-ranking formula on different feature combinations. Note that the SCO-QAT feature discussed in this article contributed the most to our NTCIR-6 performance. The other ranking features considered in NTCIR-6 were described in Lee et al. [2005; 2007].

4. PROPOSED METHODS

4.1 ABSPs—Alignment-Based Surface Patterns

In ASQA, ABSPs are used in an answer filter to confidently identify correct answers. Next, we introduce the alignment algorithm and describe the generation process, which involves the following steps: 1) generate ABSPs by multiple sequence alignment, 2) select ABSPs based on a set of question-answer pairs, 3) apply ABSPs and combine extracted relations, and 4) calculate the scores.

4.1.1 *The Alignment Algorithm.* Sequence alignment is the process that finds similar sequences in a pair of sentences. Pair-wise sequence alignment (PSA) algorithms that generate templates and match them against new text have been researched extensively. Huang et al. [2004] employ a PSA algorithm to generate patterns for extracting protein-protein interaction descriptions from biomedical texts annotated with part-of-speech (POS) tags. The sequences are padded with gaps so that similar characters can be aligned as closely as possible. Because we need surface patterns extracted from sentences that have certain morphological similarities, we employ local alignment techniques [Smith and Waternman 1981] to generate surface patterns.

To apply the alignment algorithm, we first perform word segmentation. In the following discussion each unit is a word. Our templates contain named entity (NE) as semantic tag, and POS as syntactic tag. Consider two sequences $X = (x_1, x_2, \ldots, x_n)$ and $Y = (y_1, y_2, \ldots, y_m)$ defined over the alphabet $\sum$ that consists of four kinds of tags: NE tags, POS tags, a raw word tag for every single word, and a tag "-" for a gap. We assign a scoring function, F, to measure the similarity of X and Y. $F(i, j)$ is defined as the score of the optimal alignment between the initial segment from x_1 to x_i of X and the initial segment from y_1 to y_j of Y.

$F(i, j)$ is recursively calculated as follows:

$$F(i, 0) = 0, F(0, j) = 0, \text{xi}, \text{yj} \in \Sigma, \tag{1a}$$

$$F(i, j) = \max \begin{cases} 0, \\ F(i-1, j-1) + d(xi, yj) \\ F(i-1, j) + d(xi, '-') \\ F(i, j-1) + d('-', yj) \end{cases}, \tag{1b}$$

where $d(a, b)$ is the function that determines the degree of similarity between two alphabet letters a and b. The function is defined as

$$d(a, b) = \max \begin{cases} 1, a = b \\ 1, NE(a) = NE(b) \\ 1, POS(a) = POS(b) \\ 1 - penalty, POS(a) \approx POS(b) \\ 0, a \neq b \end{cases}, \quad (2)$$

where *NE(a)* denotes the Named Entity (NE) tag of a, and *POS(a)* denotes POS tag of a. If the POS tags of a and b are different, but they have a common prefix, the degree of similarity is subtracted with a penalty.

For a sequence X of length n and a sequence Y of length m, totally $(n + 1)$* $(m + 1)$ scores are calculated by applying Equations 1a and 1b recursively. The scores are stored in a matrix $F = F(x_i, y_j)$, and the optimal local alignment can be found by back-tracking in F.

4.1.2 *ABSP Generation.* An ABSP is composed of ordered slots. For example, there are five slots in the ABSP "ORGANIZATION Na - NE 表示". This case demonstrates that a slot in an ABSP could be a semantic tag (PERSON, ORGANIZATION, LOCATION, TIME, and OCCUPATION), a POS tag, a term or a gap (which indicates that position can be any word). We generate ABSPs from a set of sentences by applying the alignment algorithm. Before alignment, the sentences are segmented and tagged with POS by a Chinese segmentation tool, AutoTag[10]. In addition, we tag the sentences with semantic tags. We use an NER engine to label PERSON, ORGANIZATION, LOCATION, and TIME tags, and a word list for "occupation" tags. After this step, the remaining words without any semantic tag are tagged "O". Thus, every segment of a sentence contains a word, a POS tag and a semantic tag in the format: "word/POS tag/semantic tag". For example, the sentence "2000年奧運在雪梨舉行" would be preprocessed into "2000年/Nd/TIME 奧運/Nb/O 在/P/O 雪梨/Nc/LOCATION 舉行/VC/O"[11].

Using the proposed alignment algorithm, our ABSP generation algorithm extracts general patterns of all three types of tags. We begin by pairing all sentences based on their similarity. Closely matched pairs are then aligned and a pattern that fits both pairs is created. We choose slots according to the corresponding parts of the aligned sentence pair with the following priority: *word* > *Semantic tag* > *POS tag*. If the sentences for a given slot have nothing in common, the algorithm creates a gap ("−") in that position. Table I shows an aligned pair of sentences. In this case, the algorithm generates the pattern "V − N Na 的 LOCATION Na 是 PERSON," which means a verb followed by a gap, two nouns, a word "的", a location, a noun, a word "是", and

[10]CKIP AutoTag: http://rocling.iis.sinica.edu.tw/CKIP/wordsegment.htm. Although AutoTag has 46 kinds of POS tags, we only use the first two letters of the tags. For example, both "Caa" and "Cab" are treated as "Ca". Details of these POS tags can be found at http://rocling.iis.sinica.edu.tw/CKIP/paper/poslist.pdf

[11]Nd means temporal noun, Nb means proper noun, P means preposition, Nc means location noun, and VC means action transitive verb.

Table I. Example of an Aligned Sentence Pair with the Resulting ABSP: A Verb Followed by a Gap, Two Nouns, a Word "的", a Location, a Noun, a Word "的", and a Person

榮獲/VJ/O		諾貝爾/ Nb/ORG	和平獎/ Na/O	的/DE/O	南韓/ Nc/LOC	總統/ Na/OCC	是/SHI/O	金大中/ Nb/PER
參加/VC/O	2000 年/ Nd/TIME	兩韓/ Nc/LOC	高峰會/Na/O	的/DE/O	北韓/ Nc/LOC	領導人/Na/O	是/SHI/O	金正日/ Nb/PER
V	–	N	Na	的	LOC	Na	是	PER

a person. The pattern is generated in this way because, in the first and third positions, the aligned pairs have the same common prefix for POS tag "V, N;" in the second position, they have nothing in common, thus resulting in a generalized gap, "−;" in the fourth and seventh positions, they have the same POS tag "Na;" in the fifth and eighth positions, they have the same words "的, 是;" and in the sixth and ninth positions, they have the same semantic tag "LOCATION, PERSON." The complete ABSP generation algorithm is detailed in Algorithm 1.

Algorithm 1 ABSP Generation

Input: Question set $S = \{s_1,...,s_n\}$,
Output: A set of uncategorized ABSPs $T = \{t_1,...,t_k\}$.
Comment: perform pair alignment for every two questions

```
T = {};
for each question s_i from s_1 to s_{n-1} do
  for each question s_j from s_i to s_n do
    perform alignment on s_i and s_j, then
      pair segments according to similarity matrix F;
    generate a common ABSP t from the aligned pairs with the maximum
    similarity;
    T ← T∪t;
  end;
end;
return T;
```

4.1.3 *ABSPs Selection.* The selection process chooses patterns that can connect question keywords and the answer. It is assumed that useful patterns usually contain *important tags*. We deem all the NE Tags, the Nb[12] the POS tag, and all the verb POS tags as *important tags*, because, based on our observations, these tags usually associate with question keywords and answers. We define pattern slots with important tags as *important slots* and question terms with important tags as *important terms*. For example, there are three *important slots* ("V", "LOCATION", and "PERSON") in the pattern "V − N Na 的 LOCATION Na 是 PERSON," and four *important terms* ("榮獲", "諾貝爾", "南韓", and "總統") in the question "榮獲/VJ/O 諾貝爾/Nb/ORGANIZATION 和平獎/Na/O 的/DE/O 南韓/Nc/LOCATION 總統/Na/OCCUPATION 是/SHI/O 誰/Nh/O".

[12]In the CKIP POS tag set, Nb means "proper noun".

We apply each generated ABSP to its source passages. When a matched source passage is found, we extract the corresponding terms from the important slots. If the extracted terms do not contain the answer and any of the important terms of the source question, the ABSP is removed. In our experiment, we collected 126 useful ABSPs from the 865 training questions. The detail is described in Algorithm 2.

Algorithm 2 ABSPs Selection

Input: A set of ABSPs $T = \{t_1,...,t_k\}$ for selection, the source question Q, the answer A, the source passages $S = \{s_1,...,s_n\}$.
Output: Selected set of ABSPs $T' = \{t_1,...,t_l\}$.

```
1: T' = {};
2: QTs ←extract important terms from Q
3: for each sentence s_i in S do
4:   for each ABSP t_j in T do
5:     perform pattern matching on s_i with t_i, if match then
6:     PTs ←extract terms that match with important slots of t_j from s_i
7:     if PTs contains A and any term in QTs then
8:         T' ← T' ∪ t_j;
9:       end if;
10:    end if;
11:  end;
12: end;
13: return T';
```

4.1.4 *Relation Extraction and Score Calculation.* We are now ready to apply the selected ABSPs in the QA system. In this work, we applied ABSPs as a filter to choose highly confident answers. We assume words matched by an ABSP have certain relations between them. For example, the pattern "<PERSON> was born on <DATE>" relates a person to his/her birth date. When a pattern matches the words, a relation is identified and we construct a Related-Terms-Set (RTS) which contains the related terms. By using ABSPs for matching, we are able to find and combine the RTSs of the question keywords and answers in passages. We calculate a score for each candidate answer according to these RTSs.

Given the passages retrieved for a question, all the ABSPs are applied to each passage. If an ABSP matches a passage, we extract an RTS, which is comprised of the matched important terms (i.e., we discard terms that do not have an 'Nb' tag, an NE tag, or a verb). More RTS are constructed if more than one ABSP matches different terms in a passage. If the RTS contains common elements (i.e., the same term is matched by at least two ABSPs,) we check the *idf* values of those elements. If one *idf* value is higher than a threshold value, the two RTSs are merged, as shown by the example in Table II. The table contains a question, two passages retrieved from a corpus, and two ABSPs that match the two passages. The first ABSP, $ABSP_1$, extracts RTS_1 {奪得/VC, 奧斯卡/Nb, 女配角/OCC} from $Passage_1$, while $ABSP_2$ extracts the terms "蜜拉索維諾/PER", "非強力春藥/ART", "獲/VJ", "奧斯卡/Nb" and forms RTS_2. Since

Table II. An Example of Relation Extraction

Questions:	女演員/OCC 蜜拉索維諾/PER 獲得/VJ 奧斯卡/Nb/ORG 最佳/A 女配角/OCC 獎/Na 是/SHI 因/Cbb 哪/Nep 部/Nf 電影/Na
$ABSP_1$:	*VC* Neu *Nb* A *OCC* – Na
$Passage_1$:	..... 而/Cbb 奪得/VC 一九九五.../Neu 奧斯卡/Nb 最佳/A 女配角/OCC 的/DE 殊榮/Na ...
RTS_1:	{奪得/VC, 奧斯卡/Nb, 女配角/OCC}
$ABSP_2$:	*PER P PAR ART PAR – DE Na X VJ Nb*
$Passage_2$:	... 蜜拉索維諾/PER 在/O/P/O 「/O/PAR 非強力春藥/ART 」/PAR 中/Ncd ..... 獲/VJ 奧斯卡/Nb 獎/Na ...
RTS_2:	{蜜拉索維諾/PER, 非強力春藥/ART, 獲/VJ, 奧斯卡/Nb}
Merged RTS:	{奪得/VC, 奧斯卡/Nb, 女配角/OCC, 蜜拉索維諾/PER, 非強力春藥/ART, 獲/VJ }

"奧斯卡" already exists in RTS_1, we examine the *idf* value of "奧斯卡" and merge it with RTS_1 to form a new RTS (the Merged RTS).

After all the RTSs for the given question have been constructed, we use the question's important terms (女演員, 蜜拉索維諾, 獲得, 奧斯卡, 女配角, in this example) to calculate an RTS score. The score is calculated as the ratio of the question's important terms to the matched important terms. In this case, the number of the question's important terms is five, and the number of matched important terms is three. Therefore, the score of answers belonging to this RTS is 3/5. For RTSs that do not contain any of the question's important terms, we discard the candidate answers they contain. If none of the RTSs contains a question's important terms, we say the question is not covered; and since we cannot find useful relations for filtering answers, we retain all the answers. After processing all the sentences selected for a question, we rank the candidate answers by the sum of their RTS scores for the sentences in which they appear and retain the top-ranked answer(s).

Algorithm 3 RTS construction for a question

Input: One question Q, Sentences for each question $S = \{s_1,...,s_n\}$, ABSPs $T = \{t_1,...,t_k\}$, *idf* thresholdσ
Output: RTSs $R = \{r_1,...,r_k\}$.

```
1: R = {};
2: for each sentence s in s_i do
3:   for each ABSP t_i from t_1 to t_k do
4:     perform pattern matching on s with t_i,
5:     if match then
6:       r' ← matched important terms (it)
7:       for each RTS r in R do
8:         if r and r' has common it and idf(it) > σ then
9:           r ← r ∪ r';
10:        else
11:          R ← R ∪ r';
12:        end;
13:      end;
14:    end;
15:  end;
16: end;
17: return R;
```

4.2 SCO-QAT: Sum of Co-occurrences of Question and Answer Terms

The basic assumption of SCO-QAT is that, in good quality passages, the more often an answer co-occurs with the question terms, the higher the confidence that the answer will be correct. We regard a co-occurrence as an indication that the answer can be inferred as correct based on the co-occurring question terms. Passages are chosen instead of documents, because we assume that co-occurrence information provided by passages is more reliable. We formulate our concept as an expected confidence score from which the SCO-QAT formula is deduced.

Let the given answer be A and the given question be Q, where Q consists of a set QT of question terms $\{qt_1, qt_2, qt_3, \ldots\ldots, qt_n\}$. The question terms are created from the word segmentation result of Q with some stop words removed (the stop word list is provided in Appendix B.: Stop Word List for SCO-QAT). Based on QT, we define QC as a set of question term combinations, or more precisely $QC = \{qc_i \mid qc_i \text{ is a subset of } QT \text{ and } qc_i \text{ is not empty}\}$. The co-occurrence confidence score of an answer A with a question term combination qc_i is calculated as follows:

$$Conf(qc_i, A) = \begin{cases} \dfrac{freq(qc_i, A)}{freq(qc_i)} & , \text{ if } freq(qc_i) \neq 0 \\ 0 & , \text{ if } freq(qc_i) = 0 \end{cases}, \tag{3}$$

where *freq(X)* is the number of retrieved passages in which all the elements of X co-occur. We assume that all question term combinations have an equal chance of being used to verify the answer's correctness. Therefore, the expected confidence score is defined as

$$\sum_{i=1}^{|QC|} \frac{1}{|QC|} Conf(qc_i, A) = \frac{1}{|QC|} \sum_{i=1}^{|QC|} Conf(qc_i, A). \tag{4}$$

Because $|QC|$ is the same for every answer, it can be removed. As a result, we have the following formula for SCO-QAT:

$$SCO\text{-}QAT(A) = \sum_{i=1}^{|QC|} Conf(qc_i, A). \tag{5}$$

We rank candidate answers according to their SCO-QAT scores. For example, given a question Q consisting of three question terms {qt1, qt2, qt3} and a corresponding answer set with two candidate answers {c1, c2}, the retrieved passages are presented as

P1: *qt1 qt2 c2*
P2: *qt1 qt2 qt3 c1*
P3: *qt1 qt2 c1*
P4: *qt1 c2*
P5: *qt2 c2*
P6: qt1 qt3 c1.

We use Equation (5) to calculate the candidate answer's SCO-QAT score as follows:

$$\begin{aligned} SCO\text{-}QAT(c1) &= \frac{freq(qt1,c1)}{freq(qt1)} + \frac{freq(qt2,c1)}{freq(qt2)} + \frac{freq(qt3,c1)}{freq(qt3)} + \frac{freq(qt1,qt2,c1)}{freq(qt1,qt2)} \\ &\quad + \frac{freq(qt1,qt3,c1)}{freq(qt1,qt3)} + \frac{freq(qt2,qt3,c1)}{freq(qt2,qt3)} + \frac{freq(qt1,qt2,qt3,c1)}{freq(qt1,qt2,qt3)} \\ &= \frac{3}{5} + \frac{2}{4} + \frac{2}{2} + \frac{2}{3} + \frac{2}{2} + \frac{1}{1} + \frac{1}{1} = 5.77 \\ SCO\text{-}QAT(c2) &= \frac{2}{5} + \frac{2}{4} + \frac{0}{2} + \frac{1}{3} + \frac{0}{2} + \frac{0}{1} + \frac{0}{1} = 1.23 \quad . \end{aligned}$$

Since the SCO-QAT score of c1 is higher than that of c2, c1 is considered a better answer candidate than c2.

4.3 Enhancing SCO-QAT with Distance Information

Generally speaking, co-occurrence information for QA tends to be unreliable when questions are short and the distance between co-occurring terms is large. We encountered some failures caused by these issues. For example, given the question "Who is the president of United States of America?" consisting of two question terms {president, United States of America } and a corresponding answer set {Bush, Chen Shui-bian}, the retrieved passages were as follows:

P1: *Taiwan president* ***Chen Shui-bian*** *thought that it is not a big . . . in* ***United States of America***.

P2: ***G. W. Bush***, *the 43rd President of the* ***United States of America*** *said . . .*

SCO-QAT cannot determine whether "G. W. Bush" or "Chen Shui-bian" is the correct answer because they have the same SCO-QAT score. However, intuitively, "G. W. Bush" is closer to the question term "United States of America" than "Chen Shui-bian," so we should consider the distance information to resolve the dilemma.

We enhance SCO-QAT by incorporating distance information to obtain the term density in passages when the number of question terms is small. Density is a stricter criterion for deciding the co-occurrence confidence in one passage than *freq(X),* which is used in the original version of SCO-QAT [Equation (3)]. The following is the extended SCO-QAT formula:

$$Conf\_dist(qc_i, A) = \begin{cases} \dfrac{1}{freq(qc_i)} \displaystyle\sum_{j=1}^{n} \dfrac{1}{avgdist(p_j, qc_i, A)} & , \text{ if } freq(qc_i) \neq 0 \\ 0 & , \text{ if } freq(qc_i) = 0 \end{cases} , \quad (6)$$

$$SCO\text{-}QAT\_with\_Distance(A) = \begin{cases} \displaystyle\sum_{i=1}^{|QC|} Conf(qc_i, A) & , \text{ if } |QT| > threshold \\ \displaystyle\sum_{i=1}^{|QC|} Conf\_dist(qc_i, A) & , \text{ if } |QT| < threshold \end{cases} , \quad (7)$$

Table III. Datasets for the Experiments in this Article. The Datasets Created by NTCIR Also Have Corresponding Expanded Datasets, Which Contain Extra Answers for Post-Hoc Experiments. We Postfix the Original Name with the Letter "e" to Indicate the Expanded Dataset Name

dataset	corpus	creator	# of questions
NTCIR5-CC-D200 NTCIR5-CC-D200e	CIRB40	NTCIR	200
NTCIR5-CC-T200 NTCIR5-CC-T200e	CIRB40	NTCIR	200
NTCIR6-CC-T150 NTCIR6-CC-T150e	CIRB20	NTCIR	150
IASL-CC-Q465	CIRB40	Academia Sinica	465
Total # of questions			1015

where n denotes the number of retrieved passages. If the passage does not contain qc_i, we set the confidence value to 0. As shown in the modified SCO-QAT function in Equation (7), we only switch to *Conf_dist* when the number of question terms is smaller then a threshold. The *avgdist* function is the average number of characters between the question term combination qc_i and the answer A in passage p_j, which is calculated as:

$$avgdist(p_j, qc_i, A) = \frac{\sum_{k \in qc_i} dist(p_j, k, A)}{|qc_i|}.$$

The *dist* function is the character-distance between the question term k and the answer A in passage p_j. If the passage does not contain k, it returns 10.

5. EVALUATION SETUP

To increase the confidence of our experiments, we created new datasets and introduced a new metric, called the Expected Answer Accuracy (EAA), to compare performances when ranking several top answers that have the same score.

5.1 Datasets

We experiment on several new QA datasets, some of which were expanded from NTCIR CLQA, while others were created by us. A QA dataset (the gold standard) is defined as a set of questions, their answers, and the document IDs of supporting documents. For the CLQA Chinese-Chinese (CC) subtask, we use three datasets from NTCIR-5 and NTCIR-6, denoted as NTCIR5-CC-D200, NTCIR5-CC-T200, and NTCIR6-CC-T150 in this article. The last item of a dataset name indicates the number of questions and the dataset's purpose, where T stands for "test" and D stands for "development" (Table III).

According to Lin et al. [2005], datasets created by QA evaluation forums are not suitable for post-hoc evaluations because the gold standard is not sufficiently comprehensive. This means we have to manually check all the extra answers not covered by the gold standard in order to derive more reliable experiment results. Since the number of questions in our experiments is quite

large, it is not feasible to examine all the extra answers and their supporting documents. Therefore, we use RU-accuracy, which is described in Section 5.2, to compare performances so that we do not have to check all the returned documents; only answers are checked. The manually examined answers are then fed back to the datasets to form three expanded datasets: NTCIR5-CC-D200e, NTCIR5-CC-T200e, and NTCIR6-CC-T150e. In addition, we created the IASL-CC-Q465 dataset to increase the confidence in our experiments. It was created by three people using a program that randomly selected passages from the CIRB40 corpus. The human creator can use whatever keywords to search for more documents about a randomly selected passage, and create one or more questions based on the collected information. As a result, we created 465 questions in the IASL-CC-Q465 dataset, and combined with the NTCIR provided datasets, we had a total of 1,015 questions.

5.2 Evaluation Metrics

In this sub-section, we describe related evaluation metrics.

5.2.1 *R-Accuracy and RU-Accuracy.* R-Accuracy and RU-Accuracy are used to measure QA performance in NTCIR CLQA. A QA system returns a list of ranked answer responses for each question, but R-accuracy and RU-accuracy only consider the correctness of the top-1 rank answer response on the list. An answer response is a pair comprised of an answer and its source document. Each answer response is judged as Right, Unsupported, or Wrong, as defined in the NTCIR-6 CLQA overview [Lee et al. 2007]:

> "Right (R): the answer is correct and the source document supports it."
>
> Unsupported (U): the answer is correct, but the source document cannot support it as a correct answer. That is, there is insufficient information in the document for users to confirm by themselves that the answer is the correct one.
>
> Wrong (W): the answer "is incorrect."

Based on these criteria, the accuracy is calculated as the number of correctly answered questions divided by the total number of questions. R-accuracy means that only "Right" judgments are regarded as correct, while RU-accuracy means that both "Right" and "Unsupported" judgments are counted.

$$R\text{-}Accuracy = \frac{\#\text{ of questions for which the top1 rank answer is Right}}{\#\text{ of questions}}$$

$$RU\text{-}Accuracy = \frac{\#\text{ of questions for which the top1 rank answer is Right or Unsupported}}{\#\text{ of questions}}$$

Because R-accuracy only occurs a few times in this article, we use "accuracy" to refer to RU-accuracy when the context is not ambiguous.

Mean Reciprocal Rank (MRR)

We use MRR to measure the QA performance based on all the highest ranked correct answers, not just the top1 answer. The MRR is calculated as follows:

$$MRR = \frac{1}{\text{\# of questions}} \sum_{question_i} \begin{cases} \frac{1}{\text{the highest rank of correct answers}} & \text{, if a correct answer exists} \\ 0 & \text{, if no correct answer} \end{cases}$$

Expected Answer Accuracy (EAA)

In addition to using the normal answer accuracy metrics, we propose a new metric called the Expected Answer Accuracy (EAA). There are some cases where one method is not better than the other one, but with higher Accuracy or MRR value. This phenomenon usually occurs when several top answers have the same ranking score. We use the EAA to resolve such problems.

The EAA score of a ranking method is defined as follows:

$$EAA = \frac{1}{\text{\# of questions}} \sum_{question_i} \frac{\text{\# of correct answers with top1 rank score}}{\text{\# of answers with top1 rank score}}$$

6. EXPERIMENTS

We conducted three experiments. In the first two, we compared SCO-QAT, SCO-QAT with distance, and some shallow answer-ranking features. The results show that SCO-QAT is more accurate than the other shallow features, and the distance information further improves SCO-QAT's performance. In the third experiment, we applied an ABSP-based filter to evaluate the effectiveness of ABSPs.

6.1 Comparing SCO-QAT with Other Single Ranking Features

Answer correctness features are usually combined to achieve the best performance. However, the features in QA are usually combined by using heuristic methods. Although some systems have used machine learning approaches in QA ranking successfully, it is rare to find the same approach being applied to other QA work. This may be because QA feature combination methods are not mature enough to deal with the variability of QA systems, or the amount of training data may not be sufficient to train good models. Therefore, instead of combined features, we only study the effect of single ranking features. We assume they are more reliable and can be applied to other systems or languages more easily.

As well as SCO-QAT, we tested the following widely used shallow features: *keyword overlap*, *density*, *IR score*, *mutual information score*, and *answer frequency*. The keyword overlap is the ratio of question keywords found in a passage, as used in Cooper and Ruger [2000], Molla and Gardiner [2005], and Zhao et al. [2005]. The IR score [Kwok and Deng 2006; Zheng 2005], which is derived by the passage retrieval module, is the score of the passage containing

Table IV. The Performance of Single Features: "Accuracy" is the RU-Accuracy, "MRR" is the Top5 RU-Mean-Reciprocal-Rank, and "EAA" is the Expected Answer Accuracy

	Data: NTCIR5-CC-D200e			Data: NTCIR5-CC-T200e		
Feature	Accuracy	MRR	EAA	Accuracy	MRR	EAA
SCOQAT	0.545	0.621	0.522	0.515	0.586	0.515
KO	0.515	0.601	0.254	0.495	0.569	0.245
Density	0.375	0.501	0.368	0.390	0.479	0.380
Frequency	0.445	0.560	0.431	0.395	0.499	0.366
IR	0.515	0.598	0.425	0.495	0.569	0.420
MI	0.210	0.342	0.210	0.155	0.138	0.290
	Data: IASL-CC-Q465			Data: NTCIR6-CC-T150		
Feature	Accuracy	MRR	EAA	Accuracy	MRR	EAA
SCOQAT	0.578	0.628	0.546	0.413	0.495	0.406
KO	0.568	0.618	0.247	0.367	0.476	0.130
Density	0.432	0.519	0.369	0.340	0.420	0.314
Frequency	0.413	0.486	0.406	0.340	0.431	0.343
IR	0.518	0.587	0.406	0.367	0.460	0.283
MI	0.138	0.280	0.124	0.167	0.281	0.142

the answer. In ASQA, the IR score is calculated by the Lucene information retrieval engine. Density is defined as the average distance between the answer and the question keywords in a passage. There are several ways to calculate density. In our experiment, we simply adopt Lin's formula [Lin et al. 2005], which performed well in NTCIR-5 CLQA. The mutual information score is calculated by the PMI method [Magnini et al. 2001].

The experiment results are listed in Table IV. For C-C datasets, SCO-QAT outperforms the other shallow features on all three metrics. It achieves 0.522 EAA for the NTCIR5-CC-D200e dataset, 0.515 for the NTCIR5-CC-T200e dataset, 0.546 for the IASL-CC-Q465 dataset, and 0.406 for the NTCIR6-CC-T150 dataset. Compared to the other features, the differences are in the range 0.063∼0.522 in terms of EAA. We performed a paired *t*-test on the results. It also showed that SCO-QAT is significantly more accurate than all the other shallow ranking features.

In addition to comparing to single ranking features, we compare the SCO-QAT results with those of other participants in the NTCIR5 CLQA task (Table V). Because QA systems use combined features, this is a single-versus-combined-feature comparison. In the NTCIR5 CLQA [Sasaki et al. 2005], there were thirteen Chinese QA runs, and the accuracy ranged from 0.105 to 0.445, with a mean of 0.315. It is impressive that SCO-QAT achieved 0.515 accuracy[13], which is much better than ASQA at NTCIR-5 [Lee et al. 2005] (the best performing system in the NTCIR5 CLQA C-C subtask).

Although frequency is the simplest of the shallow features, it performs surprisingly well. This may be due to the effectiveness of ASQA answer filtering module, or the characteristics of the Chinese news corpus, or the way questions were created, which caused questions with high frequency answers to

[13]The 0.515 accuracy is based on the NTCIR5-CC-T200e dataset. If the NTCIR5-CC-T200 dataset is used, the accuracy is 0.505.

Table V. Performance Comparison of SCO-QAT (Single Feature) and the Best Systems at NTCIR5 and NTCIR6 CLQA (Combined Features)

Subtask	System	RU-Accuracy
NTCIR5 CC	Best Participant (ASQA)	0.445
	ASQA with SCO-QAT only	0.515
NTCIR6 CC	Best Participant (ASQA full version)	0.553
	ASQA with SCO-QAT only	0.413

Table VI. Summary of the Single Shallow Feature Experiment: CC-ALL is the Combination of All the CC Dataset Results in Table IV

	Data: CC-ALL		
Feature	Accuracy	MRR	EAA
SCOQAT	0.535	0.599	0.514
KO	0.513	0.584	0.231
Density	0.399	0.493	0.363
Frequency	0.405	0.495	0.394
IR	0.491	0.538	0.424
MI	0.160	0.264	0.176

be selected. We cannot find any articles reporting the effect of applying the frequency feature only. Further investigation is therefore needed to explain the phenomenon.

As shown in Table VI, the MI approach does not perform well in our experiment, possibly because the word-ignoring rules or the corpus were unsuitable. The performance of the density approach, which is popular in QA systems, was acceptable. However, we found that it was not suitable for processing "Organization" type questions, as its accuracy was only 0.10 to 0.15.

Although SCO-QAT was the best shallow feature in the experiment, a number of problems still need to be addressed. The most important issue is that if there is more than one highly related answer to the given question, SCO-QAT cannot determine which one is better. Take the Chinese question "請問微軟的總裁為誰?" (Who is the president of Microsoft?) in the dataset, for example. In this case, SCO-QAT gives "范成炬" (Allen Fan) a higher score than "Bill Gates." Since "Allen Fan" is the president of Microsoft (Taiwan), not the whole company, SCO-QAT cannot determine which answer is correct. Another relatively minor problem is that of improper question terms, such as functional words. This could be solved by removing the improper terms, but it would require some heuristic rules or external knowledge.

6.2 Enhancing SCO-QAT with Distance Information

We experimented on the extended version of SCO-QAT, described in Section 4.3 with the question-term-number threshold in Equation (7) set to 5. The results are listed in Table VII. SCO-QAT with distance information achieved 0.568 EAA for the NTCIR5-CC-D200e dataset, 0.538 for the NTCIR5-CC-T200e dataset, 0.565 for the IASL-CC-Q465 dataset, and 0.453 for the NTCIR6-CC-T150 dataset. Compared to the original SCO-QAT, improvements in the EAA score were in the range 0.019∼0.046. According to paired t-test, SCO-QAT with distance was significantly more accurate than SCO-QAT at the 0.01 level.

Table VII. The Performance of SCO-QAT and SCO-QAT_with Distance Information: "Accuracy" is the RU-Accuracy, "MRR" is the Top5 RU-Mean-Reciprocal-Rank, and "EAA" is the Expected Answer Accuracy

	Data: NTCIR5-CC-D200e			Data: NTCIR5-CC-T200e		
Feature	Accuracy	EAA	MRR	Accuracy	EAA	MRR
SCOQAT	0.545	0.522	0.621	0.515	0.515	0.586
SCOQAT_Dist	0.570	0.568	0.643	0.535	0.538	0.597
	Data: IASL-CC-Q465			Data: NTCIR6-CC-T150		
Feature	Accuracy	EAA	MRR	Accuracy	EAA	MRR
SCOQAT	0.578	0.546	0.628	0.413	0.406	0.495
SCOQAT_Dist	0.589	0.565	0.637	0.453	0.449	0.565

Table VIII. RU-Accuracy on the NTCIR-6-CC-T150 Dataset When SCO-QAT_Dist and ABSPs are Applied

Method	Accuracy
ASQA + ABSPs	0.911 (on covered questions)
ASQA + SCO-QAT_Dist	0.453
ASQA + SCO-QAT_Dist + ABSPs	0.5

6.3 ABSP-Based Answer Filter

ABSPs have been incorporated into ASQA as an answer filter. To evaluate the filter's performance, we used 865 training questions from NTCIR5-CC-D200e, NTCIR5-CC-T200e, and IASL-CC-Q465 datasets. For each training question, we applied the generation algorithm to the top 200 most relevant passages retrieved by the passage retrieval module and generated about 500 patterns in average. Finally, we collected 126 useful ABSPs from the 865 training questions. When the ABSP-based answer filter was used in ASQA for the NTCIR-6 dataset, the RU-accuracy increased from 0.453 to 0.5, as shown in Table VIII. To determine whether the improvement was statistically significant, we applied the McNemar test. The result shows that, at the 0.01 level, the system with the ABSP-based filter is significantly more accurate than the system without the filter. Because the answer filter can only be applied when useful relations are extracted according to the matches, we also analyzed ABSP performance on the questions covered by the filter. For the NTCIR-6 dataset, the question coverage was 37.3% and the accuracy of the questions covered was 0.911. The accuracy rate was much higher than the overall accuracy rate, which was 0.5. The high accuracy score demonstrates the accuracy of surface pattern-based approaches. With regard to question coverage, although the score was not high, we covered the questions with only 126 selected ABSPs. We believe we can increase the question coverage score if we have larger training dataset.

7. DISCUSSION

Co-occurrence methods and surface pattern methods rely on global information and local information respectively. In other words, co-occurrence-based methods like SCO-QAT are suitable for questions where the answers are provided in several passages. For example, consider the question "請問涉嫌竊取美國洛薩拉摩斯實驗室核武機密的華裔科學家為誰？" (Which Chinese scientist

was accused of violating the Atomic Energy Act because of his purported mishandling of restricted data at Los Alamos National Laboratories?) In this case, there are 48 passages describing the answer and the event, so the question can be answered by SCO-QAT. Surface pattern-based methods like ABSPs are suitable for questions in which the syntax patterns for validating the answers are commonly used and easy to extract, for example, surface patterns for answering birth date questions.

Because SCO-QAT relies on passage retrieval, it is highly dependent on the quality of the passages retrieved. SCO-QAT fails when other related terms are the same type as the answer type. For example, the question ""西安事變中誰遭綁架？" (Who was abducted in the Xi'an Incident[14]?) requires a person as its answer, but ""蔣中正""(Chiang Kai-shek) the abducted person and ""張學良""(Zhang Xueliang) the abductor are usually mentioned in sentences describing ""西安事變"" (the Xi'an Incident). The improved version of SCO-QAT described in Section 4.3 incorporates distance information, so it can deal with some questions of this kind; however, there are still some cases that cannot be solved. For example, in the following sentence, "張學良於西安事變挾蔣中正以令諸侯" (Zhang Xueliang abducted Chiang Kai-shek in the Xi'an Incident), "張學良" and "蔣中正" co-occur in the same sentence and have the same distance to the question keyword "西安事變". Therefore, it is not possible to use SCO-QAT with distance information to deal with this case.

From the viewpoint of surface patterns, the above cases are easy to handle, as long as we have the surface patterns for "綁架" (abduction). However, generating surface patterns for every kind of question is almost impossible, because it is time-consuming and labor intensive. Surface pattern-based methods cannot be applied to, or cover, some questions if there is no surface pattern.

Next we describe the strategies adopted to solve the coverage problem.

7.1 Generate Patterns for Relations Instead of for Questions

Instead of generating surface patterns for question types, we generate surface patterns that can capture important relations for answering factoid questions. Also, we do not define the relation types, but let the ABSP training process choose patterns according to the training data. For example, the pattern " *(ORGANIZATION) Na Nb (PERSON)* 表示， " is useful because it matches passages that describe the relations between the question keywords and the answers. For the same relations, the traditional surface pattern approach may predefine some question types, such as Q_PERSON_WORKFOR or Q_ORGANIZATION_POSITION. It then applies question analysis methods to these extra question types, generates patterns for the question types, and manually defines the mapping between extracted question keywords and the slots in the patterns. The process obviously requires more effort than our approach because we only generate patterns for capturing important relations. Moreover, our approach can be used for any question types that need relation information to verify the correctness of an answer.

[14]Xi'an Incident in Wikipedia: http://en.wikipedia.org/wiki/Xi%27an_Incident

7.2 Simulate Long Patterns with Multiple Short Patterns

A factoid question can be viewed as a set of relation constraints about the answer or answers. From this perspective, instead of relying on one pattern, we apply several ABSPs to a single passage and merge all the identified relations based on their common terms. This technique partly solves the low coverage problem because our surface-pattern approach simulates the effect of large surface patterns with several small surface patterns. In other words, our ABSP method needs fewer surface patterns. The drawback of this approach is a loss of accuracy due to accidental merging of inappropriate relations; even so, our experiment results show that the accuracy is still acceptable.

7.3 Merge Relations from Multiple Passages

Relation constraints for an answer may not co-locate in a single passage; therefore, we merge relations extracted by ABSPs from multiple passages. This mechanism leads to the same accuracy issue as that in strategy 7.2 above.

7.4 More Semantic Tags

We tag sentences with more semantic tags to make surface patterns more abstract so that they can handle a question type with fewer patterns. The effect is much like that achieved by using POS tags, but the accuracy is usually higher.

Both the SCO-QAT and ABSP methods are affected by the word canonicalization problem. For example, "Taiwan" could be "台灣" in simplified form or "臺灣" in traditional form; "thirteen" could be "13" in Arabic numerals, "１３" in capitalized Arabic numerals, "十三" in Chinese numerals, or "拾參" in capitalized Chinese numerals; foreign person names like Jordan could be "喬丹" or "喬登"; "China" could be "大陸", "中國", and "內地". Because we do not use canonicalization in the ASQA system, SCO-QAT underestimates the score of some correct answers and ABSPs fail to merge some important relations.

Although we have demonstrated the effectiveness of SCO-QAT and ABSPs on Chinese Factoid QA, more experiments are needed before they can be applied to other languages. We believe SCO-QAT can be used directly in other languages, since SCO-QAT is a pure co-occurrence-based method that is not dependent on syntax features. However, for ABSPs to perform well in languages with complex syntactic structures, more syntactic constructs, such as noun phrases or verb phrases, may be needed. For example, in our experience, English sentences are more structured than Chinese sentences and the distances between dependent words are usually longer. This may result in a less than optimal performance, because the patterns generated by ABSPs may not be long enough to capture the relations described in English sentences.

8. CONCLUSION AND FUTURE WORK

We propose two lightweight methods, SCO-QAT and ABSPs, for use in a state-of-the-art Chinese factoid QA system. The methods require fewer resources than heavy methods, such as the parsers and logic provers used in state-of-the-art QA systems in other languages.

We show that lightweight methods can operate in resource-limited situations, and that they have great potential to boost QA performance. We improved the RU-Accuracy of the ASQA system from 0.445 to 0.535 when it was tested on NTCIR-5 CLQA C-C datasets. The result is significant and impressive because only the answer filtering and answer ranking modules are changed. The quality of the answers extracted by the answer extraction module is unchanged, but the answer selection strategy is improved. The enhanced system also performed well in NTCIR-6, achieving 0.553 RU-Accuracy (0.5 of the RU-Accuracy was contributed by SCO-QAT and ABSPs) in the C-C subtask.

The ABSP method is a variation of surface pattern methods. It tries to increase question coverage and maintain accuracy by targeting surface patterns for all question types, instead of specific question types, combining relations extracted by multiple surface patterns from multiple passages, and incorporating richer semantic tags. By using this strategy, ABSPs can achieve 37.33% coverage and 0.911 RU-Accuracy on the questions covered.

The SCO-QAT method utilizes co-occurrence information in retrieved passages. Since it calculates all the co-occurrence combinations without extra access to the corpus or the Web, it is suitable for bandwidth-limited situations. Moreover, SCO-QAT does not require word-ignoring rules to handle missing counts and it can be combined with other answer ranking features. ASQA achieved 0.535 on the NTCIR-5 C-C dataset by only ranking with SCO-QAT enhanced with distance information, which was better than all the combined features used at NTCIR-5. We attribute the success of the ABSP and SCO-QAT methods to the effective use of local syntactic information and global co-occurrence information.

SCO-QAT and ABSPs can be improved in several ways. In both methods, applying rules with taxonomy or ontology resources would solve most canonicalization problems. For SCO-QAT, it would be helpful if we were to use a better term weighting scheme. Using more syntactic information, such as incorporating surface patterns, would result in more reliable co-occurrence calculations. For ABSPs, more accurate semantic tags, which are usually finer-grained, would improve the accuracy while maintaining question coverage. Also, to increase question coverage, in addition to the strategies we adopt for ABSPs, we could also use partial matching because it allows portions of a surface pattern to be unmatched. Allowing overlapping tags is also a possibility, because some errors are caused by tagging, such as wrong word segmentation.

APPENDIX

A. ABSPs Used in ASQA at NTCIR-6 CLQA

(1) ARTIFACT Na PERSON VE ،
(2) ARTIFACT Na - PERSON
(3) ARTIFACT Na Na Na PERSON
(4) ARTIFACT TIME P LOCATION
(5) LOCATION Na PERSON - VC
(6) LOCATION Na P PERSON LOCATION Na - Na

(7) LOCATION Na P TIME VH
(8) LOCATION Na PA ARTIFACT PA
(9) LOCATION Na - NUMBER OCCUPATION - PERSON
(10) LOCATION Na OCCUPATION PERSON
(11) LOCATION - TIME V LOCATION
(12) LOCATION - Na Na PERSON
(13) LOCATION LOCATION Na PERSON Na
(14) LOCATION LOCATION 的 ORGANIZATION
(15) LOCATION NUMBER Na - OCCUPATION PERSON
(16) LOCATION OCCUPATION PERSON
(17) LOCATION OCCUPATION PERSON VJ TIME - PERSON Na
(18) LOCATION ORGANIZATION T ORGANIZATION Na
(19) LOCATION ORGANIZATION - Na PERSON
(20) LOCATION 的 LOCATION
(21) LOCATION 的 Nc LOCATION
(22) LOCATION Nc LOCATION
(23) LOCATION 的 OCCUPATION PERSON
(24) LOCATION 的 ORGANIZATION
(25) LOCATION 位於 LOCATION LOCATION
(26) OCCUPATION Na - V - PERSON
(27) OCCUPATION PERSON 與 PERSON
(28) ORGANIZATION Caa ORGANIZATION TIME VE V
(29) ORGANIZATION N - PERSON
(30) ORGANIZATION N - PERSON PA N
(31) ORGANIZATION N - PERSON V N N
(32) ORGANIZATION N - NE VE，
(33) ORGANIZATION N Na PERSON
(34) ORGANIZATION N Na PERSON V
(35) ORGANIZATION N Na NE P
(36) ORGANIZATION N Na NE PA N N N
(37) ORGANIZATION OCCUPATION PERSON
(38) ORGANIZATION Na PERSON
(39) ORGANIZATION Na PERSON - VE CO
(40) ORGANIZATION Na PERSON - 表示，
(41) ORGANIZATION Na PERSON V
(42) ORGANIZATION Na PERSON VE， ORGANIZATION
(43) ORGANIZATION Na - PERSON
(44) ORGANIZATION Na - PERSON V
(45) ORGANIZATION Na - NE VE CO
(46) ORGANIZATION Na - NE VE，

(47) ORGANIZATION Na - NE 表示，
(48) ORGANIZATION Na NE V - Nb
(49) ORGANIZATION Na NE VE CO
(50) ORGANIZATION Na NE VE CO N
(51) ORGANIZATION Na NE VE，
(52) ORGANIZATION Na NE VE， Nb
(53) ORGANIZATION Na NE 表示， Nb
(54) ORGANIZATION Na NE 說 CO
(55) ORGANIZATION Na Na PERSON VC
(56) ORGANIZATION Na Na - NE
(57) ORGANIZATION Na Nb PERSON 表示，
(58) ORGANIZATION Nc Na PERSON
(59) ORGANIZATION Nc Na PERSON V
(60) ORGANIZATION Nd Na ORGANIZATION
(61) ORGANIZATION 次長 NE
(62) ORGANIZATION VCL LOCATION
(63) ORGANIZAITON C ORGANIZATION TIME VE V
(64) ORGANIZATION - PERSON VC LOCATION
(65) ORGANIZATION - PERSON VC ORGANIZATION
(66) ORGANIZAITON - OCCUPATION PERSON
(67) PERSON V， ORGANIZATION
(68) PERSON 表示， ORGANIZATION
(69) PERSON Nd VJ Na X Na VJ ORGANIZATION
(70) PERSON P ORGANIZATION Nc Nc Na
(71) PERSON P ORGANIZATION V，
(72) PERSON P V LOCATION OCCUPATION
(73) PERSON Na VC LOCATION
(74) PERSON Na VC ORGANIZATION
(75) PERSON Nc - OCCUPATION PERSON
(76) PERSON VC - Na ARTIFACT
(77) PERSON VC ARTIFACT VC ORGANIZATION
(78) PERSON VC LOCATION Na
(79) PERSON VC OCCUPATION Na
(80) PERSON VC PERSON
(81) PERSON VC PA ARTIFACT PA
(82) PERSON VC 的 - PA ARTIFACT PA
(83) PERSON VG LOCATION OCCUPATION
(84) PERSON VG - VG - - OCCUPATION
(85) PERSON VJ ORGANIZATION - ARTIFACT
(86) PERSON 是 ORGANIZATION - OCCUPATION

(87) PERSON 的 ORGANIZATION
(88) PERSON 的 PA ARTIFACT PA
(89) TIME - LOCATION LOCATION VJ
(90) TIME P LOCATION VE DE ARTIFACT
(91) TIME ARTIFACT V
(92) TIME LOCATION Na - Na
(93) TIME 的 LOCATION Na
(94) TIME VC LOCATION Na
(95) C ORGANIZATION N Na PERSON
(96) C ORGANIZATION Na - PERSON
(97) CO ORGANIZATION N N
(98) CO PERSON - P VG LOCATION Neu Nf - Na
(99) N Na X PERSON VE， ORGANIZATION
(100) N， - ORGANIZATION N N Nb VC
(101) NE VE， ORGANIZATION N N
(102) Na - V ORGANIZATION 董事長 NE
(103) Na NE V， ORGANIZATION
(104) Na Na Nb VE， ORGANIZATION
(105) Na Nb V X ORGANIZATION N N
(106) Na PERSON P PERSON Nd V
(107) Na PERSON P PERSON V Na
(108) Na OCCUPATION FW PERSON
(109) Nc Na PERSON 說， ORGANIZATION N V
(110) Nc 、 NE X NE 、 LOCATION
(111) P ORGANIZATION Na PERSON
(112) P ORGANIZATION Na PERSON X V
(113) P PERSON T ORGNIZATION 及 LOCATION
(114) P LOCATION ORGANIZATION VH
(115) PA ARTIFACT PA OCCUPATION PERSON
(116) VE， LOCATION N N P LOCATION
(117) V Di PERSON ARTIFACT
(118) VC PA PERSON ARTIFACT PA
(119) VH 的 Na OCCUPATION PERSON
(120) VH 的 ORGANIZATION OCCUPATION PERSON
(121) VH 的 LOCATION OCCUPATION PERSON
(122)， ORGANIZATION Na PERSON
(123)， ORGANIZATION Na Na PERSON 的
(124)， ORGANIZAITON ORGANIZATION Na
(125) 在 LOCATION Nc LOCATION
(126) 最 VH 的 OCCUPATION PERSON

B. Stop Word List for SCO-QAT

請問	哪一座	哪一家	哪一國
哪一些	哪一場	哪一種	哪一部
那一部	那一間	哪一個	那一個
哪一支	那一支	哪一項	那一項
哪一位	那一位	哪一艘	那一艘
哪一間	那一間	哪國籍	哪國
哪個人	哪個	那個人	那個
哪家	那家	哪種	那種
哪位	那位	在哪裡	哪裡
在那裡	那裡	在哪裏	哪裏
在那裏	那裏	哪件	在哪處
哪處	在那處	那處	那件
是什麼	為什麼	什麼	有多少個
有多少	是多少	為多少	多少
有多遠	多遠	是誰	為誰
為何人	為何國	為何	為名
何部	何處	何種動物	何種
何人	何地	何國	何時
何年	的名字	電影	其他
其它	一些	單位	因素
知名	廠商	誰	這部
這	每	是	的
～	：	(	)
《	》	〈	〉
？	?	，	,
「	」		

REFERENCES

BARZILAY, R. AND LEE, L. 2003. Learning to paraphrase: An unsupervised approach using multiple-sequence alignment. In *Proceedings of the Joint Human Language Technology Conference/Annual Meeting of the North American Chapter of the Association for Computational Linguistics (HLT-NAACL'03)*, 16–23.

BOUMA, G., MUR, J., AND NOORD, G. V. 2005. Reasoning over dependency relations for QA. In *Proceedings of the IJCAI Workshop on Knowledge and Reasoning for Answering Questions (KRAQ'05)*, 15–21.

CLARKE, C. L. A., CORMACK, G., KEMKES, G., LASZLO, M., LYNAM, T., TERRA, E., AND TILKER, P. 2002. Statistical selection of exact answers (multitext experiments for TREC'02). In *Proceedings of the 11th Text Retrieval Conference (TREC'02)*, 823–831.

CLARKE, C. L. A., CORMACK, G. V., AND LYNAM, T. R. 2001. Exploiting redundancy in question answering. In *Proceedings of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'01)*, 358–365.

COOPER, R. J. AND RUGER, S. M. 2000. A Simple Question Answering System. In *Proceedings of the 9th Text Retrieval Conference (TREC'00)*.

CUI, H., SUN, R., LI, K., KAN, M.-Y., AND CHUA, T.-S. 2005. Question answering passage retrieval using dependency relations. In *Proceedings of the 28th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'05)*, Salvador, Brazil, 400–407.

DAY, M.-Y., LEE, C.-W., WU, S.-H., ONG, C.-S., AND HSU, W.-L. 2005. An integrated knowledge-based and machine learning approach for Chinese question classification. In *Proceedings of the IEEE International Joint Conference on Natural Language Processing and Knowledge Engineering (NLP-KE'05)*.

HARABAGIU, S., MOLDOVAN, D., CLARK, C., BOWDEN, M., HICKL, A., AND WANG, P. 2005. Employing two question answering systems in TREC'05. In *Proceedings of the 14th Text Retrieval Conference (TREC'05)*.

HSU, W.-L., WU, S.-H., AND CHEN, Y.-S. 2001. Event identification based on the information map-INFOMAP. In *Proceedings of the IEEE International Conference on Systems, Man, and Cybernetics (SMC'01)*, Tucson, AZ, 1661–1666.

HUANG, M., ZHU, X., HAO, Y., PAYAN, D. G., QU, K., AND LI, M. 2004. Discovering patterns to extract protein - protein interactions from full texts. *Bioinformatics 20*, 3604–3612.

KWOK, K.-L. AND DENG, P. 2006. Chinese question-answering: Comparing monolingual with English-Chinese cross-lingual results. In *Proceedings of the 3rd Asia Information Retrieval Symposium (AIRS'06)*, 244–257.

LAURENT, D., SÉGUÉLA, P., AND NÈGRE, S. 2006. Cross lingual question answering using QRISTAL for CLEF 2006. In *Proceedings of the 7th Workshop of the Cross-Language Evaluation Forum (CLEF'06)*.

LEE, C.-W., DAY, M.-Y., SUNG, C.-L., LEE, Y.-H., JIANG, T.-J., WU, C.-W., SHIH, C.-W., CHEN, Y.-R., AND HSU, W.-L. 2007. Chinese-Chinese and English-Chinese question answering with ASQA at NTCIR-6 CLQA. In *Proceedings of NII-NACSIS Test Collection for Information Retrieval Systems (NTCIR'07)*, Tokyo, Japan, 175–181.

LEE, C.-W., SHIH, C.-W., DAY, M.-Y., TSAI, T.-H., JIANG, T.-J., WU, C.-W., SUNG, C.-L., CHEN, Y.-R., WU, S.-H., AND HSU, W.-L. 2005. ASQA: Academia sinica question answering system for NTCIR-5 CLQA. In *Proceedings of NII-NACSIS Test Collection for Information Retrieval Systems (NTCIR'05)*, Tokyo, Japan.

LIN, F., SHIMA, H., WANG, M., AND MITAMURA, T. 2005. CMU JAVELIN System for NTCIR5 CLQA1. In *Proceedings of NII-NACSIS Test Collection for Information Retrieval Systems (NTCIR'05)*.

LIN, J. 2005. Evaluation of resources for question answering evaluation. *Proceedings of the 28th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'05)*, 392–399.

LIN, S.-J., SHIA, M.-S., LIN, K.-H., LIN, J.-H., YU, S., AND LU, W.-H. 2005. Improving answer ranking using cohesion between answer and keywords. In *Proceedings of NII-NACSIS Test Collection for Information Retrieval Systems (NTCIR'05)*.

MAGNINI, B., PREVETE, M. N. R., AND TANEV, H. 2001. Is it the right answer? Exploiting Web redundancy for Answer Validation. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics (ACL'02)*, 425–432.

MOLLA, D. AND GARDINER, M. 2005. AnswerFinder—Question answering by combining lexical, syntactic and semantic information. In *Australasian Language Technology Workshop (ALTW'05)*.

MUSLEA, I. 1999. Extraction patterns for information extraction tasks: A survey. In *Proceedings of the Workshop on Machine Learning for Information Extraction (MLIE'99)*, Orlando, Florida.

RAVICHANDRAN, D. AND HOVY, E. 2001. Learning surface text patterns for a question answering system. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics (ACL'02)*, Philadelphia, Pennsylvania, 41–47.

RAVICHANDRAN, D. AND HOVY, E. 2002. Learning surface text patterns for a question answering system. *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics (ACL'02)*, 41–47.

SAIZ-NOEDA, M., SU'AREZ, A., AND PALOMAR, M. 2001. Semantic pattern learning through maximum entropy-based WSD technique. In *Proceedings of the 5th Conference on Natural Language Learning (CoNLL'01)*, Toulouse, France.

SASAKI, Y., CHEN, H.-H., CHEN, K.-H., AND LIN, C.-J. 2005. Overview of the NTCIR-5 cross-lingual question answering task. In *Proceedings of the NII-NACSIS Test Collectionfor Information Retrieval Systems (NTCIR'05)*, Tokyo, Japan, 175–185.

SMITH, T. F. AND WATERNMAN, M. S. 1981. Identification of common molecular subsequences. *J. Mol. Biol. 147*, 195–197.

SOUBBOTIN, M. M. AND SOUBBOTIN, S. M. 2001. Patterns of potential answer expressions as clues to the right answers. *Proceedings of the 10th Text Retrieval Conference (TREC'01)*.

STAAB, S., ERDMANN, M., AND MAEDCHE, A. 2001. Engineering Ontologies using Semantic Patterns. In *Proceedings of the IJCAI-2001 Workshop on E-Business and Intelligent Web*. Seattle, Washington.

TAKAHASHI, T., NAWATA, K., INUI, K., AND MATSUMOTO, Y. 2004. NAIST QA System for QAC2. In *Proceedings of the NII-NACSIS Test Collection for Information Retrieval Systems (NTCIR'04)*, Tokyo, Japan.

VAPNIK, V. N. 1995. *The nature of statistical learning theory*. Springer.

WU, C.-W., JAN, S.-Y., TSAI, R. T.-H., AND HSU, W.-L. 2006. On using ensemble methods for Chinese named entity recognition. In *Proceedings of the 5th SIGHAN Workshop on Chinese Language Processing*.

ZHAO, Y., XU, Z. M., GUAN, Y., AND LI, P. 2005. Insun05QA on QA track of TREC'05. In *Proceedings of the 14th Text Retrieval Conference (TREC'05)*.

ZHENG, Z. 2002. AnswerBus question answering system. *Human Language Technology Conference (HLT'02)*, 24–27.

Received December 2007; revised February 2008, May 2008; accepted August 2008