行政院國家科學委員會專題研究計畫　成果報告

問答系統技術研發(3/3)－異質資訊源問答系統之研究

計畫類別：個別型計畫
計畫編號：NSC93-2213-E-002-009-
執行期間：93 年 08 月 01 日至 94 年 07 月 31 日
執行單位：國立臺灣大學資訊工程學系暨研究所

計畫主持人：陳信希

計畫參與人員：林川傑、曾郁淳

報告類型：完整報告

處理方式：本計畫可公開查詢

中　華　民　國 94 年 10 月 17 日

行政院國家科學委員會補助專題研究計畫成果報告

問答系統技術研發

計畫類別：個別型計畫
計畫編號：NSC 93－2213－E－002－009－
執行期間：2004 年 8 月 1 日至 2005 年 7 月 31 日

計畫主持人：陳信希
共同主持人：
計畫參與人員： 林川傑、曾郁淳

成果報告類型(依經費核定清單規定繳交)：完整報告

本成果報告包括以下應繳交之附件：
□赴國外出差或研習心得報告一份
□赴大陸地區出差或研習心得報告一份
□出席國際學術會議心得報告及發表之論文各一份
□國際合作研究計畫國外研究報告書一份

處理方式：除產學合作研究計畫、提升產業技術及人才培育研究計畫、列管計畫及下列情形者外，得立即公開查詢

執行單位：國立台灣大學資訊工程學系

中　華　民　國　九十四年十月十七日

Chapter 1 Lessons in TREC QA Tasks

1. Introduction

Question Answering (QA) becomes a hot research topic in recent years due to the very large virtual database on the Internet. QA is defined to find the exact answer, which can meet the users' need more precisely, from a huge unstructured database. Traditional information retrieval systems cannot afford to resolve this problem. On the one hand, users have to find out the answers by themselves from the documents returned by IR systems. On the other hand, the answers may appear in any documents, even that the document is irrelevant to the question. In this chapter, we will present some of our results in TREC QA evaluation series.

2. Description of Our System at TREC8

Two possible approaches, i.e., keyword matching and template extraction, can be considered. Keyword matching postulates that the answering text contains most of the keywords. In other words, it carries enough information relevant to the question. Using templates is some sort of information extraction. The contents of documents are represented as templates. To answer a question, we have to select an appropriate template, then fill the template and finally offer the answer. The major difficulties in this approach are to find general domain templates, and to decide which template can be applied to answer the question.

Some other techniques are also useful. For example, to answer the questions "Who ..." and "When ...", the identification of named entities like person names and time/date expressions will help to locate the answer.

In our preliminary study, we adopt keyword-matching strategy coupling with expanding the keyword set selected from the question sentence by the synonyms and the morphological forms. The detail will be presented below.

The system is composed of three major steps: (1) preprocessing the question sentences, (2) retrieving the documents containing answers, and (3) retrieving the sentences containing answers.

2.1 Preprocessing the Question Sentences

Our main strategy is keyword matching. This approach has a drawback, i.e., the words used in the question sentences and in the sentences containing the answers may be different. For example, verbs can be in different tenses and synonyms can also be used. Therefore, we have to make necessary changes and expansions in the question sentences.

At first, the parts-of-speech are assigned to the words in question sentences. Then, stop-words are removed. The remaining words are transformed into the canonical forms and selected as the keywords of the question sentences. For each keyword, we find all of its synonyms from WordNet 1.6. Those terms form an expansion set for the keyword. If the keyword is a noun, a verb, an adjective, or an adverb, all the possible morphological forms of the words in the expansion set are also added into this set. Here the morphological forms are the plural of a noun, different tenses of a verb, and the comparison of an adjective or an adverb. They are shown as follows:

noun AAA: AAAs | AA[s,z,sh]es

verb BBB: BBBs | BB[s,z,sh]es + BBBed BBBing | BB[e]d BB[e]ing

adjective or adverb CCC: CCCer CCCest | CC(y)ier CC(y)iest

The irregular nouns and verbs can be transformed by looking up the WordNet.

2.2 Retrieving the Documents Containing Answers

We employ a full text retrieval system to find the documents that may contain the answers. The purpose is to decrease the number of documents we have to search the answering sentences. Each keyword of a question sentence is assigned a weight. Those words tagged as NNP and NNPS, which denote proper nouns, have assigned higher weights. This is because they should be presented in the answer. The score of a document is computed as follows:

$$score(D) = \sum_{t \in T} weight(t)$$

$$D: \text{document},\ T = \{x | D \cap X \neq \phi\} \text{ where } x \text{ is a keyword and } X \text{ its expansion set}$$

The document containing one keyword or any words in its expansion set earns a score from this keyword. For example, consider the Question 30:

<num> Number: 30

What are the Valdez Principles?

Its keywords are “Valdez” and “Principles”, and the expansion sets are [valdez/ valdezes/] [principle/principles/rule/rules/precept/precepts/rationale/rationales/], respectively. If a document contains “principles” and “rules”, but no “valdez” and “valdezs”, its score is only the weight of “Principles”.

Those documents that have scores no less than the threshold are selected as the answering documents. Threshold is set to the sum of weights of the words in the original question sentence. Note that the removed words have no scores. If no documents have scores greater than the threshold, we assume that no answers can be found for the question.

2.3 Retrieving the Sentences Containing Answers

Finally, we examine each sentence in the answering documents. Those sentences that contain most words in the expanded question sentence are retrieved. The top five sentences are regarded as the answers. If there are more than five possible answers, we randomly select five of them. To meet the limit of 250 bytes, we truncate the sentences that exceed the limit. On the contrary, if the answer is shorter than the limit, we concatenate it with the next sentences.

2.4 Results and Discussions

The system runs on the 198 questions provided by Q&A Track of TREC-8. The weights of proper noun keywords are set to 100, and the others are set to 1. Among these 198 questions, 60 have answers. Total 25 of them are correct, and 20 answers are at the top scores. The following shows some examples.

<num> Number: 29

What is the brightest star visible from Earth?

Ans: In the year 296036, Voyager 2 will make its closest approach to Sirius, the brightest star visible from Earth. Deep space is benign, so dust and cosmic rays will erode Voyager 2 extraordinarily slowly. In a billion or more years, Sagan said, "there w

<num> Number: 102

Who is the Voyager project manager?

Ans: Until December, Voyager 2 occasionally will glance at Neptune and dark space to improve the accuracy of observations its cameras and instruments made during the Neptune flyby, said Voyager project manager

Norm Haynes. Pictures of empty space let engi

We examine the results of formal runs, and find that the system can be improved from several aspects:

(1) execution speed of the system

Owing to the long time required, 138 questions in the formal run do not have answers. After revising our algorithm and running again, we answer 136 questions. The evaluation is done by ourselves. Total 62 of them are correct, and 42 answers are at the top scores.

(2) anaphor resolution

The answering sentence may contain pronouns referring to the constituents in the previous sentences. We have to find the antecedents. Similarly, date expressions like today have to be substituted by an exact time.

(3) phrasal searching

Phrasal searching is helpful in some kind of questions. For example, to answer the questions

<num> Number: 115

What is Head Start?

<num> Number: 40

Who won the Nobel Peace Prize in 1991?

the key phrases "head start" and "Nobel peace prize" are very useful to find the answers.

3. Description of Our System at TREC9

3.1 QA Track

In TREC-8, we've experimented on expanding questions by adding inflections of verbs and nouns, as well as their synonyms (Lin and Chen, 1999). However, the performance was not as good as our expectation. This year we propose three models to see whether expansion is helpful or not. Model 1 is a base model. Only inflections are added. Model 2 adds synonyms from WordNet (Miller, 1990). And Model 3 tries to resolve co-reference in a simple way. Each of them will be described in detail in later sections.

Besides, we select answers according to the named entities that the question might be relevant. Our QA system will guess the interested entity type by looking at the questions. Position of the interested answer terms is also important. If the length of answering sentences is longer than restricted length, the final answer text has to include the actual answer. We also propose a method to implement this idea. The proposed algorithm will be described later.

3.2 Model Description

3.2.1 Interested Entity Type

After taking a question as input, our system first guesses which entity type the question is interested in. The method is simply rule-based. If the question starts with "who", "when", and "where", it may ask for a person name, a time/date expression, and a location name, respectively. If it starts with "what" or "which", or it is the "Name a ..." -type question, then the system goes on to look at the first noun behind it. We collected some keywords to indicate the interested entity types, such as "country" for location

name, “person” for personal name, and so on.

3.2.2 Named Entity Extraction

Named entity extraction plays an important role in our experiments. It is introduced while deciding question focusing, doing question expansion, and measuring similarity between document passage and question sentence.

For named entity extraction, we employ several named entities dictionaries, such as gazetteer, a collection of family name, *etc*. Different from simply dictionary look-up, these dictionaries also include other useful information. For a personal name, we can know that it is a family name, a male first name, or a female first name. For a country name, we can get its adjective form as well as how to call its people. For other location names, it provides the names of provinces or countries it belongs to as well. Organization names are accompanied by their abbreviations. We have not employed the information of types of personal names and the superior administrative division yet.

Time/date expression is simply keywords (Sunday, January, etc.) The resolution of expressions like “yesterday”, “last week”, and so on, is still undergoing. Other named entities like quantity and numbers are not handled yet.

3.2.3 Base Model - Question Expansion by Named Entity and Inflection Forms

In Base Model, we first decide if there is a named entity in the question sentence. If so, we record its equivalence (e.g. abbreviation of an organization name). Notice that a named entity can be more than one word. For the rest words in the question sentence, we remove stop words and attach the root form and all the inflection forms of each of them. These newly invited terms are for the use of similarity comparison later.

The next step is to segment documents into passages as comparison units. The document set we use this year is the set of the 50 most relevant documents to the questions. The relevant document set is offered by NIST. In the Base Model, a passage is simply a sentence.

For each passage, we also identify named entities in it, but their equivalences are not attached. The inflections are not added either. This is because we have already introduced them in the question side.

Then we measure its similarity to the expanded question sentence. For each word (or phrase) occurs in the passage and also in the expanded question, it contributes a score to the similarity. By the recent experiment, if it is a named entity, it contributes 2 points; otherwise 1 point. If it occurs in the original question, the contributed score is doubled.

Besides, if a word (or a phrase) does not occur in the question but is of the interested type of the question, the FOCUS tag is set and the position of this word is recorded.

While giving answers, those words (or phrases) that are assigned the FOCUS tag are reported first. The passage of higher score is considered to be more possible to carry the answer and is ranked higher.

To meet the length restriction, we have to truncate the passages longer than 250 bytes. We decide the focusing center of each answering passage first. Truncate characters 125 bytes ahead of the center and also the exceed part if the remaining passage is still longer than 250 bytes. For those assigned a FOCUS tag, the center is the average position of all the found named entities of interest. For those did not, the center is the average position of words that also occur in the question sentence.

3.2.4 Model 2 – More Expansion by Synonyms

Besides the basic structure of Base Model, we also expand questions by the synonyms of ordinary nouns or verbs, i.e., those which are not named entities. Synonyms are obtained by looking up the WordNet (Miller, 1990). We do so because we want to save those answers written in different terms.

3.2.5 Model 3 – Passage with Co-Reference Resolved

This model is also based on the Base Model. But we want to resolve co-reference problem first before measuring similarity with the question sentence. We proposed a simple strategy to do so: take the first sentence as a passage. If the next sentence contains pronouns except “it”, it is merged into the previous passage. Or if the next one contains a phrase of the pattern “the A” and the word “A” occurs in the previous passage, it is merged into the previous one, too. It can help resolve anaphora problem as well as the co-referential noun phrases.

3.3 Evaluation

Table 1 lists the results of our three models. We submitted three runs, each run for each model, i.e., qantu01 for Base Model, and so on. Each answer text can be judged as Wrong, Correct, and Unsupported. "Unsupported" means that the document associated to the answer text does not really support the answer. The Strict Evaluation only counts Correct ones, and the Lenient Evaluation takes both Correct and Unsupported ones as correctly answered.

Table 1. Results of Three Models in the QA Track at TREC-9

Run ID	Strict		Lenient		Strict (Debugged)	
	MMR	Failed	MMR	Failed	MMR	Failed
qantu01	0.315	377 (55.3%)	0.348	354 (51.9%)	0.333	368 (55.0%)
qantu02	0.315	376 (55.1%)	0.341	354 (51.9%)	0.327	365 (53.5%)
qantu03	0.278	394 (57.8%)	0.309	370 (54.3%)	0.284	394 (57.8%)

From Table 1, half of the questions failed to be answered. It is better than last year that we only answered 1/3 of the questions correctly. There are 24 more questions in average answered by unsupported documents.

Comparing the performance of different models, Base Model and Model 2 are almost the same, but Model 3 is worse than the other two. Model 2 answered one more question than Base Model did, but Base Model offered unsupported answers at higher ranks than Model 2 did in the Lenient Evaluation. Model 3 is worse in either evaluation.

It seems that adding synonyms does not help a lot. It even lows down the speed. The most difficulties we met in QA are often paraphrases, not only synonyms. Therefore, it might be more efficient to tackle the paraphrases problem.

The reason that Model 3 worked badly may be the over-simplified co-occurrence resolution. For those questions failed to be answered here but successful in the other two runs, it was often the case that the passages containing the answer texts have been expanded into large ones. The occurrence of co-reference candidates is too frequent to simply concatenate sentences.

But co-reference resolution is helpful for question answering. During the investigation, we found that a portion of questions can be answered by keyword matching

with co-reference resolved. To integrate the co-reference resolution part into the system, or find an alternative way to tackle it will be another important future work.

4. Description of Our System at TREC10

In the past years, we attended the 250-bytes group. Our main strategy was to measure the similarity score (or the informative score) of each candidate sentence to the question sentence. The similarity score was computed by sums of weights of co-occurred question keywords.

To meet the requirement of shorter answering texts proposed in this year, we adapt our system, and experiment on a new strategy that is focused on named entities only. The similarity score is now measured in terms of the distances to the question keywords in the same document. The MRR score is 0.145. Section 2 will deal with our work in the main task.

We also attended the list task and the context task this year. In the list task, the algorithm is almost the same as that in the main task except that we have to avoid duplicate answers and find the new answers at the same time. Positions of the candidates in the answering texts should be considered. We will talk about this in Section 4.3.

In the context task, how to keep the context, and what the answers of the previous questions can help are the main issues. In our strategy, the answers of the first question are kept when answering the subsequent questions, but the answers of the other ones (denoted by question *i*) are kept only if question *i* has a co-referential relationship to its previous one. Section 4.4 will describe this strategy in more detail.

4.1 Main Task

In the previous 250-bytes task, we measured the similarity of the question sentence and each sentence in the relevant documents, and reported the top 5 sentences with the highest scores and with the question focus words. In our experiment, the real answer sometimes lies in the sentence that is not so "similar" to the question. It becomes harder to extract text shorter than 50 bytes and containing the answer in this manner. Therefore, we experiment on another strategy, which is "candidate-focused" rather than "sentence-focused".

After reading a question, the system first decides its question type and keywords as usual. Now every named entity in the relevant documents becomes our answer candidate. For each candidate, we find out its distances to the question keywords in the same document, and sum up the reciprocals of these distances. One question keyword only contributes once, i.e., if a keyword occurs more than once, only the one nearest to the candidate contributes the score. Moreover, we assign higher weights to the keywords that are named entities. After scoring all the candidates, the highest top five are proposed, together with the texts surrounding the candidates within 50 bytes. The texts are extracted in such a way that the candidates can be placed in the middle.

In our experiment, we found that if there is a question keyword right proceeding or following the candidate, it will dominate the score despite of the other question keywords. To solve this problem, we divide the distance by three, i.e., we consider three words as a unit to measure the distance. The scoring function is shown as follows:

$$score(x) = \sum_{t \in Q \cap D} \frac{1}{\left\lceil \min\left(\left| pos_D(t) - pos_D(x) \right| \right) / 3 \right\rceil} \times weight(t) \qquad (1)$$

where x is an answer candidate, Q is the question sentence, D is the document currently

examined, t is a term occurring in both Q and D, and $pos_D(t)$ is one of the occurrence positions of t in D.

The algorithms of deciding question type and extracting named entities are the same as those in last year, which was proposed in Lin and Chen (2000). If we cannot tell which question type a question belongs to, or the question type is not concerned with a named entity, we consider every kind of entities as candidates. To extract different answers as more as possible, we ignore those answering texts whose named entity answers have appeared in the previous answering texts.

Two runs were submitted this year. When question keywords were prepared in the first run *qntuam1*, variants of ordinary words (inflections of verbs, plural forms of nouns, etc.) and named entities (adjective forms of country names, abbreviations of organization names, *etc*.) are added into the keyword bag. Stems of keywords are also added with a lower weight. Note that no matter how many variants or stems of a keyword are matched in a document, only one of them contributes the score. We select the one that can contribute the highest score.

In the second run *qntuam2*, the synonyms and explanations provided by WordNet (Fellbaum *Ed*., 1998) are also added, with lower weight to reduce the noise. Moreover, if there are m words in an explanation text, and n words occur in the document, the matching score of this explanation is defined as $\sqrt{n/m} \times weight(e)$, where $weight(e)$ is the weight of this explanation.

MRRs of these two runs are 0.145 and 0.101 under strict strategy, respectively.

4.2 List Task

List task is a new task beginning in this year. A question does not only ask for its information need but also a specified number of answers. Therefore, the system has to offer different answers to the specified number. An example is Question 1:

Question 1: Name 20 countries that produce coffee.

In this case, the system is asked to provide 20 names of different countries. Besides deciding which country produces coffee, the system also has to decide if the answer is duplicated, or if two answers are identical to each other.

The main algorithm to this task is almost the same as the main task. The only difference is that we extract the answering text in the manner that the candidates will be located at the beginning. By this way, if more than one answer appears in the same sentence, the previously proposed candidates will not appear again in the subsequent answering texts. The algorithm of the main task has already ignored the same answers (which is lexical identical), so we do not do other things to check answer identity.

Two runs were submitted as the same as those in the main task. Scores of the average accuracy are 0.18 and 0.14, respectively.

4.3 Context Task

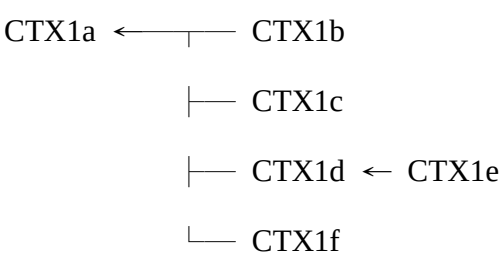
There is another new task this year. A series of questions are submitted, which are somewhat relative to the previous questions. For example, in Question CTX1:

a. Which museum in Florence was damaged by a major bomb explosion in 1993?
b. On what day did this happen?
c. Which galleries were involved?
d. How many people were killed?

e. Where were these people located?

f. How much explosive was used?

Question CTX1a asks the name of the museum. Question CTX1b continues to ask the date of the event mentioned in Question CTX1a, so this question and its answer are important keys to Question CTX1b. Question CTX1c asks more details of Question CTX1a, but irrelevant to Question CTX1b. So is Question CTX1d. But Question CTX1e refers to both Question CTX1a and CTX1d. We can draw a dependency graph of this series of questions as below:



If a question is dependent on one of its previous question, it is obvious that the information relative to this previous question is also important to the present question. Thus the system has to decide the question dependency.

We proposed a simple strategy to judge the dependency. Because the first question is the base question of this series, every subsequent question is dependent to the first one. After reading a question, if there is an anaphor or a definite noun phrase whose head noun also appears in the previous question, we postulate that this question is dependent on its previous question.

Next issue is that how we can use the dependency information in finding answers as well as its context information. After answering a single question, the system has located some answering candidates together with documents and segments of texts in which these candidates appear. Such information can be used to answer its subsequent

dependent questions, as well as the keywords of the question itself. Note that context information can be transitive. In the above example, Question CTX1e consults the information that Question CTX1d itself owns, and Question CTX1d refers to, i.e., Question CTX1a.

In our experiment, we only consider the keywords and their weights as the context information. Furthermore, we assign the lower weights to the keywords in the context information so that the importance of recent keywords cannot be underestimated. The answers to the previous question remain their weights because they are new information. The question type is decided by the present question.

The accompanying issue is that how confident an answer is included in the context information. This is because we may find the wrong answers in the preceding questions and those errors may be propagated to the subsequent questions. Moreover, do these five answers have the same weight? Or we trust the answers of the higher ranks than those of the lower ones, or only the top one is considered.

These issues are worthy of investigating, but not yet implemented in the experiment of this year. We assign weights to the previous answers according to the following equation:

$$weight(x) = weight\_NE(x) \times \sqrt{(6 - rank(x))/5} \times weight\_PreAns(x) \qquad (2)$$

where *weight_NE*(*x*) assigns higher weight if *x* is a named entity; *rank*(*x*) is the rank of *x*, and *weight_PreAns*(*x*) is a discount to the previous answers because they may be wrong. The square root part tries to assign higher weights to the higher-ranked answers.

Because only relevant documents to the first questions are provided, and we do not implement an IR system on TREC data, we cannot do a new search when answering the

subsequent questions. Our solution is to search the same relevant set of the first question.

We submitted one run this year. Its main algorithm followed the first run of the main task.

There is still no formal evaluation of this task. The MRR of all 42 question of our result is 0.139. 4 of the first questions are correctly answered. Answers of at least one of the subsequent questions can also be found in each of these 4 series. Only one of the series is fully answered.

4.4 Discussion

Comparing the results of two runs of the main task and the two runs of the list task, we can find that synonyms and explanations introduce too much noise, so that the performance is worse. However, paraphrase is an important problem in question answering. Explanation provides only one of the paraphrases, thus we have to do more researches on paraphrases.

After investigation of the results of the list task, we found that there is a small bug when reporting answers. Although duplicate answers were neglected, equivalent answers were not. In other words, adjective forms of country names were regarded as different answers to their original names, which produced redundancy and lowered the performance.

In this year, the question types of many questions are not named entities. Many of them in the main task are “definition” questions. For example,

Question 896: Who was Galileo?

Question 897: What is an atom?

In our system, we only take named entities as answer candidates, so we cannot answer such type of questions, and the performance is rather worse than that of last year.

The same problem happened in the context task, too. Therefore, it is not obvious that our proposed model to the context task is good or bad. Further investigation and experiment are needed to verify this point.

Chapter 2 Selection of Answer Candidates in Question Answering Using Information Fusion

1. Introduction

In recent years, question answering has become a popular research topic. Since 1999, TREC QA-Tracks (Voorhees, 2001) provided important evaluation test beds to develop question answering systems. There have been 1,893 questions, together with their correct answers found in the document set, as well as the surrounding text of the answers.

Answer type is important information used among most teams in TREC QA-Tracks. QA systems first analyze input questions and decide which types of answers are required. For example, if we know that a question is asking for a person, it would be better to report a personal name as an answer.

Because answer types cannot be enumerated completely, it is impossible to list all the answer types and design an answer candidate extractor for each type. In this chapter, we propose three models to extract answer candidates automatically from the corpus based on information fusion.

2. Answer Types and Candidates

Each participating team of TREC QA-Tracks has its own answer type classification. Harabagiu *et al*. (2001) encoded 38 answer types in an ANSWER TAXONOMY. Hovy *et al*. (2001) defined 140 types in the Webclopedia project. These answer types are mostly named entities, such as persons, countries, dates, plants, *etc*. The participants have to implement an answer candidate extractor, or a named entity identifier, corresponding to their own answer type classification. Here are some examples taken

from TREC QA-Track questions with their possible answer types attached at the end:

Q971: How tall is the Gateway Arch in St. Louis, MO? [LENGTH]

Q998: What county is Phoenix, AZ in? [COUNTY]

Q1228: What is the melting point of gold? [TEMPERATURE]

When the answer type of a question is decided, a QA system finds out all occurrences of terms which match this answer type, and considers them as answer candidates. The QA system will rank these candidates, and propose the most proper answer candidates will be proposed.

Answer types can be divided into two classes – say, named entities and entity sets. For named entities, we want to know the name given to a specific entity, such as "*Canada*" (a country), "*Venus*" (a planet), or "*Titanic*" (a ship), *etc*. For entity sets, what we want is a concept denoting a set of entities, such as *duck* (a kind of bird), *rose* (a kind of flower), or *dictionary* (a kind of book), *etc*.

Answer candidates of the first class are often identified by named entity recognizers for the pre-classified answer types of each QA system. Candidates of the second class need some world knowledge to capture. One possible resource is WordNet, which includes the hierarchy of entities. To answer questions like "*What kind of bird can …*?", any descendant of "bird" in WordNet can be regarded as answer candidates.

In fact, not all of the questions can be classified into pre-defined answer types. In the named-entity class, there are so many entity types which can be *named* that it is not easy to define all possible named-entity sets, not to mention to design a system to identify them all. In the entity-set class, not all terms in the world are collected in WordNet (e.g., "*birthstone*" in TREC questions). Besides, the knowledge collected in WordNet (Fellbaum, 1998) is absolute hypernymy/hyponymy relationship. For example,

WordNet does not provide relationship between "*habitat*" and "*mature tree*" in the following example:

Q217: What is the habitat of the chickadee?

Ans: oak tree, mature tree, meadow, …

Hyponyms of "*habitat*" in WordNet 1.7 is "*habitation*", which has two hyponyms: "*aerie, aery, eyrie, eyry*" and "*lair, den*". Maybe the information "*oak trees can be a habitat*" is collected in some knowledge base, but we do not know where it is.

3. Information Fusion

In question answering, there may exist a single piece of text which offers the information needed to answer a question. In such a case, we can extract the answer directly from the text. For example:

Q894: How far is it from Denver to Aspen?

Ans: 204 miles

Text: Aspen is 204 miles from Denver.

In the above example, this single passage explicitly mentions a DISTANCE-QUANTITY, and its end locations are Aspen and Denver, which exactly matches question Q894.

But there may not always exist sufficient information in a sentence to answer a question. A QA system may have to gather together pieces of information scattering in different documents or different pieces in a document in order to find the answer.

Information fusion is the process to handle pieces of information from different documents to answer a question. Sometimes the answer selection is decided from multiple pieces of texts. Sometimes there are more than one answer found in the corpus, but we have to decide whether these answers are exclusive, individual, or to be combined.

Here are some examples that information fusion has to deal with:

(1) From multiple passages to one answer

A question can be decomposed into two or more than one sub-question. For example,

Q: Where was the first president of the United States born?

It can be decomposed into two sub-questions: "WHO was the first president of the United States", and "WHERE is his birthplace". It is possible that the answers for the first sub-question may appear in many documents while the answer for the second sub-question appears in other documents.

(2) Contradictory answers

When different answers are reported, they may be contradictory. The most significant case is news stories for the same event reported in different time. For example,

Q: Who murdered Mary?

D_1 (in 1996): John was judged guilty for murdering Mary.

D_2 (in 1997): The police found new evident that Tom murdered Mary.

For QA systems, "*John*" and "*Tom*" are both effect answers from their surface texts. But there is only one true answer to this question, which is "*Tom*".

(3) Individual answers

Sometimes different answers are individually correct. For example,

Q378: Who is the emperor of Japanese?

The name of any previous Japanese Ten-On will be considered as a correct answer.

(4) Answers which have to be combined

There are two kinds of possible cases to combine answers. One is aggregation of

quantity answers. For example,

Q: How many people were killed by cancer in Europe?

A QA system first finds out the death tolls in the European countries, and gives the sum of these numbers as the answer.

The other case is summarization of multiple passages. Questions asking for opinions, methods, status, or procedures often require longer answering passages. Texts extracted from different documents contain redundant or novel information which has to be removed or added before being reported to users.

Cases 2, 3, and 4 can be regarded as "answer fusion", because the fusion is mainly done on answer part. In Case 1, information fusion is used to resolve question terms and helps to detect correct answers in the next step.

4. Automatic Answer Candidate Selection

4.1. What-Question Type

For 5W1H questions, the targets for *who*, *where*, and *when* are clearer than those of the other three. The answer for *how*-question is non-entities, so that it is not major focus of this paper. The following only considers *what*-question and *which*-question.

There are four cases of *what*-questions:

1. "*What X VP?*" or "*N V what X?*"

   E.g.Q427: "*What culture developed the idea of potlatch?*"

   E.g.Q934: "*Material called linen is made from what plant?*"

   Answer candidates are those which are *X*'s, such as cultures or plants in this example.

2. "*What be the X-NP?*"

   E.g.Q586: "*What is the chemical symbol for nitrogen?*"

Answer candidates are those which are *X*'s, such as chemical symbols in this example.

3. "*What*" alone as a subject or an object where its main verb is not *be*-verb

   E.g.Q552: "*What caused the Lynmouth floods?*"

   Its answer type does not directly appear in the question.

4. DIFINITION questions

   E.g. Q600: "*What is typhoid fever?*"

   Answers to such questions are definitions or descriptions.

In this paper, we experimented on only the first and the second cases. For the fourth case, i.e., DIFINITION questions, no answer candidates are needed to answer a question. Instead, gloss information or definition pattern is more helpful. For the third case, one possible way to find answer candidates is to gather all the terms as subjects (or objects) of this main verb. It remains future work and is not discussed in this paper.

For the first and the second cases, answer candidates are those which can be X's. If Y is the answer to a question Q "What X does something?", the information of "Y is an X" and "Y does something" may not appear in the same passage, even not in the same document. Information fusion is needed to gather these pieces of information together in order to answer such questions.

Our idea of answer candidate selection by information fusion is: find instances of *X* in a knowledge base; assign *Y* as one of the instances and check if "*Y* does something". If so, this instance is reported as an answer. Instances finding procedure is described in Section 5.

4.2. Question Focus

For a *which*-question or *what*-question in the first and the second cases, our system first identifies its *X* part, which is referred as "question focus" by Harabagiu *et al* (2000). We use this term but with slightly different meaning.

After syntactic parsing, if the word "*what*" or "*which*" alone is an NP, then it is in the second case and our system extracts the noun phrase after the *be*-verb as its question focus. If "*what*" or "*which*" is in a noun phrase with other words, it is in the first case and our system assigns its question focus as the noun phrase which "*what*" or "*which*" is in, but excludes the word "*what*" or "*which*".

Because it does not guarantee that we can find at least one instance of this question focus in the knowledge base, we have to relax the range of focus if necessary. Other possible foci are the head noun phrase of the question focus, and the remaining phrase with removing leading article, attaching propositional phrase, or any modifier. If the question focus is in the form of "*kind of NP*", "*type of NP*", or "*name of NP*", *etc*., possible focus is the noun phrase after "*of*".

In the following example, a question and its possible foci are demonstrated in sequence:

Q254: What is California's state bird?

Foci: California's state bird

state bird

bird

4.3. Corpus Candidates

DIFINITION Instances

In order to find instances of an entity set, we adopted DEFINITION patterns from Ravichandran and Hovy (2002), and from Soubbotin (2001). DEFINITION questions are a special group in question answering. Such a question asks for a definition of a term, or a description of a specific person or entity.

In Ravichandran and Hovy's system, they made experiments on six question types. One of the six question types is DEFINITION. They collected pairs of questions and the corresponding answers as examples, and automatically learned their co-occurrence patterns in the knowledge base. Some example DEFINITION patterns are listed below:

<NAME> -LRB- <ANSWER> - -RRB-

<NAME> and related <ANSWER>s

<ANSWER> -LRB- <NAME> -COMMA-

in which <NAME> denotes a question term, and <ANSWER> the corresponding answer part.

Soubbotin also used DEFINITION patterns, but they made them manually. Some examples are:

<NAME> is a <ANSWER>

<ANSWER> -COMMA- <NAME> -COMMA-

<NAME> is called <ANSWER>

The reason that we use definition to find instances is: for the instances of an entity set, the name of the entity set is just like the definition of the instances. Unlike the usage of these patterns in finding answers of DEFINITION questions, this time <ANSWER> part (the DEFINITION part) in the patterns is known (the entity set), and we'd like to extract

<NAME> part as instances.

Syntactic information is integrated into these patterns. Since answers are mostly entities, we forced the extracted <NAME> parts to be noun phrases (NP) or quantitative phrases (QP). We extracted the minimal noun phrase if there is no other text to the left or right of the <NAME> tag.

Equivalent Instances

In some cases, the name of the entity set is not the best definition of its instance. Moreover, it may not be an appropriate definition of the instance. For example, "*oak tree*" can be an instance of "*habitat*", but the definition of "*oak tree*" is "*a deciduous tree that has acorns and lobed leaves*".

To capture such instances, we further extracted equivalent entities in the knowledge base. That is, if any form of "*A* is *B*" appeared in the corpus, than we thought *A* could be an instance of *B*, or vise versa *B* could be an instance of *A*. Again, during extraction, *A* or *B* was restricted to an NP or QP.

4.4. Answer Candidates Selection Models

We experimented on three models to find answer candidates automatically. They are:

(1) Model A: Extracting Self-Evident NPs

If an NP's head is the same as the question focus, it is regarded as an answer candidate.

E.g. QFocus: artery

AnsCand: pulmonary artery

(2) Model B: Looking for WN Descendants

If a term is a descendant of the question focus in WordNet, it is considered as an answer candidate.

E.g. QFocus: color

AnsCand: red

WN: red, redness

=> chromatic color, spectral color…

=> color, colour, …

(3) Model C: Extracting Corpus Candidates

If a term in the corpus matches one of the DEFINITION patterns, or an equivalent relationship (*A* is *B*) is found, it is considered as an answer candidate.

E.g. QFocus: elephant

AnsCand: Loxodonta Africana

Pat: Loxodonta Africana (African elephants)

5. Experiments

5.1. Experiment Design

We used question sets provided by TREC QA-Tracks from TREC-9 to TREC-2000. We chose *what*- and *which*-questions, but dropped those which were asking persons, countries, cities, time, and quantity. This is because the answer candidates of these questions can be provided by a common named entity recognizer. After filtering out questions with no answer in TREC QA-Tracks, 251 questions were selected to do the experiment.

Question foci were half-automatically decided. We first parsed all the selected

questions by ApplePie Parser, then decided its what-question type as described in Section 4.1, and extracted the focus part together with all of its sub-NPs. Human effort was introduced to check errors produced by the parser in order to focus on only answer candidate problems.

When implementing Model A, top 1,000 documents of a given question were retrieved and served as a corpus to extract self-evident noun phrases related to this question. Each noun phrase with the head the same as one of the foci of the question was collected as an answer candidates. We also tested on smaller corpus, only top 100 documents, to see the coverage.

To evaluate Model B, we used the formal answers provided by TREC QA-Tracks. For a given question, we checked if one formal answer was a descendant of the question focus in the WordNet. If so, this question was counted as "covered", because all the WordNet descendant entries were regarded as answer candidates.

The corpus of Model C was created by querying Google[1]. Each question focus was submitted as a query to Google, but forced it to only retrieve documents containing the whole phrase if the question focus had more than one word. We retrieved the first 10,000 sentences containing the question focus in the top 1,000 documents. Each sentence in the retrieved corpus was matched against the DEFINITION patterns described in Section 4.3. If matched, the noun phrase in the <NAME> part and all its sub-NPs were extracted as answer candidates. Equivalent relationship was also examined.

[1] Google: http://www.google.com/

5.2. Results

The results are listed in Table 1. Self-Evident NPs (Model A) cover 62 questions in top 100 documents, and 85 in top 1000 documents. WordNet Descendants (Model B) cover 59 questions. Google Candidates (Model C) covers 54 questions.

The fourth column lists the coverage of combined models, where "A+C" denotes he combined model of Model A and C, and so on.

Table 1. Coverage of Models (in Numbers of Questions with Correct Answer Candidates)

Self-Evident NPs (top 100)	62	A+C	113
Self-Evident NPs (A)	85	B+C	92
WordNet Descendants (B)	59	A+B	120
Google Candidates (C)	54	A+B+C	137

5.3. Discussion

Interestingly, even though self-evident NPs alone have the largest coverage, these three models in fact cover different set of questions. Therefore, they can be good complements to one another. Many descendants in WordNet do not contain the same words as their ancestors, while many self-evident noun phrases are not collected in WordNet, especially those named entities. The model of Google candidates can extract named entities or senses not self-evident and not collected in the WordNet. Some examples are listed below. Each of them was extracted in only one model.

Q254: What is California's state bird?

A: quail (WordNet Descendants)

Q261: What company sells the most greeting cards?

A: the Hallmark card company (Self-Evident NPs)

Q355: What is the most expensive car in the world?

A: Bugatti Royale (Google Candidates)

The combined model of Model A and C covers 113 questions, and the combined model of Model B and C covers 92 questions. This means that model C does improve the coverage of answer candidates comparing to the coverage of Model A or B alone. Finally, the combined model of all three models covers 137 questions, which are more than a half of the testing questions.

The result of Model B is not exactly the performance using WordNet, because the formal answers provided by TREC dropped the words already occurring in the questions. For example, the answer of Q 1256: "*What is the only artery that carries blue blood from the heart to the lungs?*" is "*pulmonary artery*", which is a descendant of "*artery*". But the given formal answer is "*pulmonary*". Even so, the missing coverage of WordNet is the set of self-evident phrases. It does not affect the coverage of combined model too much.

There are many possible reasons of the low coverage of Corpus Candidates. One is that we only match patterns in top 1,000 documents. Many extracted candidates are redundant, especially those frequent entities.

The performance of DEFINITION patterns in this experiment is not yet clear. It was often that erroneous noun phrases were extracted. For example, one of the apposition patterns, "<NAME> -COMMA- <ANSWER> -COMMA-" is often mixed with the conjunction case. Further investigation of these patterns is necessary.

6. Conclusion and Future Work

In this paper, we investigated the coverage of different answer candidate extraction models. We also proposed a method to extract candidates from a large corpus. The extraction was based on the idea of information fusion, and patterns were employed to detect possible candidates.

The results of our experiments show that the three models, i.e., Self-Evident Noun Phrases, WordNet Descendants, and Corpus Candidates, have their respective coverage, and they can complement one another.

In the future, the extraction patterns of Corpus Candidates should be more carefully investigated. We will also try to find out new patterns to capture those un-answered questions.

The detection of answer candidates is very time-consuming. It will be great if such detection can be done in indexing time of IR system, and the relationships between foci and candidates can be kept in the index. It is so called QA-based indexing mentioned in Hirschman and Gaizauskas (2001).

Chapter 3 Web as a Translation Aid for Query Processing and Answer Fusion in Multilingual Question-Answering Systems

1. Introduction

Question-answering (QA) attracts much attention due to that huge heterogeneous data collection is available on the Internet. Figure 1 shows a typical multilingual QA system. A Chinese query is segmented and part-of-speech tagged. After query translation, both the original query and the translated query, e.g., Chinese and English queries, are sent to an information retrieval (IR) system. IR system retrieves the relevant Chinese and English documents. According to the foci of the query, Chinese and English answers are extracted from the relevant documents. Finally, the answers are fused and reported. This paper will show how to use the web as an aid in query translation and answer fusion.

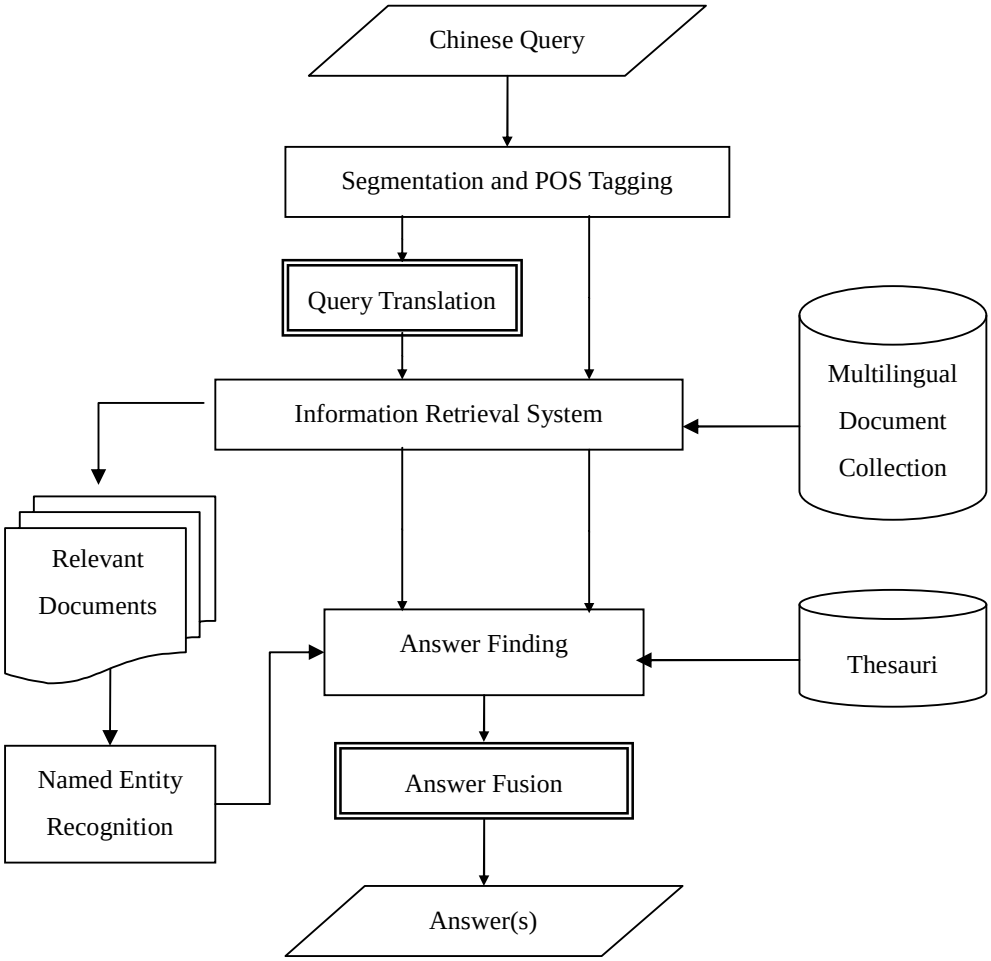


Figure 1. A Multilingual QA System

2. The Web as a Translation Aid

After bilingual dictionary lookup, those out-of-vocabulary query terms are translated by using the web as a multilingual corpus. For example, the named entity 亨利・杜南 in the Chinese query 亨利・杜南是哪一國人? (What is Jean Henri Dunant's nationality?) is an important query term, but not in the bilingual dictionary. Figure 2 demonstrates a snapshot after Google search, where snippets in a sorted sequence are returned. Figure 3 shows one of snippets in which the corresponding English translation appears. Here, a

snippet consists of title, type, body and source fields. The following depicts how to extract the translation pairs from snippets.

Figure 2. A Snapshot after Google Search "亨利・杜南"

[DOC] 香港紅十字會(青年及義工事務部)

檔案類型: Microsoft Word 2000 - HTML 版

... 為紀念本會創辦人亨利杜南先生(Jean Henri Dunant)的誕辰，每年的五月八日被定為世界紅十字日。世界各地的紅十字會均會以不同形式的活動慶祝世界紅十字日. 和推廣紅十字運動。 經歷南亞海嘯後，紅十字會之救災及備災工作更為社會各界人仕關注及 **...**

www.redcross.org.hk/wrcd2005/快訊一(final).doc - 類似網頁

Figure 3. A Snippet Containing Translation of the Named Entity "亨利・杜南"

The basic algorithm is as follows. Top-*k* snippets returned by Google are analyzed. For each snippet, we collect those continuous capitalized words, and regard them as candidates. Then we count the total occurrences of each candidate in the *k* snippets, and

sort the candidates by their frequencies. The candidates of the larger occurrences are considered as the translation of the query term.

The above algorithm does not consider the distance between the query term and the corresponding candidate in a snippet. Intuitively, the larger the distance is, the less possible a candidate is. We modify the basic algorithm as follows. We drop those candidates whose distances are larger than a predefined threshold. In this way, a snippet may not contribute any candidates. To collect enough candidates – say, *cnum*, we may have to examine more than *k* snippets. Because there may not always exist *cnum* candidates, we stop collecting when maximum (*max*) snippets are examined. Finally, the candidates are sorted by scores computed as follows.

$$score(qt,c_i)=\frac{freq(c_i)}{2}-\frac{AvgDist(qt,c_i)}{3}$$

where $score(qt,c_i)$ denotes a score function of a query term qt and a candidate c_i,

$freq(c_i)$ denotes the frequency of c_i, and

$AvgDist(qt,c_i)$ denotes the average distance between qt and c_i.

In this way, we prefer those candidates c_i of higher occurrences with the query term qt and smaller average distances.

3. Experiments

We adopt the 500 questions of TREC 2002 QA track (Voorhees, 2002), and translate them into Chinese by human. There are total 3,490 words in the 500 Chinese questions. Of these, 1,393 words are unique. After bilingual dictionary lookup, 118 words are out of vocabulary. We use them to evaluate the performance of the proposed methods in Section 2. Three metrics shown below are considered.

(1) *#correct*: total number of query terms being resolved correctly,

(2) *AvgRank*: average ranks of the correct candidates in the solved questions,

(3) *Time*: how much time taken to find all the candidates.

Figures 4-6 shows the results corresponding to these three metrics under different methods and *cnum*. Six methods shown as follows are experimented, and the factor *cnum* is tried from 10 to 100.

(1) *method1*: the basic algorithm in Section 2.

(2) *max1000*: maximum 1000 snippets (title and body) are explored in the revised algorithm.

(3) *max500*: maximum 500 snippets (title and body) are explored in the revised algorithm.

(4) *max1000_title*: max 1000 snippets are explored and only title field of a snippet are used in the revised algorithm.

(5) *max1000_quotes*: max 1000 snippets (title and body) are explored and query term is quoted in the revised algorithm.

(6) *livetrans*: the online *livetrans* system (Cheng, *et al*., 2004) are explored.

Figure 4 shows that the number of query terms being resolved correctly in the revised algorithms (i.e., methods 2-5) is increased when *cnum* is increased. After cnum$\geqq$40, the *#correct* of the four methods, i.e., *max1000_title* > *max1000_quotes* > *max1000* > *max500*, is better than the baseline and *livetrans*.

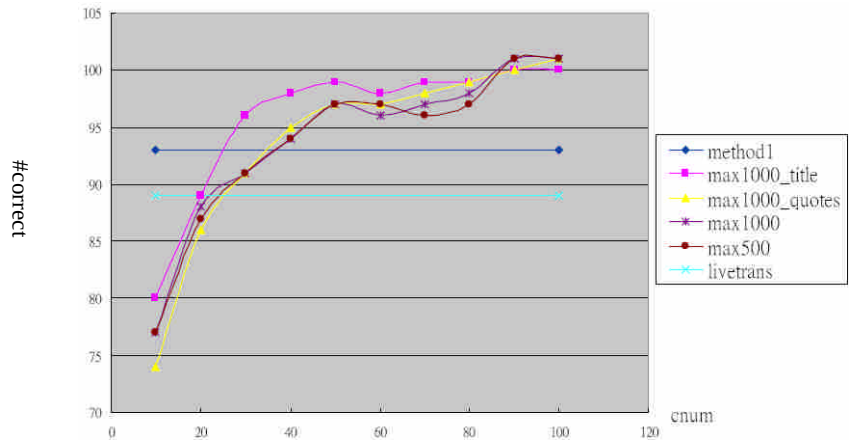


Figure 4. Total Number of Query Terms Being Solved

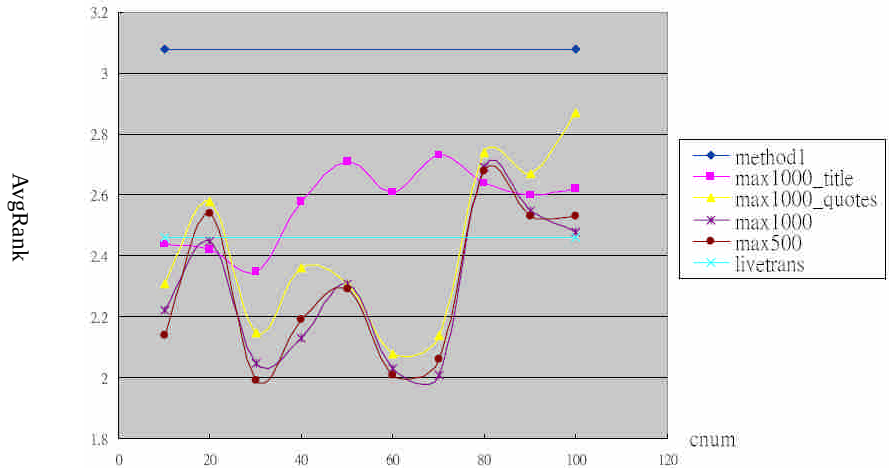


Figure 5. Average Ranks

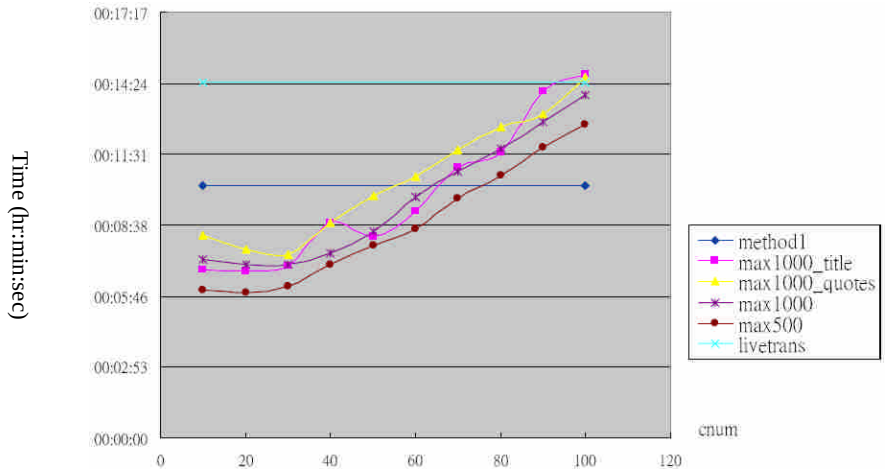


Figure 6. Time Taken

For *max1000_title* method, title field, which is a short summary of a snippet, contains important words. When terms in different language appear in title field, they often form the corresponding translation. For *max1000_quotes* method, matching a quoted query in Google requires all the query terms should appear, and their order cannot be changed. That is more concrete than matching unquoted one.

Figure 5 shows the metric of average rank. The baseline performs the worst. The two methods *max500* and *max1000* have the lower average ranks, and then *max1000_quotes* and *max1000_title*. When considering the time issue, Figure 6 shows *max500* spends the less time than all the other methods. The online *livetrans* takes more time because it tries to retrieve the relevant images besides text.

4. Extension to Answer Fusion

In a multilingual QA system, we submit a question to extract the plausible answers from a multilingual document collection. The same named entities may be reported in

different languages. For example, in the Chinese question "1997 年擔任日本首相的是誰?" (Who was the Japanese Prime Minister in 1997?), Table 1 lists the first five answers from English and Chinese document sets, respectively.

Table 1. Answers in Different Languages

Answers from English Documents	Answers from Chinese Documents
Yoshiro Mori	森喜朗
Keizo Obuchi	小淵惠三
Junichiro	陳世昌
Mori	橋本龍太郎
Ryutaro Hashimoto	官房

In this example, 森喜朗, 小淵惠三, and 橋本龍太郎 denote the same persons as Yoshiro Mori, Keizo Obuchi, and Ryutaro Hashimoto, respectively. We can merge the two sets of answers in the following way.

(1) Multiply out the English answers E_i ($1 \leqq i \leqq 5$) and the Chinese answers C_j ($1 \leqq j \leqq 5$), and generate 25 combinations.

(2) For a combination (E_i, C_j), submit E_i and C_j together to Google, and employ the similar way as the methods specified in Section 2 to verify if E_i and C_j appear in the neighborhood. If the combination has strong collocation, then delete (E_i, X) (where $X \neq C_j$) and (X, C_j) (where $X \neq E_i$), and try the remaining combinations. Figure 7 shows an example of submitting "小淵惠三 Keizo Obuchi" to Goggle. The collocation is marked in red.

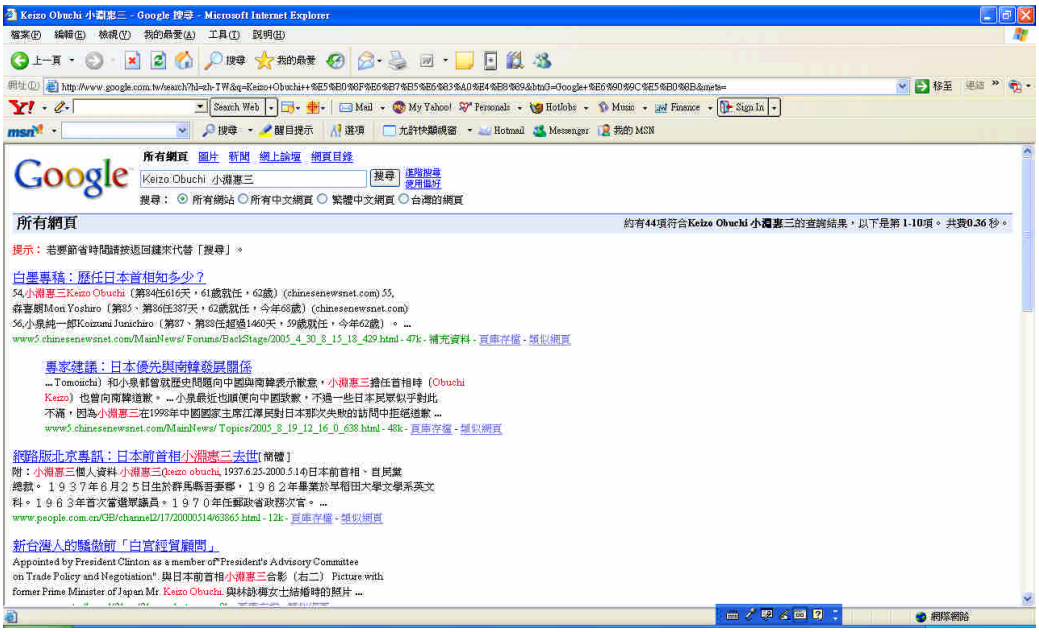


Figure 7. An Example of Submitting 小淵惠三 and Keizo Obuchi to Google

5. Conclusion

This paper employs the web as a live multilingual corpus to translate questions and merge answers in different languages. The methods of quoted query terms (*max1000_quotes*) and title only (*max1000_title*) have better coverage. The method *max500* has both better average ranks and processing speed.

Chapter 4 Open-Domain Question Answering on Heterogeneous Data

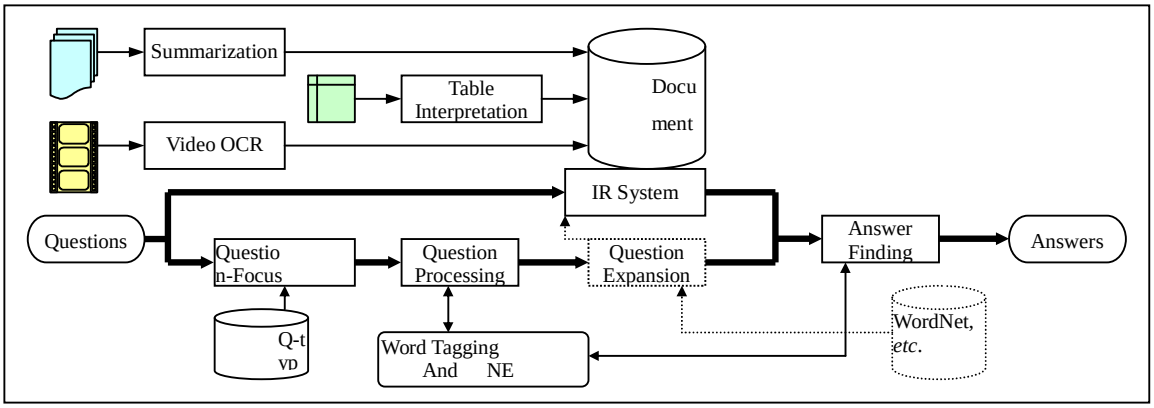
1. Introduction

Question answering has become a hot research topic in computational linguistics in recent years. QA Track in TREC has been held by NIST for two years, which has offered a new evaluation on this topic. Keyword matching was one of the major methods used among the participating groups (Moldovan, *et al*., 2000; Singhal, *et al*., 1999). Named entity information was also found important, especially, when question focus had been detected by hand-crafted rules. Many groups employed IR systems to reduce the size of documents for finding answers.

However, the target of TREC QA Track is aimed at plain text collection only. Besides, the collection is in English. Nowadays, data in different medias has become more and more popular. It is also more valuable to provide users information from heterogeneous data. Many issues arise if heterogeneous data are taken into account in a QA system:

(1) Where is the information to support answering?

In textual data, information is in text itself. Consider other kinds of data. For a table, the information is not only the texts in table cells, but also the relationships among cells. This information has to be clarified before applications. For video programs, information is carried by image, sound, speech, and captions for each frame. How to find answers in video programs becomes more challenging that that in plain text.



(2) What is the basic information unit?

In textual data, a basic unit is often defined as a sentence, a paragraph, or a passage segmented according to some linguistic information. There may be no such linguistic information in many other kinds of heterogeneous data. Therefore, the basic information unit has to be redefined for QA on heterogeneous data.

(3) What kind of questions can it be?

Since the heterogeneous data carry more information than text, many other possible kinds of questions can be issued. For example, prices are often listed in tables, so comparison between price tables becomes possible. It is also possible to ask a question where the answer is embedded in a fragment of a film.

(4) How does a QA system to measure similarity?

Most similarity measurements are based on lexical matching. We have to study the different similarity measurements for heterogeneous data.

(5) How does a QA system present answer to users?

There are more informative ways for visualizing the answers. Comparative answers can be shown in a table. Answers found in films are also shown in fragments of films.

In this paper, we propose a QA system for English/Chinese text at first, and then extend its function to handle some heterogeneous data, including summaries, tabular data, and video programs. The necessary adaptation to deal with these kinds of data is addressed. Section 2 depicts the core QA system; Sections 3, 4, 5, and 6 deal with the individual problems for plain text, summarization, tabular data, and video programs, respectively.

2. The Core QA System

The proposed QA system consists of three modules (QuestionFocus-Deciding, Question- Processing, and Answer-Finding), and an optional Question-Expansion module, together with an IR system to support IR task. Its architecture is illustrated in Figure 1.

A question is issued by a user in natural language. Question-Focus Deciding Module first decides the question focus of this question. “Question Focus” here denotes the interested information that the question requests for, such as “person name”, “reason”,

etc. Deciding question foci helps us locate answers more precisely. The construction of this module will be described in detail in Section 2.1.

In Question-Processing Module, the question sentence is word-segmented (if necessary) and POS-tagged first. The named entities in the question sentence are also identified. Only named entities and nouns, verbs, adjectives, and adverbs are kept as keywords.

The optional Question-Expansion Module will add synonyms, morpheme inflections, abbreviations of organization names, or other information of locations as keywords. Newly added keywords contribute smaller weights than the original keywords.

Moreover, an IR system is employed to retrieve relevant documents for searching answers. The advantage of using IR results is to reduce the amount of documents we have to examine. The disadvantage is that the answer texts may not appear in the so-called relevant documents, thus the answers can never be found. The model of IR system is described in Section 2.2. The IR results are transferred to the Answer-Finding Module for finding answer texts.

The Answer-Finding Module searches each passage in the relevant documents and measures its similarity to the question sentence. If a passage contains more information in the question and the interested type of question focus, then it is more likely to carry the answer information and is ranked higher. Section 2.3 shows how the answers are extracted in detail.

2.1 Question-Focus Deciding Module

The patterns of question sentences are quite different from Chinese to English. In English, 5W1H is the main question words. Patterns can be hand coded including these question words (Moldovan, *et al*., 2000; Singhal, *et al*., 1999).

In Chinese, we do not yet have the information of question words, together with question patterns. For example, there are at least three kinds of ways to express "what": "何", "什麼", and "甚麼". There are few researches on questions for Chinese language processing. Chang (1997) analyzed questions in Chinese, and classified them into seven categories. But her classification was based on the functions of questions in discourse, not on the question foci.

In order to find all the possible question words, question patterns, and their mapping to the question foci, we conducted an experiment. All the questions in Academia Sinica Balanced Corpus (Chen, *et al*., 1996) were extracted. The question words and question foci were hand-tagged in these 16,851 sentences. We defined nine question foci, and one more category NOFOCUS for the questions which are functionally not requesting information. Appendix A lists the question foci and some examples of the hand-tagged questions.

Question-Focus Decision rules were trained by C4.5 (Quinlan, 1993). Question words and the terms preceding or following them were selected as features. Question words occurring less than 4 times and preceding (following) terms occurring less than 100 times were discarded while training. We got 200 rules with 81.5% correctness. Appendix B lists some of the Question-Focus Decision rules.

2.2 IR system for Question Answering

Our IR system was based on vector space model (VSM). In English, index terms are words except stop words; in Chinese, index terms are bigrams of Chinese characters and English strings (if any).

After examining some retrieval results, we found it was useful to integrate Boolean model into IR system for a better QA performance. This is because every keyword in the question sentence is equally precious in QA task. The more keywords being included in a document implies the higher possibility to find the answer in that document. Therefore, we employ the Boolean score as the first sorting key, and the VSM score as the second key while ranking relevant documents.

2.3 Answer-Finding Module

The Answer-Finding Module searches each passage in the relevant documents and measures its similarity to the question sentence. Similarity is defined as the sum of weights contributed by the terms matched:

$$sim(Q,P) = \sum_{t \in Q \bigcap P} weight(t) \qquad (1)$$

Where Q is the question, and P is a passage in the relevant documents. Named entities

contribute higher weights than the original question terms, because named entities often carry more information. The expanded terms contribute lower weights to reduce the noise that might be introduced.

A passage can be chosen as a sentence, a meaningful unit that carries the smallest piece of information, or a video segment, depending on the data type we are processing. In Sections 3, 4, and 5, "passage" will be defined for different media, respectively.

The answerable passages were ranked in the order of their similarity scores. In other words, those passages meeting the question focus were reported first.

3. QA for Plain Text

The passages selected for plain text are sentences. We made experiments on both English and Chinese documents.

3.1 Experiment on English

In English experiment, we conducted an experiment as the same as QA Track in TREC-9 (Voorhees, 2000). There are 693 questions to be answered. We gave five 250-byte-length answers for each question. The metric is MRR (Mean Reciprocal Rank):

$$MRR = \sum_{i=1}^{N} r_i \Big/ N \ , \qquad (2)$$

where $r_i = \begin{cases} 1/rank_i & rank_i > 0 \\ 0 & rank_i = 0 \end{cases}$, $rank_i$ is the rank of the first correct answer of the i^{th} question, and N is total number of questions. That is, if the first correct answer is at rank 1, the score is 1/1=1; if it is at rank 2, the score is 1/2=0.5, and so on. If no answer is found, score is 0.

After evaluating by hand, the MRR is 0.348, and 354 (51.9%) questions failed to find any answers.

3.2 Experiment on Chinese

In Chinese, the test data is collected from 6 news sites in Taiwan through the Internet. There are total 17,877 documents (near 13MB) from January 1, 2001 to January 5, 2001.

In order to compare with the multi-document news summarization work in Section 4, we concatenated the articles of the same news event into one article, and took these event articles as our document collection for experiment. After clustering, there are 3,146 events.

Questions were formulated by research assistants. We deliberately prepared two kinds of questions to see how these QA models work in different situations. One group of questions is the ones lexically similar to the answer texts, and the other is not similar. After filtering out the questions that had no answers in the test collection, there were 127 answer-like questions and 96 not-answer-like questions. Examples of these questions are listed in Appendix C.

We gave five sentences as answers for each question. After evaluating by hand, the MRR is 0.62, and 52 (23.3%) questions failed to find any answers. Table 1 depicts the comparison of the two sets of questions.

Table 1. Plain-Text QA Results

	All	Answer-like Questions	Not-Answer-like Questions
MRR	0.6243	0.6790	0.5519
No Answer	52 (23.3%)	25 (19.7%)	27 (28.1%)

4. QA on Summarization

Summarization is a kind of data that can be served as a knowledge base for question answering. In Internet, some web sites provide only summaries for retrieval. Besides, search engines reply fragments of texts as summaries to the relevant documents. Multi-document summarization is also necessary for users to reduce the reading time. Therefore, summaries will be a good resource to find information.

Many papers have touched on single document summarization (Hovy and Marcu, 1998a) and multiple document summarization (Chen and Huang, 1999; Chen and Lin, 2000; Mani and Bloedorn, 1997; Radev and McKeown, 1998). We employed our multi-document summarizer on the news articles describing the same event (i.e., in the same cluster).

Because summaries are in text, techniques developed to deal with plain text can be fully adopted here. Doing QA on summaries has its trade-off. The main advantage is that it saves both time and space. In our experiment, it took 12,202 seconds to search the full text for the answers of 223 questions, but it only took 1,100 seconds to search in the summaries. After adding the time of doing summarization, it took only 1,366 seconds, in which case it saved 9/10 of the time as full text did.

Table 2. Summarization QA Results

	All	Answer-like Questions	Not-Answer-like Questions
MRR	0.4298	0.5051	0.3302
No Answer	104 (46.6%)	49 (38.6%)	55 (57.3%)

The QA results on summarization are listed in Table 2, in the same format as Table 1. If the answer information happens to be the main information of the document, summarization can help to increase the probability to find the answer, because unimportant information has been dropped. But if it is not, generic summarization will drop out answering texts. Therefore, the study of how query-based summaries, instead of generic (Hovy and Marcu, 1998b), can keep the answers is a topic for future research.

5. QA on Tabular Data

Tables, which are simple and easy to use, are very common presentation scheme for writers to describe schedules, organize statistical data, summarize experimental results, and so on, in texts of different domains.

Because tables provide rich information, table acquisition is useful for many applications including question answering. Previous researches on table extractions mainly targeted on plain texts (Hurst and Douglas, 1997; Ng, Lim and Koo, 1999). Only Hurst (1999) and Chen, *et al*. (2000) dealt with HTML-tagged tables.

The information given by a table is not only the texts in table cells, but also the relationships among cells. We used the method (Chen, *et al*., 2000) to distinguish

attribute cells from value cells in a table. Their relationship can be interpreted by attributes-value sequence shown as follows:

[attrib 1][attrib 2]…[attrib n][value]

Each interpretation contains enough information and can be regarded as a basic information unit.

In testing, we collected 3,123 html documents from the web sites of airline companies and on-line travel service companies. The documents are in Chinese. There were 14,884 table tags found in the collection, and 1,777 of them were judged as real tables.

Total 40 Questions are formulated on the document collection. An attributes-value sequence is regarded as a sentence while measuring similarity in QA. MRR is 0.36, and 19 (48.7%) questions were not answered. We also tested on the original data without table interpretation. MRR is 0.27, and 25 (64.1%) questions were not answered. The result showed that the table interpretation is useful for QA task.

6. QA on Video Films

Data in motion pictures are also a good resource for finding answers. We can search information in the films, or ask a question and request the answers in the video. That is very useful in the network era because data in the digital libraries/museums, or images of TV news and programs are available.

To extract answers from video programs is quite different from the task done on texts. We have to consider the issues of information source, the basic information unit, and the similarity between text and image.

6.1 Video OCR

Because captions in video programs are the transcriptions of narratives and dialogues, and are the basis of the stories of films, we employed them in QA task. Besides, it is also easier to handle captions than images.

To define a basic information unit, pause duration is a good clue. Long duration implies that old information is over and new information will be cited. Since captions

appear along with the voice, we can take the duration time of captions to do the segmentation.

Extracting captions is like doing OCR on video images. Some researches have been done (Sato, 1999), but few were tested on Chinese captions. Here, a Video OCR system presented by Liu (2001) was adopted.

6.2 Adapted Word Similarity

Since Video OCR has not reached perfect performance, it is not reasonable to measure similarity only by character matching. OCR similarity for character images is also considered.

In QA for textual data, if two words qw_i (in question) and pw_j (in passage) are matched, they contribute a score of $1 \times weight(qw_i)$. To take OCR similarity into account, the contributed score is modified as below:

$$score(qw_i, pw_j) = 0 \quad \text{if } |qw_i| \neq |pw_j| \text{ else} = \left(\sum_{k=1}^{|qw_i|} Ocr(qc_k, pc_k) \Big/ |qw_i| \right) \times weight(qw_i) \quad (3)$$

where $|qw_i|$ denotes the number of characters in qw_i, and qc_k is the k^{th} character in qw_i (the same convention for pw_j). Ocr(qc_k, pc_k) is the OCR similarity of characters qc_k and pc_k.

6.3 QA Experiment on Video OCR

The test data come from 19 Discovery programs, about 298KB, which are pronounced in English and with Chinese captions. Each program is about 1 hour long. Total 15,353 lines of captions are extracted. On average, there are 808 lines in a program.

A basic information unit of a film is considered as a passage for extracting answers. The passages are segmented at pause duration longer than 5 seconds in our experiments. There are totally 3,876 passages. On average, there are 204 passages in a program.

39 questions are collected from the web site of Discovery Channel (http://chinese. discovery.com/sch/), as links in the program list page. They offer questions according to each program for educational purpose. The QA results are not good. Original model

answered 3 questions, while OCR-similarity- integrated model answered one more. The main reason is that these questions are pretty hard in question level (Moldovan, *et al*., 2000).

7. Conclusion

This paper sketches a new view of question answering on heterogeneous data. Table 3 compares the heterogeneous data in QA task. After defining information passages and similarity measurement, our QA system is capable of handling data consisting of plain texts, summaries, HTML documents with tables, and videos.

Table 3. Comparison of Heterogeneous Data

Plain text	**Summary**	**Table**	**Video**
Document	Document	Document and Table	Film, Captions
Sentence, Passage	Sentence, Passage	Interpretation, Value-Cells	Film Fragment Divided by Pause
Lexical Matching	Lexical Matching	Lexical Matching	Lexical Matching and OCR Similarity
Presented as Text	Text	Text or Tables	Film Fragment

There are several interesting future directions, for example, how query-based summarization can be helpful a QA task, how to integrate the context of tables, and so on. Besides, background linguistic technologies for OCR texts, such as word segmentation, IR, and named entity extraction, have to be redefined.

Appendix A.

(1) Question Foci

PERSON, LOCATION, TIME, QUANTITY, SELECTION, METHOD, DESCRIPTION, REASON, and OBJECT.

(2) Hand-Tagged Questions

These are some examples of hand-tagged questions for training Question-Focus

decision rules. Boxed texts are question words. A question focus is given in front of each question, and is printed in bold.

LOCATION 草嶺谷 在 哪裡 ？

(Where is Grass Valley?)

TIME 臺灣 歷史 從 什麼 時候 開始呢 ？

(When did Taiwan history start?)

METHOD 怎樣 增加 鈣 的 吸收 ？

(How to improve the absorption of Calcium?)

Appendix B.

Question Focus Decision Rules

These are some examples of Question-Focus decision rules. "Term" is the question word found in the sentence, and TermNext (TermPrev) is the term following (preceding) the question word.

Rule 3: Term= 何處(where)-> class LOCATION

Rule 17: Term = 誰(who)-> class PERSON

Rule 21: Term = 如何(how), TermNext = 來(to come, to do)-> class METHOD

Appendix C.

Chinese Questions for Experiments on Plain Text and Summarization.

Q1. 中國第一縷曙光落在哪裡？

(Where does the first sunlight shine on China?)

Q77.「英雄」為何種類型遊戲？

(What kind of game is "The Hero"?)

Q280. 21世紀的明星產業將會是什麼？

(What will be the star industry in the 21st century?)

References

Chang, C.Y. (1997) *A Discourse Analysis of Questions in Mandarin Conversation*, Master Thesis, National Taiwan University, June 1997.

Chen, H.H. and Huang, S.J. (1999) "A Summarization System for Chinese News from Multiple Sources," *Proceedings of 4th IRAL*, Taiwan, pp. 1-7, 1999.

Chen, H.H., Tsai, S.C., and Tsai, J.H. (2000) "Mining Tables from Large Scale HTML Texts," *Proceedings of 18th COLING*, pp. 166-172, 2000.

Chen, H.H. and Lin, C.J. (2000) "A Multilingual News Summarizer," *Proceedings of 18th COLING*, pp. 159-165, 2000.

Chen, K.J., Huang, C.R., Chang, L.P., and Hsu, H.L. (1996) "Sinica Corpus: Design Methodology for Balanced Corpora," *Proceedings of the 11th PACLIC 11*, pp. 167-176, 1996.

Pu-Jen Cheng, Jei-Wen Teng, Ruei-Cheng Chen, Jenq-Haur Wang, Wen-Hsiang Lu, and Lee-Feng Chien (2004) "Translating Unknown Queries with Web Corpora for Cross-Language Information Retrieval," *Proceedings of the 27th ACM-SIGIR*, pp. 146-153.

Christiane Fellbaum (Ed.) (1998) WordNet: An Electronic Lexical Database, The MIT Press, 1998.

Sanda Harabagiu, Dan Moldovan, Marius Pasca, Rada Mihalcea, Mihai Surdeanu, Razvan Bunescu, Roxana Girju, Vasile Rus, and Paul Morarescu (2001), "The Role of Lexico-Semantic Feedback in Open-Domain Textual Question Answering," *the Proceedings of the 39th ACL and 10th EACL*, pp. 274-281, 2001.

Sanda Harabagiu, Marius Pasca, and Steve Maiorano (2000), "Experiments with Open-Domain Textual Question Answering," the Proceedings of the 18th COLING, pp. 292-298, 2001.

Lynette Hirschman and R. Gaizauskas (2001) "Natural Language Question Answering:

the View from Here," Natural Language Engineering, Cambridge University Press, Vol. 7, No. 4, 2001, pp. 275-300.

Eduard Hovy, Ulf Hermjakob, and Chin-Yew Lin (2001), "The Use of External Knowledge in Factoid QA," the proceedings of TREC 2001, pp. 644-652, 2001.

Hovy, E. and Marcu, D. (1998a) *Automated Text Summarization*, Tutorial in *17th COLING-ACL*, Montreal, Quebec, Canada, 1998.

Hovy, E. and Marcu, D. (1998b) *Multilingual Text Summarization*, Tutorial in *AMTA-98*, 1998.

Hurst, M. (1999) "Layout and Language: A Corpus of Documents Containing Tables," *Proceedings of AAAI Fall Symposium*, 1999.

Hurst, M. and Douglas, S. (1997) "Layout and Language: Preliminary Experiments in Assigning Logical Structure to Table Cells," *Proceedings of ANLP '97*, pp. 217-220, 1997.

Lin, C.J. and Chen, H.H., "Description of NTU System at TREC-9 QA Track," *Proceedings of The Ninth Text REtrieval Conference* (*TREC-9*), 2000, pp. 389-406.

Liu, C.C. (2001) *Video OCR and Video Search*, Master Thesis, National Taiwan University, 2001.

Mani, I. and Bloedorn, E. (1997) "Multi-document Summarization by Graph Search and Matching," *Proceedings of 4th National Conference on Artificial Intelligence*, Providence, pp. 622-628.

Moldovan, D., Harabagiu, S., Pasca, M., Mihalcea, R., Girju, R., Goodrum, R., Rus, V. (2000) "The Structure and Performance of an Open-Domain Question Answering System," *Proceedings of 38th ACL*, pp. 563-570, October 2000.

Ng, H.T.; Lim, C.Y. and Koo, J.L.T. (1999) "Learning to Recognize Tables in Free Text," Proceedings of *37th ACL*, pp. 443-450, 1999.

Quinlan, J.R. (1993) *C4.5: Programs for Machine Learning*, Morgan Kauffman, 1993.

Deepak Ravichandran and Eduard Hovy (2002), "Learning Surface Text Patterns for a Question Answering System," *the Proceedings of ACL*, 2002.

Radev, D.R. and McKeown, K.R. (1998) “Generating Natural Language Summaries from Multiple On-Line Sources,” *Computational Linguistics*, Vol. 24, No. 3, pp. 469-500, 1998.

Sato, T., Kanade, T., Hughes, E. K., Smith, M. A., and Satoh, S. (1999) “Video OCR: Indexing Digital News Libraries by Recognition of Superimposed Captions,” *Multimedia Systems*, Vol. 7, pp. 385-394, 1999.

Singhal, A., Abney, S., Bacchiani, M., Collins, M., Hindle, D., Pereira, F. (1999) “AT&T at TREC-8,” *Proceedings of TREC 8*, Gaithersburg, pp. 317-330, November 1999.

M. M. Soubbotin (2001), “Patterns of Potential Answer Expressions as Clues to the Right Answers,” *the Proceedings of TREC 2001*, pp. 293-302, 2001.

Ellen Voorhees (2000) “QA Track Overview (TREC) 9,”[on-line] Available: http://trec.nist.gov/ presentations/TREC9/qa/index.htm

Ellen Voorhees (2001) “Overview of the TREC 2001 Question Answering Track,” the Proceedings of TREC-10, pp. 42-51, 2001.

Ellen Voorhees (2002) “Overview of the TREC 2002 Question Answering Track,” *Proceedings of the Eleventh Text Retrieval Conference*, Gaithersburg, Maryland, November 19-22, 2002.