# Search Computing Based on Google API for QA System

Zhi Teng[1], Ye Liu[1], Fuji Ren[1,2] and Shingo Kuroiwa[1]

[1] Faculty of Engineering, The University of Tokushima
2-1 Minamijosanjima, Tokushima, 770-8506, Japan

[2] School of Information Engineering, Beijing University of
Posts and Telecommunications, Beijing, 100876, China
{teng, ren, kuroiwa}@is.tokushima-u.ac.jp

**Abstract** Search computing has been widely used in the field of natural language processing. In recent years, the QA System has been successfully applied to a number of applications, but the limited amounts and the incertitude of answer were much in evidence. The wealth of information on the web makes it an attractive and simple resource for seeking quick information. Recently being quite successful in providing keyword based access to web pages, commercial search portals still lack the ability to answer questions expressed in a natural language of Chinese. In this paper we propose a new method based on the GoogleWEB API for the QA System in restricted domains. The experiment showed that this method can get the more accurate result.

**Keyword** Search computing, natural language processing , QA System, Google WEB API

## 1. Introduction

The QA System has been successfully applied to a number of applications, but the limited amounts and the incertitude of answer were much in evidence. Although the QA system can provide the answer the user can't know whether the answer is right or wrong, so if we can provide a very comprehensive explanation of answer by web page using the information on the web when the user put question to the QA system, then the user can judge the answer by themselves. In our experiment we use a Web search engine and a robust parser for the QA System in restricted domains. It can afford a very comprehensive explanation of answer by web page use the information on the web.

With the information revolution well under way, the degree of communication and number of communication methods is growing rapidly. People converse frequently via a number of mediums. One such medium is Internet chat using various instant messaging clients (e.g., AOL Instant Messenger, MSN Messenger, etc). These communications provide an excellent platform to perform research on informal communications. The wealth of information on the web makes it an attractive resource for seeking quick answers to simple, factual questions such as \"who was the first American in space?" or \"what is the second tallest mountain in the world?" Yet today's most advanced web search services (e.g., Google and AskJeeves) make it surprisingly tedious to locate answers to such questions [1]. The goal of question answering (QA) is to identify and present to the user an actual answer to a question formulated in a natural language, rather than identifying documents that may be topically related to the question or may contain the answer [2]. Thus in our experiment, we try to process the text snippets returned by the search engine from the web, extract and evaluate the information of answer, finally show the answer in a natural language.

In recent years, the Internet technology for application-to-application communication referred to as the web service is improving at a rapid rate. For example, Google, a popular Internet search engine, provides the web service called the Google WEB API [3] [4]. The service enables users to develop software that accesses and manipulates a massive amount of web documents that are constantly refreshed. In our system, we use the Google WEB API as the search engine. We think the method of Google WEB API is the best, so we put reliance on the result of search by it. In our experiment we only devote all energies to process the text snippets returned by the Google WEB API, extract and evaluate the information of answer.

The remainder of the paper is organized as follows. Chapter 2 describes the overall architecture of our QA system environment. Chapter 3 describes the document search. Chapter 4 describes the article ectraction. Chapter 5 describes the answer extracton. Chapter 6 describes the answer evaluation. Chapter 7 describes the media building. Chapter

8 describes the Experimentation and Chapter 9 gave the conclusions.

## 2. Overall Architecture

In our experiment we developed a QA System that uses the Web as a Corpus in the field of personality search.The overall framework of our QA system environment is presented in Figure 1. QA system environment consists of four major parts: (I) document search, (II) article ectraction, (III) answer extraction and (IV) answer evaluation.

To answer user question, firstly, our system search the similarity docements from web use the user question by Google WEB API then our system find out the important information about uesr question from the simailarity docements. System will extract the especial keyword form the inportant information. Finally, evaluation the keyword and return the keyword as answer.

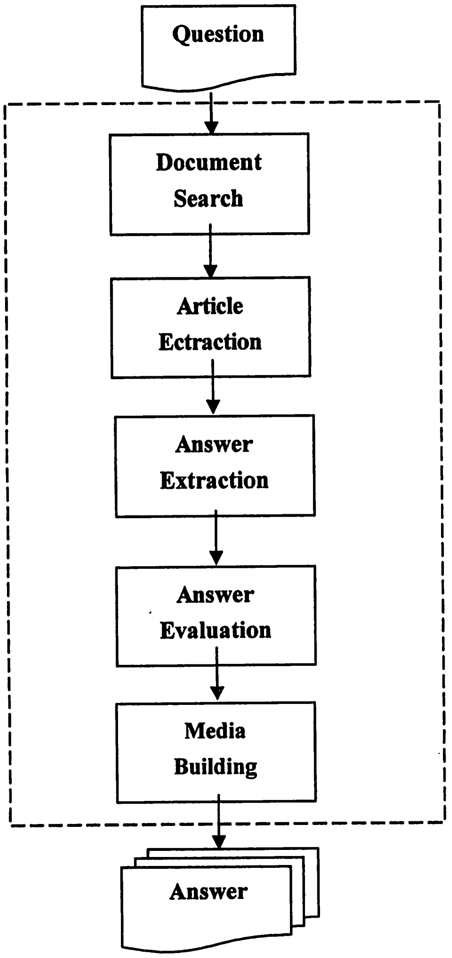


Figure 1: Overview of QA System

## 3. Document Search

The Google WEB API

Our QA system depend on the web as its knowledge base, the choice of search engine essentially determines our scope of knowledge, and the method of retrieving that knowledge. The idiosyncrasies of the search engine also affect more than just the syntax of the queries. For example, with Boolean search engines one could issue a query A AND ( B OR C ), but for search engines which do not support disjunction, one must decompose the original query into two, A B and A C, in order to get the same set of pages. We considered a few search engine candidates, but eventually chose Google. Google has two overwhelming advantages over others: it has the widest coverage among search engines (as of October 2000, Google has indexed 1.2 billion web pages), and its page ranking function is unrivaled. Wider coverage means that the system has a larger knowledge base and access to more information. Moreover, with a larger collection we have a higher probability of finding target sentences. Google's ranking function helps the system in two ways. First, Google's ranking function is based on the PageRank [5] and the Hits algorithms [6], which use random walk and link analysis techniques respectively to determine sites with higher information value. This provides the system with a selection of higher quality pages. Second, Google's ranking function is also based on the proximity of words, i.e., if the document has keywords closer together, it will be ranked higher. While this has no effect on queries with long phrases, it helps significantly in cases when we have to rely on keywords.

A small portion of the code of Google search client api for c/c++ , as below:

```
struct typens__ResultElement
{
     xsd__string  summary;
     xsd__string  URL;
     xsd__string  snippet;
     xsd__string  title;
     xsd__string  cachedSize;
     xsd__boolean  relatedInformationPresent;
     xsd__string  struct typens__DirectoryCategory *
directoryCategory;
     xsd__string  directoryTitle;
};
```

```
if(
doGoogleSearch
 (
   m_handle,
   strq,
   0,
   3,
   maxResults,don't modify it.
   false,
   "",
   false,
   "lang_zh-CN",
   "utf-8",
   "utf-8",
   &out
  )==0
)
for(int i=0;i<out._return_->resultElements->__size;i++)
{
 URL[i]=out._return_->resultElements->__ptr[i].URL;
 title[i]=out._return_->resultElements->__ptr[i].title;
 snippet[i]=out._return_->resultElements-
>__ptr[i].snippet;
 };
```

## 4. Article Extraction

Through the function of these code, we can get some information of the summary, URL, snippet, title, cachedSize and so on. (You can see something about google WEB API in [7]) . The system searches the possible information according to the query by the search engine from the web. Through the Google search client api we can get the information like:

*His Universal Printer, <b>invented</b> at this time, printed full information about gold<br> prices, instead of showing them only <b>...</b> In January,1880, the <b>electric light</b><br> was patented. Edison then built a factory for the production of his...*

In here, we must delete the sign like *<b>*, *</b>* and *<br>* etc. We only extract the data from snippet for answer extraction

Here, we introduce the experiments through one instance:

"谁发明了电灯？"

"Who invented the electric light?"

In next step, the text snippets must be processed by the morphological analysis program. The morphological analysis program can't process the sign of Spacebar, so here we must delete the Spacebar from the text snippets. In the end, the text snippets are like:

*His Universal Printer, invented at this time, printed full information about gold prices, instead of showing them only In January,1880, the electric light was patented. Edison then built a factory for the production of his...*

## 5. Answer Extraction

Our experiment is tested in field of personality search, so the answer is a name of a personality. Here we used the morphological analysis program to extracte the word of personality.

The word of the name is labeled as "/nr" by the morphological analysis program through the rule database and analyzer rules, so we only extract the word which in front of the sign of "/nr". For the same instance:

*...In January,1880, the electric light was patented. Edison then built a factory for the production of his...*

The snippets of the text to be processed by the morphological analysis program like as:

...In/p January/t 1880/m the/u electric/n light/n was/vd patented/v Edison/nr then/ad built/v a/q factory/n for/p the/u production/n of/p his/r...

In here we extract the Edison/nr.

## 6. Answer Evaluation

In our experiment we extracted the answer from the snippets of the text which the first ten result by the Google search client api. We must to evaluate the answer that have been extracted and then display the best answer which have the highest degree of points. We set two steps to evaluate the answer: Frequencies and Keyword.

### 6.1 Frequencies evaluation

The answer is a name of a personality. The method is that the system will measure the frequencies of name in the first ten snippets of the text by the Google search client api. Equation 1 shows the calculation of the Frequency

$$W_{(n,t)} = \sum_{i=1}^{n} F_t \qquad (1)$$

In equations 1 $W_{(n,t)}$ is the frequencies of name t appear in snippets; n is set to 10; Ft is the frequencies of name t appear in snippets i. Compute the $W_{(n,t)}$ value of all name t and find out the answer which has the maximal value of the $W_{(n,t)}$ and the best answer will be obtained.

Though the method is easy to achieve, it has disadvantages. Because the Google WEB API tries to find the most similar data to the question. Sometimes the data by search engine from the web is like the question, but the meaning is different. For instance, "Who invented the electric light?" ( "谁发明了电灯？ " )but the result by search engine is "Who invented the telegraph?" ( "谁发明了电报？ ). In this instance, it would have an effect on the accuracy of answer, so after the frequencies evaluation we use the Keyword evaluation.

### 6.2 Keyword evaluation

In our experiment, we think that the best important word in the question is the noun, so we regard the noun as the Keyword. We process the question by the morphological analysis program. For instance, the result is like as: "Who/r invented/v electric/n light/n". We extract the word of "electric" and "light" and then we search the "electric" and "light" in the first ten title of the text snippets that searched out by the Google search client api and evaluate the ten snippets. The measurement is as follows:

$$W_{(m,t)} = \sum_{j=1}^{m} F_k \qquad (2)$$

In equations 2 $W_{(m,t)}$ is the frequencies of all keyword appearing in snippets that include the name t; m is the number of the snippets that include the name t; $F_k$ is the frequencies of all keyword that appears in snippets j. Compute the $W_{(m,t)}$ value of all name t.

The measurement of Frequencies and Frequencies is as follows:

$$W_t = W_{(n,t)} + W_{(m,t)}/2 \qquad (3)$$

In equations 3 $W_t$ is the evaluation result of name t.

## 7 Media Building

The information feedback mode of a QA system is very important. A QA system should not only use the text to talk with the user but also can use the media about the speech, the picture, the video, the WEB page etc. In our system we used three kinds of media: the text, the speech and the WEB page. We display the result web page and finally answer in a natural language and speech. The configuration of media is presented in Figure 2.

Text:

As the general program, this system receives the input text from the keyboard and output by the text.

Speech:

About the speech interface to conversation system, some researches appear to be at the forefront of this field. James, E. & Zheng, Z. [8] describe a multimodal interface to a open domain QA system designed for rapid input of questions using a commercial dictation engine, and indicate that speech can be used for automatic QA system by their evaluation. On speech-driven dialogue system of Chinese, Zhang et al. [9] described two Chinese spoken dialogue systems about real-time stock market quotations inquiry and Shanghai Traffic Route respectively, which used a model of situation semantic frame-key technology.

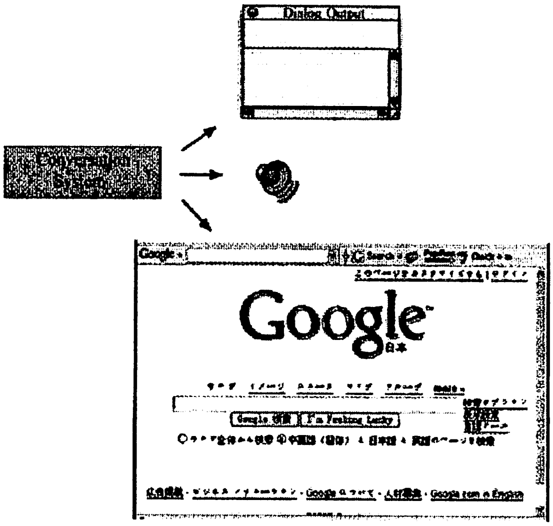


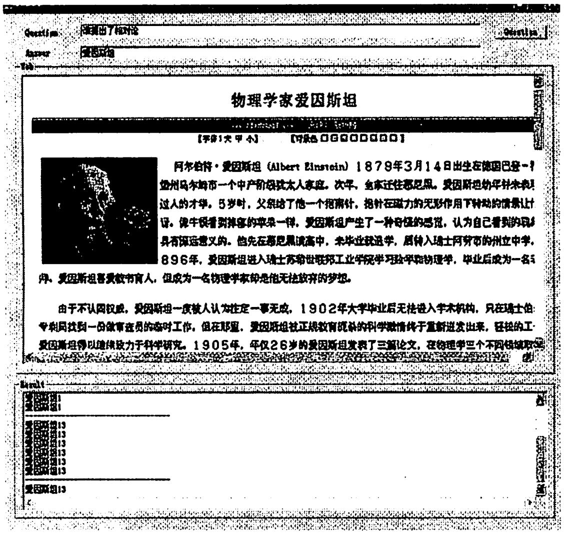
Figure 2: The configuration of media

In this system, as a part of conversation system technical research, the object is the Speech Synthesis. A speech interface for the conversation system was implemented along with the text interface, using an acoustic model HMM (Hidden Markov Model), a pronunciation lexicon and a language model FSN (Finite State Network) on the basis of the feature of Chinese sentence patterns[10].

In this system, it is necessary to remove a lot of unknown characters for the speech synthesis of the conversation content, since the conversation content that is retrieved comes from the web. Therefore, the system executes speech synthesis on the conversation content by result of morphological analysis allowing the quality of speech synthesis to be advanced [11].

WEB Page:

Through the web page the user can judge whether the answer is right or wrong by themselves and understand some others information. The interface of system is presented in Figure 3.



Question: 谁提出了相对论

Answer: 爱因斯坦

物理学家爱因斯坦

阿尔伯特·爱因斯坦（Albert Einstein）1878年3月14日出生在德国巴登-符登州乌尔姆市一个中产阶级犹太人家庭。次年，全家迁往慕尼黑。爱因斯坦幼年并未表过人的才华。5岁时，父亲给了他一个指南针，指针在磁力的无形作用下转动的情景让讶。像牛顿看到掉落的苹果一样，爱因斯坦产生了一种奇怪的感觉，认为自己看到的现具有深远意义的。他先在慕尼黑读高中，未毕业就退学，后转入瑞士阿劳市的州立中学。896年，爱因斯坦进入瑞士苏黎世联邦工业学院学习数学和物理学，毕业后成为一名师。爱因斯坦喜爱教书育人，但成为一名物理学家却是他无法放弃的梦想。

由于不认同权威，爱因斯坦一度被人认为注定一事无成。1902年大学毕业后无法进入学术机构，只在瑞士伯专利局找到一份做审查员的临时工作。但在那里，爱因斯坦被正规教育扼杀的科学激情终于重新迸发出来，轻松的工爱因斯坦得以继续致力于科学研究。1905年，年仅26岁的爱因斯坦发表了三篇论文，在物理学三个不同领域取

爱因斯坦1
爱因斯坦1

爱因斯坦13
爱因斯坦13
爱因斯坦13
爱因斯坦13
爱因斯坦13
爱因斯坦13

爱因斯坦13

Figure 3: The interface of QA system

## 8. Experiment

### 8.1 Test Data

In our Experiments, we invited 16 students who will ask 15 questions for every one, totally 240 different questions. Before the students ask the question, they didn't know the theory of this system. They ask whatever that they want to ask about personality and then the system process the question in the end do the manual evaluating on result of experiment. Some example of test data is showed as below:

*Who invented the first electric light?*

*Who invented the telephone?*

*Who discovered America?*

*Who Invented the Thermometer?*

*Who invented the automobile?*

*Who Discovered Quantum and Particle Physics?*

### 8.2 Test Result

Before this experiment we have graded the answer from the internet according to the validity. They are "Excellent", "Good", "Wrong" and "No Answer".

Excellent: the answer is right.

Good: the answer is possibly right. (Can't judge the validity, for instance: who is the first one invented the telephone? Bell, Green or Edison?)

Wrong: the answer is wrong.

No Answer: the system doesn't give the answer.

We invited a student who had not relation wtih this experiment to judge the correctness of this test result. We test the system by two methods: Frequencies and Frequencies&& Keyword. The result was presented in Table1 and the histogram was presented in Figure 4 and Figure 5. Experiments showed that this method could achieve better results in practice.

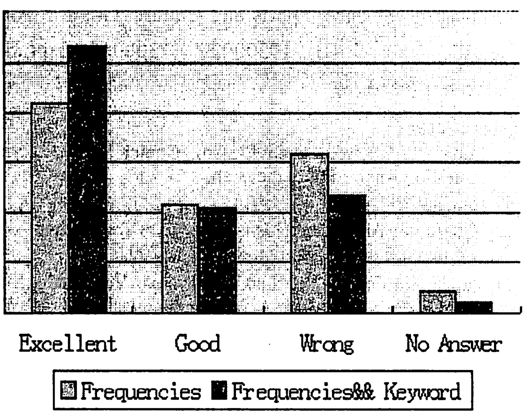


Figure 4: The Histogram of Accuracy

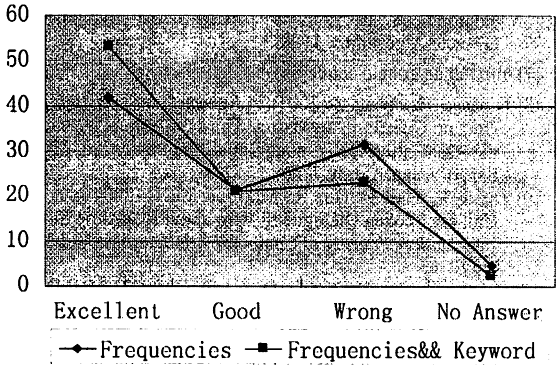


Figure 5: The Histogram of Accuracy

Table 1.

| graded | Frequencies | Frequencies&<br>Keyword |
|---|---|---|
| Excellent | 101 | 128 |
| Good | 52 | 51 |
| Wrong | 76 | 56 |
| No Answer | 11 | 5 |
| Accuracy<br>(Excellent+ Good) | 63.75% | 74.58% |

## 9. Conclusion

This paper outlines a search computing method that uses a Web search engine(Google WEB API ) and a robust parser for the QA System in restricted domains. Although the result of Accuracy (Excellent+ Good) is only 74.58% but it is inspirer. The result showed that this method has potential for the QA System in restricted domains.

## Acknowledgment

This research has been partially supported by the Ministry of Education, Science, Sports and Culture, Grant-in-Aid for Scientific Research (B), 19300029, 17300065, Exploratory Research, 17656128.

## References

[1] Cody C. T. Kwok, Oren Etzioni, Daniel S. Weld, Scaling Question Answering to the Web.

[2] Dumais, S., Banko, M., Brill, E., Lin, J., and Ng, A. Web Question Answering: Is More Always Better? ACM Conference on Information Retrieval, 2002.

[3] Shuichi Kawashima, Toshiaki Katayama, Yoko Sato, Minoru Kanehisa, "KEGG API: A Web Service Using SOAP/WSDL to Access the KEGG System", Genome Informatics 14: 673-674 (2003).

[4] Calishain T. and Domfest R., Google Hacks, O'Reilly, 2003.

[5] Monika R. Henzinger, Allan Heydon, Michael Mitzenmacher, and Marc Najork. Measuring index quality using random walks on the Web. 31(11{16):1291{1303, May 1999.

[6] J. Kleinberg. Authoritative sources in a hyperlinked environment. In Proc. 9th ACM-SIAM Symposium on Discrete Algorithms, 1998.

[7] http://code.google.com/apis/

[8] James, E. & Zheng, Z.(2003). A Speech Interface for Open-Domain Question-Answering. The 41st Annual Meeting of the Association for Computational Linguistics (ACL2003), Sapporo, Japan, July 7-12.

[9] Zhang, L., Gao, F., Guo,R., Mao J. & Lu, R.(2004a). A Chinese Spoken Dialogue System about Real-time Stock Information, *Journal of Computer Applications*, Vol.24, No.7, pp.61-63.

[10] Zhang, H., Yu, H., Xiong, D. & Liu, Q.(2003). HHMM-based Chinese Lexical Analyzer ICTCLAS, Proceedings of 2nd SigHan Workshop, pp.184-187.

[11] Haiqing Hu, Fuji Ren, Shingo Kuroiwa and Shuwu Zhang, "A Question Answering System on Special Domain and the Implementation of Speech Interface", Lecture Notes in Computer Science, Volume 3878, pp.458-469, Springer-Verlag GmbH, Jan 2006