# 國立交通大學

## 資訊科學系

## 碩士論文

# 問答系統之查詢擴充
# －以網路資源為本的非監督式學習策略

Web-Based Unsupervised Learning to Query Formulation
for Question Answering

研究生　　：王怡嘉

指導教授　　：梁　婷　副教授

共同指導教授：張俊盛　教授

中華民國　九十三　年十二月

# 問答系統之查詢擴充－以網路資源為本的非監督式學習策略

# Web-Based Unsupervised Learning to Query Formulation for Question Answering

研究生：王怡嘉　　Student：Yi-Chia Wang

指導教授：梁　婷　　Advisor：Tyne Liang

共同指導教授：張俊盛　　Co-advisor：Jason S. Chang

國立交通大學
資訊科學系
碩士論文

A Thesis
Submitted to Institute of Computer and Information Science
College of Electrical Engineering and Computer Science
National Chiao Tung University
in partial Fulfillment of the Requirements
for the Degree of
Master
in

Computer and Information Science

December 2004

Hsinchu, Taiwan, Republic of China

中華民國九十三年十二月

# 問答系統之查詢擴充－以網路資源為本的非監督式學習策略

研究生：王怡嘉

指導教授：梁　婷
共同指導教授：張俊盛

國立交通大學資訊科學系碩士班

## 摘要

本論文提出一個專為問答系統所設計的學習機制，其以網路資源為本並著重於擴充查詢詞的探討。在分析完使用者所輸入的自然語言問句並辨識出其對應的問句樣式（question pattern）之後，我們根據學習的結果將問句樣式轉換成一組可增進搜尋引擎查詢效能的查詢詞。

在訓練過程中，我們首先從網路上自動收集所需的訓練資料，依據語言學的知識及統計學上的技術從問句訓練資料中抽取出可清楚表達答案類別的問句樣式，最後再透過問句與答案段落之間的關聯性統計以計算出最適合每個問句樣式的擴充查詢詞組。在執行階段，輸入的問句之問句樣式會被轉換成其對應的擴充查詢詞，用以增加搜尋引擎擷取出答案的機會。

我們將提出的想法實作成程式雛形。獨立於訓練資料的實驗結果的確證明我們的方法表現得比一般關鍵詞查詢法好，並且可以明顯地減少使用者尋找答案時所需瀏覽的文章篇數。總而言之，本文針對問答系統中最關鍵的步驟—查詢詞擴充—提出了一個有效且簡單的解決方法。

關鍵詞：問答系統、問句類型分析、查詢詞擴充。

# Web-Based Unsupervised Learning to Query Formulation for Question Answering

Student：Yi-Chia Wang

Advisors：Dr. Tyne Liang
Advisors：Dr. Jason S. Chang

Department of Computer Information Science
National Chiao Tung University

## ABSTRACT

This thesis investigates ways of learning how to formulate and expand a query to find the answer on the Web for a given natural language question. In our approach, the question pattern extracted from a given question is transformed into a set of query terms to improve the performance of an underlying search engine.

In the training phase, the method involves crawling the Web for passages relevant to many pairs of question and answer, extracting of question patterns for fine-grained answer classification based on linguistic and statistical information, and aligning question patterns and keywords with n-grams in the answer passages. At runtime, any given question is converted into a question pattern which is then transformed to their top-ranking alignment counterparts as a way of formulating an expanded query so as to increase the possibilities of retrieve passages containing the answers.

We also describe *Atlas* (Automatic Transform Learning by Aligning Sentences of question and answer), a prototype implementation of the proposed method. Independent evaluation on a set of questions shows that Atlas performs better than a naive keyword-based approach. This method also obviously reduces the human effort of seeking answers, since our system has higher recall rates when a handful of summaries are examined. Our straightforward method improves the most critical stage in question answering systems and also sheds new light on the long-standing problems of query expansion and relevance feedback.

Keyword: question answering, question type extraction, query expansion.

# Acknowledgement

這篇論文能夠順利完成，首先得感謝我的兩位指導教授—梁婷老師與張俊盛老師。在兩位老師的引導下，使學生對自然語言處理及資訊擷取的相關研究產生濃厚的興趣，此外，論文研究方面，老師們也很有耐心的指導並且常常提供珍貴的建議，怡嘉真的很幸運，能同時接受兩位老師的教誨。

接著，特別感謝實驗室的學長、同學、及學弟妹們對於我的關懷與協助。尤其是清華大學博士班吳鑑城學長，當研究過程遇到困難時總是及時給予我幫助；簡嘉言學長提供其開發的文章標記工具，使得怡嘉節省了許多資料前處理的時間；在論文寫作方面，張裕嘉同學花了許多時間幫忙修改；媛茜及捷扉兩位學妹則幫忙實驗的評估部分。

最後，我要感謝我親愛的家人及所有的好友，特別是我的摯愛—原，在我遇到挫折時不斷的鼓勵我，謝謝你們對怡嘉的愛護。

謹將這篇碩士論文獻給所有在我身旁關懷我的人。十分感謝大家。

# Table of Contents

# List of Tables

# List of Figures

# Chapter 1 Introduction

## 1.1 Background

There are many questions to which people want to know the answers in their work, study or even daily life. Berliner et al. (1992) indicated in her book *“The Book of Answers*” that there have been approximate 300,000 questions being answered per year by the stuff of Reference Hotline of New York Public Library. These questions are about every subject from art to zoology, such as *“What is the capital of America?*”, “*Who invented the toothbrush?*”, and “*Which sport do the Chicago Bears play?*” Finding answers to these diverse questions requires a lot of work. For each of the questions, NYPL librarians turns to as many as 1,800 books on the Reference Room’s shelves, including “*Who’s Who in American,*” “*AAA Travel Guides,*” “*Encyclopedia Britannica,*” etc. In order to reduce human efforts, one might think it is desirable to develop a computer system which can answer questions automatically in nearly every category of knowledge.

An automated *question answering* (QA) system receives user’s natural language question as input and provides the exact answer to the question by *understanding* the input question and consulting its knowledge database of a text collection. In fact, this research issue once has been a hot topic in Artificial Intelligence in the 1960’s (Ittycheriah et al., 2000). However, the limitations of computational speed, storage capacity, database, and text processing techniques have made the idea of Question Answering difficult to realize at the time.

Without an effective solution to automatic question answering, people have sought to use *information retrieval* (IR) systems (or Web search engines) until now. To do this, the user usually first transform her question into a list of keywords, then submit those as a query to an

IR system, and finally examine many returned documents or summaries to find the exact answer. The techniques developed for information retrieval systems have been quite successful in retrieving documents relevant to users' keyword queries in general; however, it is noted that the documents containing these keywords may not contain the answer to the question.

In recent years, the combination of increased CPU speed, enlarged storage space, Web resource, very large document collections, and improvements in both information retrieval and natural language processing techniques has reignited the interest in question answering services and brought the study to a completely new era. For instance, Google Answers[1] provides the environment for users to interact with human experts on the Web, while MIT's Computer Science and Artificial Intelligence Laboratory explores Internet resources to develop a Web-base QA system, START[2] (Katz et al., 2003). In order to make QA services more accessible, many experimental QA systems are available through the Internet.

Question Answering research has become even more active after it was introduced to *Text REtrieval Conference* (TREC) since 1999. In TREC QA track, the participating systems have to find the answers to a set of questions from a large collection of documents provided by the sponsor of TREC. The system outputs are returned and judged by human experts.

## 1.2 Components of a Question Answering System

Most of the state-of-the-art QA systems have at least three major components (Hovy et al., 2000; Ittycheriah et al., 2000; Kwok et al., 2001) as shown in Figure 1: question analysis,

[1] Google Answers (http://answers.google.com)
[2] START (http://www.ai.mit.edu/projects/infolab/)

information retrieval, and answer extraction.

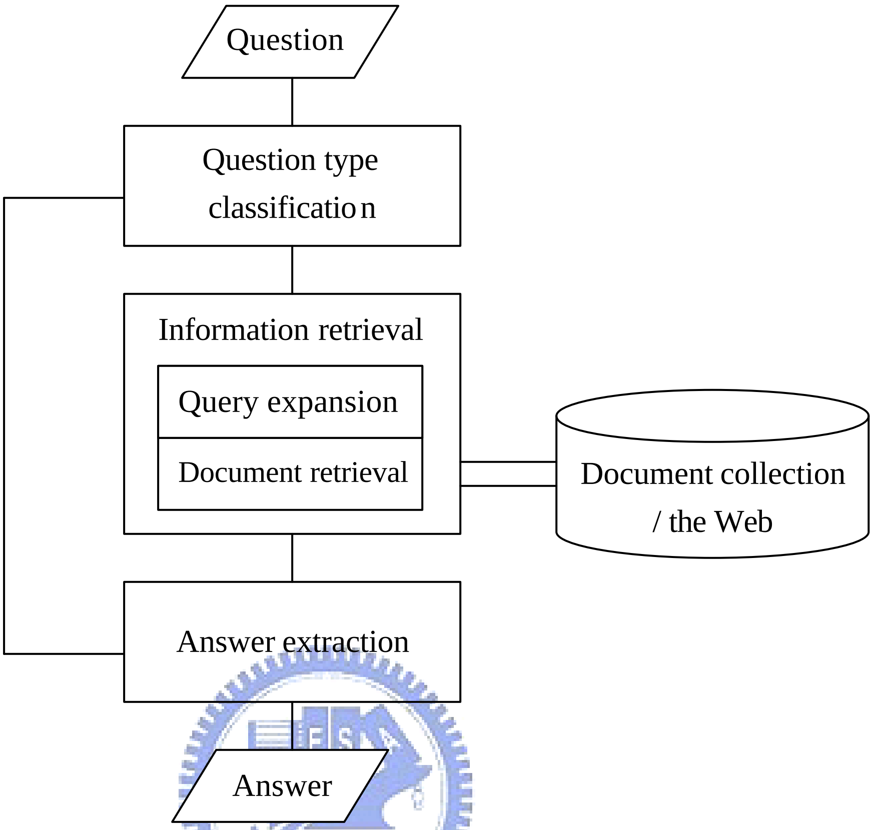


Figure 1: A general QA system.

The purpose of question analysis is to identify the intention and answer type of a question. The identified type will be utilized in the final step to extract the answer. Most questions have to be recognized by various types of name entities, including *person, location, organization*, and *time*. For instance, the question *"Who is the first American President?"* suggests that the expected answer should be a *"person"*, while *"What is the capital of China?"* asks for a *"location"* answer.

The second component, Information Retrieval, deals with retrieving document relevant to the given question. The process of information retrieval usually involves two subcomponents. The first subcomponent is query formulation/expansion that analyzes the given natural language question and paraphrases it into the corresponding query of an IR system. The second one deals with retrieving documents (or passages) matching the query. The knowledge database of this subcomponent is either an internal database, text collection, or the Web pages. An information retrieval system for the Web pages is often called a search engine. Since the Web is much richer in resources, many QA systems have been implemented by tapping into the Web resource using an underlying search engine. In this thesis, we will concentrate our discussion on QA using a Web search engine.

Finally, QA systems have an answer extraction (or answer pinpointing) module to precisely locate the answers from relevant documents or passages primarily based on the question type. It is typically divided into two stages. First, several sentences that probably contain the answer from relevant documents were selected by measuring the similarity between the question and each sentence in the documents (Echihabi & Marcu, 2003). Second, tentative answers are extracted from these sentences by considering the information of the intention of the question.

## 1.3 Motivation

We are motivated by the need to effectively formulate and expand query terms for a given natural language question. A naive solution is then submitting the keywords in the question as a query to a search engine. However, keywords in a question are not usually very effective in retrieving relevant documents since documents that contain the answer probably do not have all these keywords.

Take the question *"Who invented glasses with two foci?"* as an example. Typically, we will send the keyword query "*invented glasses two foci*" to a search engine to retrieve documents or passages. Examining the returned summaries by Google, one would notice that the most returned summaries are irrelevant information about astronomy or physics rather than the inventor "*Benjamin Franklin*" of bifocal glasses. Intuitively, if we can figure out that the answer passage for these ‘*who invented*" questions tend to have a certain word or phrase ("*inventor of*" for example) and transform the question pattern to these effective query terms, we have a better chance to retrieve documents with the answer.

To achieve this goal, recent work has started to exploit the syntactic or even semantic knowledge (Hovy et al., 2000; Kwok et al., 2001; Lin, 2002) to carry out query expansion for QA. Methods have been proposed to utilize WordNet (such as synonyms and hypernyms) to expand keywords in the question in the hope of increasing the possibility of retrieving documents that contain the answer. However, the outcomes show limited improvement because WordNet is a general-purpose ontology and is not particularly suitable for information retrieval.

In this thesis, we present the system *Atlas* (Automatic Transform Learning by Aligning Sentences of question and answer), which automatically learns the transformation from the question to an effective query by using the Web as corpus. First, we crawl the Web to gather relevant answer passages automatically. For each question, we then extract the question pattern that represents the intention according to linguistic and statistical information. Finally, we apply word alignment techniques to questions and answer passages to identify alignment counterpart of the question pattern and keywords in the answer passages. At run-time, any given natural language question can be analyzed and transformed into a query based on the alignment counterparts of its question patterns.

For instance, consider the natural language question "*Who invented the light bulb?*"

Using the keywords in the question directly, we end up with the keyword query, *“invented light bulb,”* for a search engine such as Google. We observed that such a query has room for improvement in terms of bringing in more instances of the relevant passages containing the answer *“Edison.”* Our experiment indicates that the proposed method will determine the best transformations for the question pattern “*who invented*”, including “*inventor of*”, “*was invented*”, and “*invented by*”. Intuitively, these transformations will convert the question into an expanded query for Google, “("*was invented*" || "*invented by*") ("*light bulb*")” which is more effective in fining the answer in one of the top-ranking summaries returned by Google, such as *“The light bulb was invented by an illuminated scientist called Thomas Edison in 1879!”*

The remainder of this thesis is organized as follows. In Chapter 2, we survey the related work. In Chapter 3, we describe our method for unsupervised learning of transformations for question and answer pairs which are automatically acquired from the Web and how we use the aligned results for effective query expansion in a QA system. The experiment and evaluation results are given in Chapter 4. We conclude with discussion and future work in Chapter 5.

# Chapter 2 Related Work

Extensive work on question answering has been reported in the literatures on how to build a question answering system. Moldovan et al. (2002) presented an in-depth performance analysis of a state-of-the-art QA system that was ranked high in three TREC QA track evaluations (cf. Voorhees,1999). This QA system proceeds in three steps: question processing, document retrieval, and answer processing. This serial architecture is representative of many QA systems in the literature (Abney et al. 2000; Prager et al. 2000; Ittycheriah et al. 2000; Hovy et al. 2000). In this thesis, we focus on only question processing. We also consider how to exploit resources on the Web to train and develop the proposed QA system, which includes using large set of quizzes and Web pages that contain answers to those quizzes.

Question processing, including question classification and query formulation, is crucial for retrieving documents or passages that contain answers to the question. Moldovan et al. (2002) pointed out that over 70% of the errors can be attributed to ineffective question type classification, keyword selection, and query expansion. While question type classification is unique to QA, query formulation is a common issue between QA and IR. Query formulation involves keyword selection, phrase formation, and keyword expansion. In general, a query is expanded online by adding terms which are in the top-ranked documents. Since these top-ranked documents may not be truly relevant, this general approach is sometimes called pseudo relevancy feedback (Rocchio 1971; Mitra et al., 1998). In contrast, we will show how to learn, *off line with truly relevant passages*, effective query expansion for a set of questions of a specific type.

Once a query is formed, a typical QA system will submit it to a search engine to obtain data from a local collection of texts (Voorhees, 2001), pre-compiled structured data and un-structured Web pages (Brill et al., 2001; Radev et al., 2001), or a combination of multiple sources (Hovy et al., 2000; Lin, 2002). Most QA systems use a standard search engine to retrieve documents most similar to an input query based on TF-IDF term weighting scheme (Salton and Buckley, 1988). Lin (2002) described how to access semi-structured data that has been organized as databases on the Web for question answering. Web-based databases provide a variety of information ranging from geography to economy; thus, it is a good strategy to use online databases to supplement a QA system with text based data collection.

With retrieved passages, QA systems will extract answers using strategies like surface patterns, named entities recognition, external knowledge source, and data redundancy. Soubbotin and Soubbotin (2001) described an approach that uses character-level surface patterns for locating answers. Ravichandran and Hovy (2002) showed how to extract patterns in an unsupervised manner from the Web. Berger et al. (2000) used a machine translation model trained on an FAQ corpus to extract n-grams appearing near answers so as to find answers for a specific question type. For instance, the question "*how tall is Mount Everest*" and an answer passage "*He started with the highest , 29,028 foot Mt. Everest, in 1984,*" an answer extraction pattern like "<answer> <queryTerm>" for "*how tall*" questinos will be derived. The authors used the question word and the following word to specialize the answer extraction patterns. In contrast, we use a more complicated question pattern to learn query expansion rules.

Girju (2001) proposed to build ontology during search and answer extraction process when handling definition and cause/effect questions. If the question is about the cause of some event or problem, then the ontological relationship of cause will help locate the answer. Girju showed that this improves answer extraction for both types of question. Recently, Mann

(2002a) extended this approach to proper names. He pointed out that questions in TREC 8, 9 and trivia quizzes tend to indicate a typed proper name preference and therefore fine-grained proper noun ontologies will be very useful for answer extraction and validation. He proposed to construct these ontologies by simply using the pattern of common noun followed with a proper noun in a very large corpus. There is also a trend in the use of existing knowledge resources in question answering. Webclopedia (Hovy et al., 2001) uses WordNet (Miller, 1995) to assist in answering definition questions, while IBM’s statistical question answering system uses an external encyclopedia (Ittycheriah et al., 2000) for query expansion.

Increasingly, researchers use the Web as corpus for answering question. On the Web, it tends to be more than one document containing the answer. Researchers (Kwok et al., 2001; Brill et al., 2001; Buchholz, 2001; Clarke et al., 2001) have exploited such data redundancy to extract or verify answer candidates. Mann (2002b) also used trivial games and Web data to learn the words occurring near the answers to locate the answers.

Some QA approaches try to convert original input questions into a more effective query with the goal of retrieving documents more likely to contain the answers. Hovy et al. (2000) utilized WordNet hypernyms and synonyms to expand queries to increase recall. Hildebrandt, Katz, and Lin (2004) looked up in a pre-compiled knowledge base and a dictionary to expand a definition question. However, blindly expanding a word using its synonyms or dictionary gloss sometimes causes undesirable effects. Furthermore, it is difficult to determine which of many related words should be used to expand the query. In contrast to this approach, we use real-life questions and relevant answer passages to rank terms to derive the best transformations for query expansion.

Radev et al. (2001) proposed a probabilistic algorithm called QASM that learns the best query expansion for a natural language question. The query expansion takes the form of a series of operators, including INSERT, DELETE, REPLACE, etc., to paraphrase a factual question into the best search engine query by applying Expectation Maximization algorithm. Although QASM is theoretically sound, it seems to derive more or less the obvious things to do in query formulation. For instance, the top two operators learned to paraphrase a "*location*" question are DELETE-AUXILIARY and DELETE-PREPOSITION. In contrast, we adopt these common practice directly (by removing stop words) and focus on paraphrasing the NP, VP, or AP construction that contains the question word.

Additionally, Hermjakob et al. (2002) described an experiment to observe and learn from human subjects who were given a question and asked to write queries which are most effective in retrieving the answer to the question. First, several randomly selected questions are given to users to "manually" generate effective queries that can bias Web search engines to return answers. The questions, queries, and search results are then examined to derive seven query reformulation techniques that can be used to produce queries similar to the ones issued by human subjects. This approach is time-consuming and is limited by the number of questions which can be handled at one time. It is doubtful whether the rules will be very general if only a few questions are used in this learning process that involves both human subjects and the developer. Instead of handcrafting question-to-query transformation rules that rely on human intervention, we introduce a statistical model for automatic learning the query formulation rules based on a large number of questions and answer passages.

In recent study, Shen et al. (2003) proposed to submit the keywords in "*why*" questions as queries to Google and to retrieve documents that contain the answers to these question. The authors analyzed the answer corpus with the help of a part-of-speech (POS) tagger and came to the conclusion that expanding the query with three words, "*reason*," "*why*," and "*because*" is effective. In contrast, we propose an approach capable of automatically learning how to expand queries for all kinds of question.

On the other hand, Echihabi and Marcu (2003) proposed a noisy channel approach to question answering. Their method also involves collecting answer passages from the Web and aligning words or concepts across from a question to its relevant answer passages. However, in addition to full parsing of the sentences, their method also required complicated decision of making a "cut" in the parse trees of the question and answer sentences to determine how to align word, syntactic, or semantic categories. Our method is also based on alignment but it is much simpler to implement since it simply performs alignment at the surface level of words and n-grams without full parsing.

Agichtein et al. (2003) presented the *Tritus* system that automatically learns transformations of wh-phrases (e.g. expanding "*what is*" to "*refers to*") by using Web-based FAQ data. However, the wh-phrases are restricted to sequences of function word beginning with an interrogative, (i.e. who, what, when, where, why, and how). These wh-phrases tend to coarsely classify questions into a few types. In contrast, our method automatically identifies a content word (i.e. adjective, noun, or verb) that reflects a finer-grained classification of question. We also illustrate how to learn transformations for such a fine-grained question type using data that are the same as those being handled at run-time. *Tritus* uses heuristic rules and thresholds of term frequencies to learn transformations, while we rely on a mathematical model for statistical machine translation.

In contrast to previous work in question answering and query formulation, we address the problem in transforming a natural language question to a search engine query, by optimizing transformatinos specifically for a fine-grained question type with the goal of increasing the odds of retrieving documents that contain the answers. Our method is able to learn automatically such effective transformations by exploiting the Web as corpus and distributional regularity of the answer passages based on statistical word alignment techniques.

# Chapter 3 Learning Question to Query Transformation

Recall that submitting natural language questions to search engines is not the best way to retrieve a passage containing the answer. A promising approach is to expand the original question by adding terms that co-occur with the answers to a specific type of question.

We will focus on this aspect of query expansion for QA. More specifically, we present a method for QA which automatically learns best transformations from a given natural language question to an effective query by using the Web as corpus. In our model, we derive such an effective query by applying a set of transformations to a given question. To that end, we first automatically obtain a collection of answer passages (*AP*s) as our training corpus from the Web by using a set of (*Q*, *A*) pairs. After that, we identify the question pattern for each training *Q* by using statistical and linguistic information. Here, a question pattern $Q_p$ is defined as a question word plus its keywords that is the most related to the question word in *Q*. $Q_p$ is meant to represent the intention of the question. Finally, we decide the transformations *T*s for each $Q_p$ by choosing those phrases in the *AP*s that are statistically associated with $Q_p$ and adjacent to *A*s.

In the rest of this chapter, we will describe the proposed method in detail. Section 3.1 shows how to automatically crawl the Web for training materials, while Section 3.2 describes the strategies for detecting and extracting question patterns from questions for subsequent processing. In Section 3.3, we show how effective transforms are obtained by aligning words across from questions to answer passages. Finally, we describe the run-time procedure for converting a user's question into a query for a search engine in Section 3.4.

## 3.1 Web Crawl for Relevant Answer Passages

The proposed method is based on a data-driven learning approach; thus a lot of training data is needed. However, it is easier to gather a large quantity of training data from the Web. We describe a method that can mine a large amount of question/answer passage pairs from the Web by using a set of question/answer pairs as seeds.

More formally, we attempt to retrieve a set of (*Q*, *AP*) pairs on the Web for training purpose, where *Q* stands for a natural language question, and *AP* is a passage containing at least one keyword in *Q* and *A* (the answer to *Q*). The seeds of such data gathering process are a set of (*Q*, *A*) pairs which can be acquired from many sources, for instance, trivia game Websites, QA benchmarks such as TREC-QA track, and Frequently Asked Question (FAQ) files. The output will be a large collection of (*Q*, *AP*) pairs. We describe the procedure in details as follows:

1. For each (*Q*, *A*) pair, the keywords $k_1$, $k_2$,…, $k_n$ is extracted from *Q* by removing stopwords.
2. Submit ($k_1$, $k_2$,…, $k_n$, *A*) as a query to a search engine *SE*.
3. Download the top *n* summaries returned by *SE*.
4. Separate sentences in the summaries. Make sure that HTML tags, URL, special character references (e.g., "&lt;") are removed.
5. Retain only those sentences which contain *A* and some $k_i$.

For instance, consider the case of gathering answer passages from the Web for the (*Q*, *A*) pair where *Q* = *"What is the capital of Pakistan?"* and *A* = *"Islamabad."* The query submitted to a search engine and potential answer passages returned by a search engine are shown in Table 1:

Table 1: An example of converting a question (*Q*) with its answer (*A*) to a SE query and retrieving answer passages (*AP*).

| (*Q*, *A*) | *AP* |
|---|---|
| What is the capital of Pakistan?<br>Answer:( Islamabad) | Bungalow For Rent in Islamabad, Capital Pakistan. Beautiful Big House For … |
| ($k_1$, $k_2$,…, $k_n$, $A$) | Islamabad is the capital of Pakistan. Current time, … |
| capital, Pakistan, Islamabad | …the airport which serves Pakistan's capital Islamabad, … |

Note that it is difficult to guarantee all the retrieved passages to be relevant. In other words, there could be some amount of irrelevant, noisy passages. However, a statistical method can be applied to filter out those irrelevant passages to a certain degree.

## 3.2 Question Analysis

This section describes the presented identification of the so-called “question pattern” which is critical in transforming a given question into a query (Section 3.3).

### 3.2.1 A Question Pattern

Intuitively, a question can be in a certain type according to the semantic nature of its answer. More specifically, a question can be classified as a type of PERSON (who-question), PLACE (where-question), TIME (when-question), OBJECT (what-question), REASON (why-question), etc. Although this classification may be useful for pinpointing the answer, it is too coarse to be useful for query expansion For instance, consider the question *“Who invented the telephone*?” Knowing the answer is a PERSON does not suggest how to expand the query effectively. Instead, characterizing “*who invented*” under a fine-grained question classification may lead us to learn that more effective query terms such as “*inventor of*” and “*invented by*.” These effective terms can be derived easily, since they appear quite frequently in the answer passages of “*who invented*” questions (e.g. “*Who invented eye glasses?*”, “*Who invented light bulb?*”, and “*Who invented toothbrush?*”). To develop a fine-grained classification of questions, we need words in the question in addition to the question word. We pick the additional words according to phrase structure of the question; these words should be content words rather than function words and they may not immediately follow the question word (e.g. “*who*”). For instance, in the question “*What is the normal color of a black box used in airplanes?*”, the desired classification should be “*what color*” rather than “*what is*.”

To address the issue of fine-grained classification, we develop a new approach which is somewhat different from those proposed in the QA literature (e.g. Agichtein et al., 2003). Instead of computing the frequency of all n-grams in questions and choosing several high-frequency “question phrases,” we run the questions through a part-of-speech (POS) tagger and a basic phrase chunker to identify the head words in the chunk containing the question word or immediately following the question word to form a question pattern.

Formally, we define a “question pattern” for any question as following form:

**[question-word] + (head-word)**$^{+}$

where “question-word” is one of the interrogatives (Who/What/Where/When/How) and “head-word” represents the headword in the subsequent chunks that tend to reflect the intended answer more precisely. Typically, a headword may contain one or two words. For instance, “*who had hit*” is a reasonable question pattern for “*Who had a number one hit in 1984 with ‘Hello’ ?* ”, while “who had” seems to be too coarse.

In order to determine the appropriate question pattern for each question, we examined and analyzed a set of questions which are POS-tagged and phrase-chunked. With the help of a number of simple heuristic rules based on POS and chunk information fine-grained classification of questions are produced.

### 3.2.2 Tagging and Chunking

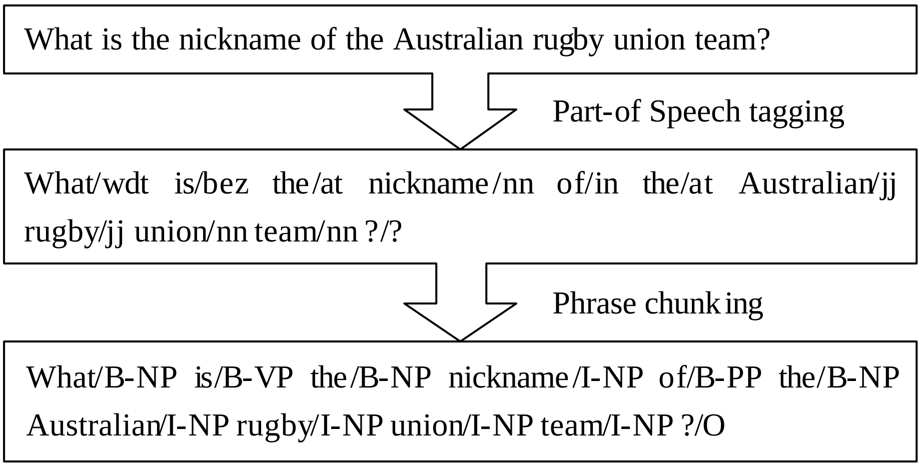
Part-of-speech tag and chunk provide sufficient linguistic information for extracting question patterns. Figure 2 shows an example of performing shallow syntactic analysis for the question "*What is the nickname of the Australian rugby union team?*" using a tagger trained on the Brown Corpus and a chunker trained on CoNLL2002 data, where B-NP denotes the beginning of an noun phrase (NP), I-NP denotes the rest of the NP. Verb phrases (VP), adjective phrase (AP), and other chunks are tagged similarly.

**(1)** *What is the nickname of the Australian rugby union team?*

POS: what/wdt is/bez the/at nickname/nn of/in the/at Australian/jj ruby/nn union/nn team/nn ?/?

Chunk: which/NP-B is/VP-B the/NP-B nickname/NP-I of/PP-B the/NP-B Australian/NP-I ruby/ NP-I union/ NP-I team/ NP-I ?/O

Phrase: which/NP is/VP the nickname/NP of/PP the Australian ruby union team/NP ?/O



What is the nickname of the Australian rugby union team?

What/wdt is/bez the/at nickname/nn of/in the/at Australian/jj rugby/jj union/nn team/nn ?/?

What/B-NP is/B-VP the/B-NP nickname/I-NP of/B-PP the/B-NP Australian/I-NP rugby/I-NP union/I-NP team/I-NP ?/O

Figure 2: Shallow parsing of the question (1).

For the purpose of extracting question patterns, we further group the words in the same chunk together (see Table 2). With POS and chunk information, we can extract the question pattern “*what nickname*” for the sample question according to some heuristic rules described next.

Table 2: An example of a tagged and chunked question.

| Question | POS Tag | Phrase Tag |
|---|---|---|
| **what** | **wdt** | **NP** |
| is | bez | VP |
| the **nickname** | at **nn** | **NP** |
| of | In | PP |
| the Australian rugby union team | at jj jj nn nn | NP |

### 3.2.3 Linguistic Analysis for Question Pattern Extraction

After analyzing recurring patterns and regularity in quizzes on the Web, we designed a simple procedure to recognize question patterns. We present this procedure as a small set of prioritized rules (see Figure 3).

(Rule 1) Question word in a chunk of length more than one (e.g. “*which female singer*”)

(Rule 2) Question word followed by a light verb and NP chunk (e.g. “*who made flight*”)

(Rule 3) Question word followed immediately by a verb (e.g. “*who painted*”)

(Rule 4) Question word followed immediately by a passive VP or an NP (e.g. “*what is called*”)

(Rule 5) Question word followed by the copulate “to be” and an NP (e.g. “*what is the river*”)

Figure 3: Rules used to identify the question pattern in a given question

First, we identify the question word which is one of the wh-words (“*who,*” ‘*what,*” “*when,*” “*where,*” “*how,*” or ‘*why*”) tagged as determiner or adverbial question word (i.e., “*wdt,*” “*wql,*” and “*wrb*”). According to the result of POS tagging and phrase chunking, we further decide the main verb and the voice of the question. Then, we proceed to apply the following **expanded rules** to extract words to form question patterns:

**Rule 1.a If the question word is tagged with “wdt” and it is in a NP chunk of length greater than one, its question pattern will contain the question word and the headword of the chunk.**

**Rule 1.b If the question word is tagged with “wql” and it is in a NP chunk of length greater than one, its corresponding question pattern will contain the question word and the following adjective (“jj” and “ap”).**

For instance, consider the following Examples (2) to (4):

**(2)** ***Which*** *female* ***singer*** *performed the first song on Top of the Pops?*

POS: which/wdt femle/jj singer/nn performed/vbd the/at first/cd song/nn on/in top/nn of/in the/at pops/nns ?/?

Chunk: which femle singer/NP performed/VP the first song/NP on/PP top/NP of/PP the pops/PP ?/O

**(3)** ***How many*** *American states begin with the letter “M”?*

POS: how/wql many/jj American/jj states/nns begin/vb with/in the/at letter/nn “/“ M/nn ”/” ?/?

Chunk: how many American states/NP begin/VP with/PP the letter/NP “/O M/NP ”/O ?/O

**(4)** *In* ***what year*** *was Hong Kong returned to China?*

POS: in/in what/wdt year/nn was/bed Hong/np Kong/np returned/vbd to/to China/np ?/?

Chunk: in/pp what year/NP was/VP Hong Kong/NP returned/VP to/PP China/NP ?/O

After we apply Rule 1.a to Example (2), the question word ‘*who*” and the headword “*singer*” in the same NP chunk will be chosen to form the question pattern. Consider another question in Example (3). Rule 1.b applies and the question pattern is the question word plus an adjective, “*how many*.” The question in Example (4) is handled similarly.

**Rule 2 If the question word is a chunk by itself and the main verb is a light verb (i.e., have, do, know, think, get, go, say, see, come, make, take, look, give, find, use), then the question pattern is composed of the question word, the light verb, and the head of the first NP or PP chunk following the light verb.**

By applying Rule 2 to Example (5), it question pattern will be “*who made flight*.”

**(5)** ***Who*** *in 1961* ***made*** *the first space* ***flight****?*

POS: who/wps in/in 1961/cd made/vbd the/at first/od space/nn flight/nn ?/?

Chunk: who/NP in/PP 1961/NP made/VP the first space flight/NP ?/O

**Rule 3 If the question word is a chunk by itself followed by a VP or NP chunk without a light verb, the question pattern is the question word and the head word of the VP.**

By applying Rule 3 to Example (6), it question pattern will be “*who painted*.”

**(6)** ***Who painted “****The Laughing Cavalier*”?

POS: who/wps painted/vbd “/“ the/at Laughing/vbg Cavalier/nn ”/” ?/?

Chunk: who/NP painted/VP “/O the laughing cavalier./NP ”/O ?/O

**Rule 4 If the question word is in a chunk by itself and the question is in passive voice, the question pattern will contain the question word, “to be,” and the headword of the passive VP.**

Applying Rule 4 to the following Example (7) and (8), we will get question patterns *"what is called"* and *"what is known"* respectively.

**(7)** ***What is*** a group of geese ***called***?

POS: What/wdt is/vbz a group/np of/in geese/nns called/vbn ?/?

Chunk: what/NP is/VP a group/NP of/PP geese/NP called/VP ?/O

**(8)** *In Bible,* ***what is known as*** *the Decalogue?*

POS: in/in Bible/np ,/, what/wdt is/vbz known/vbn the/at Decalogue/np ?/?

Chunk: in/PP Bible/NP ,/O what/NP is known/VP the Decalogue/NP ?/O

**Rule 5 If the question word is in a chunk by itself follow by a "to be" chunk and an NP chunk, the question pattern is the question word and the headword of the first NP.**

Appling Rule 5 to Example (9), we will get a question pattern "*what river*"

**(9)** ***What*** *is* the second longest **river** in the world?

POS: What/wdt is/vbz the/at second/od longest/jjt river/nn in/in the/at world/nn ?/?

Chunk: what/NP is/VP the second longest river/NP in/PP the world/NP ?/O

Finally, we have the last rule to hand all the other cases:

**Rule 6 If none of the above rules are applicable, the question pattern will contain the question word only.**

It is noticed that the heuristic rules (as 1~6) are intuitive. Moreover, the generated and recurring patterns suggest generality of the patterns and the feasibility of gathering training data to learn the terms that co-occur with the answers. These question patterns also indicate a preference for the answer to belong to a fine-grained type of proper nouns as observed by Mann (2002a) (see Table 3). In the next section, we describe how we exploit these patterns to learn how to carry out effective query expansion.

Table 3: Question patterns suggest preference to fine-grained type of proper noun.

| Questions | Question Pattern | type of anwers |
|---|---|---|
| Which rock ‘n’ roll musician | which-musician | musi**cian** |
| Which singer … | which-singer | singer (musician) |
| Who sang … | who-sang | singer (musician) |
| Who’ s the lead singer | which-singer | s**inger (musician**) |
| What female Disco singer | what-singer | singer (musician) |
| What helicopter pilot | what-pilot | pilot |
| Who **made flight** | who-made-flight | pilot |
| Which astronaut | what-astronaut | astronaut (pilot) |
| What Russian astronaut | what-astronaut | astro**naut (pilot)** |
| Who is the author | who-author | author |
| Who wrote | who-wrote | author |
| What car company | what-company | company |
| What Hollywood studio | what-studio | studio (company) |

## 3.3 The Method of Learning the Best Transformations for Question Patterns

This section describes the procedure for learning transformations *T*s which convert the question pattern $Q_p$ into bigrams appearing in relevant *AP*s. The reason why we use bigrams in *AP*s instead of unigrams is that bigrams tend to have more unique meaning than single words and are more effective in retrieving relevant passages. In fact, an earlier experiment on unigrams has been conducted, and as we have predicted the results were not good. The process consists of three steps which are shown in Figure 4.

(1) Apply a word alignment algorithm to questions and relevant answer passages (Section 3.3.1) and tally the alignment counts of question patterns.

(2) Tally the high-frequency bigrams preceding or following the answers in the answer passages (Section 3.3.2)

(3) Combine the results in (1) and (2) to derive rank and transformations (Section 3.3.3)

Figure 4: Procedure for learning transforms.

### 3.3.1 Word Alignment Technique for Learning Question Pattern Transformations

First, we use word alignment techniques originally developed for statistical machine translation to find out relationships between question patterns in *Q* and bigrams in *AP*. We use Competitive Linking Algorithm proposed by Melamed (1997) to align a set of (*Q*, *AP*) pairs, mined by the method described in Section 3.1.

Our method involves a number of preprocessing steps for each ($Q$, $AP$) pair for filtering useless information:

1. Perform part-of-speech tagging on both $Q$ and $AP$.
2. Replace all instances of $A$ with the tag <ANS> in $AP$s. For example, the answer "Islamabad" in $AP$s for the question *"What is the capital of Pakistan?"* is replaced with <ANS>. The purpose of <ANS> is to provide information on the location of the answers.
3. Identify the question pattern, $Q_p$ and keywords which are *not* proper nouns in $Q$ since proper nouns are not the target for query expansion. We denote the question pattern as $q_1$ and remaining keywords as $q_2$, ..., $q_n$.
4. Convert $AP$ into bigrams and eliminate bigrams with low term frequency (tf) or high document frequency (df). For the purpose of extracting effective transformations, those bigrams containing two function words are removed, and the remaining bigrams are denoted as $a_1$, $a_2$, ..., $a_m$.

For example, consider $Q$ = *"How old was Bruce Lee when he died?"* Applying these preprocessing steps, we will have $q_1$ = *"how old"* and $q_2$ = *"died"* where *"Bruce Lee"* is eliminated because it is a proper noun and the stopwords, *"was"*, *"when"* and *"he"* are removed.

After the preprocessing steps, we then proceed to align $q$'s and $a$'s via Competitive Linking Algorithm (CLA) procedures as follows:

**Input** A collection $C$ of ($Q$; $A$) pairs, ($Q$; $A$) = ($q_1$ = $Q_p$ , $q_2$, $q_2$, ..., $q_n$ ; $a_1$, $a_2$, ..., $a_m$)

**Output** Best alignment counterpart $a$'s for all $q$'s in $C$.

1. For each pair of ($Q$; $A$) in $C$ and for all $q_i$ and $a_j$ in each pair of C, calculate LLR($q_i$,

$a_j$), logarithmic likelihood ratio (LLR) between $q_i$ and $a_j$, which reflects their statistical association.

**Log-likelihood ratio : LLR(x, y)**

$$LLR(x;y) = -2\log_2 \frac{p_1^{k_1}(1-p_1)^{n_1-k_1}(1-p_2)^{n_2-k_2}}{p^{k_1}(1-p)^{n_1-k_1}\,p^{k_2}(1-p)^{n_2-k_2}}$$

$k_1$ = number of pairs that contain x and y simultaneously.
$k_2$ = number of pairs that contain x but do not contain y.
$n_1$ = number of pairs that contain y
$n_2$ = number of pairs that does not contain y
$p_1 = k_1 / n_1$  $p_2 = k_2 / n_2$  $p = (k_1+k_2) / (n_1+n_2)$

2. Discard ($q$, $a$) pairs with a LLR value lower than 7.88.

3. For each pair of ($Q$; $A$) in $C$ and for all $q_i$ and $a_j$ therein, carry out Steps 4-7:

4. Sort list of ($q_i$, $a_j$) in each pair of ($Q$ ; $A$) by decreasing LLR value.

5. Go down the list and select a pair if it does not conflict with previous selection.

6. Stop when running out of pairs in the list.

7. Produce the list of aligned pairs for all $Q$s and $AP$s.

8. Tally the counts of aligning ($q$, $a$).

9. Select top $m$ bigrams, $t_1$, $t_2$, ..., $t_k$, for every $q$, where $q$ is a question pattern or keyword.

The LLR statistics is generally very effective in distinguishing related terms from unrelated ones. However, if two terms occur frequently in questions, their alignment counterparts will also occur frequently, leading to erroneous alignment due to indirect association. CLA is designed to tackle the problem caused by indirect association. Therefore, even if we only make use of the alignment counterpart of the question pattern, we still put in the question keywords so as to reduce the errors caused by indirect association. For instance, consider the question *"How old was Bruce Lee when he died?"* Our goal is to learn the best transformations for the question pattern *“how old.”* In other words, we want to find out what terms are associated with *“how old”* in the answer passages. However, if we only consider the alignment counterparts of *“how old”* without considering those of the keyword such as *“died,”* we run the risk of getting *“died in”* or *“is dead”* rather than *“years old”* and *“age of.”*

If we have sufficient data for a specific question pattern such as *“how long,”* it will be highly possible for us to obtain alignment counterparts that are effective terms for query expansion. Examples of question patterns, alignment counterparts, and alignment counts are shown in Table 4. The data reveal that high alignment count indeed indicates strong statistical association and effectiveness for query expansion.

Table 4: Examples of alignment results

| Question term | Expansion | Co-occurrence |
|---|---|---|
| How old | age of | 36 |
| How old | years old | 34 |
| How old | ascend the | 13 |
| How old | the youngest | 13 |
| How old | throne in | 9 |
| How old | know | 6 |
| … | … | … |
| who invent | invented the | 37 |
| who invent | invented by | 26 |
| who invent | was invented | 8 |
| who invent | discovered by | 6 |
| who invent | discovery of | 5 |
| … | … | … |
| die | died in | 60 |
| die | died of | 24 |
| die | the age | 22 |
| die | died on | 19 |
| die | death of | 15 |
| die | died at | 12 |
| die | age of | 9 |
| die | and died | 8 |
| die | version of | 7 |
| die | die eight | 5 |
| die | yearsof | 5 |
| … | … | … |

### 3.3.2 Distance Constraint and Proximity Ranks

In addition to the association strength reflected by alignment counts and co-occurrence, the distance of the bigrams to the answer should also be considered, we observe that terms in the answer passages close to the answers intuitively tend to be useful in retrieving answers. Thus, we calculate the bigrams appearing within a window of 3 words on both sides of the answers to provide additional constraints for query expansion. Table 5 shows examples of bigrams, counts, and ranks in the proximity of the answers to those “*who invent*” questions in the answer passages.

Table 5: Bigrams with counts in the proximity the answers to ‘*who invented*” questions.

| Bigrams | Proximity Counts | Proximity Rank |
|---|---|---|
| invented by | 28 | 1 |
| invented the | 25 | 2 |
| discover penicillin | 14 | 3 |
| at age | 9 | 4 |
| discovered by | 8 | 5 |
| invent vulcanize | 8 | 5 |
| in 1928 | 5 | 6 |
| 1839 by | 3 | 7 |
| … | … | … |

### 3.3.3 Combing Alignment and Proximity Ranks

In this subsection, we describe how to decide the best bigrams as the transformations for a specific question pattern based on a combined rank of alignment count and proximity count. We simply take the average of these two counts to re-rank bigrams. The average rank of a bigram $b$, $Rank_{\text{avg}}(b) = (Rank_{\text{align}}(b) + Rank_{\text{prox}}(b))/2$, where $Rank_{\text{align}}(b)$ is the rank of $b$’s alignment count and $Rank_{\text{prox}}(b)$ is the rank of $b$’ s proximity count. The $n$ top-ranking bigrams for a specific type of question will be chosen to transform the question pattern into query terms. For the question pattern ‘*how old*,” the candidate bigrams with alignment counts, co-occurring counts, and average ranks are shown in Tables 6 through 8.

Table 6: Bigram rank of *“how old”* in alignment results.

| Question phrase | Alignment Counterparts | Alignment Rank |
|---|---|---|
| how old | age of | 1 |
| how old | year old | 2 |
| how old | ascend the | 3 |
| how old | the youngest | 4 |
| how old | throne in | 5 |
| … | … | … |

Table 7: Ranked bigrams near the answers to “*how old*” questions.

| Question phrase | Bigram | Proximity Rank |
|---|---|---|
| how old | age of | 1 |
| how old | years old | 2 |
| how old | throne in | 3 |
| how old | when she | 6 |
| how old | at the | 7 |
| how old | at age | 4 |
| how old | when her | 10 |
| how old | be only | 5 |
| how old | year later | 12 |
| how old | response in | 9 |
| … | … | … |

Table 8: Average rank calculated from for the bigram counterparts of “*how old*”.

| Bigrams | Alignment Rank | Proximity Rank | Avg. Rank | Final Rank |
|---|---|---|---|---|
| age of | 1 | 1 | 1 | 1 |
| years old | 2 | 2 | 2 | 2 |
| ascend the | 3 | - | - | - |
| throne in | 4 | 3 | 3.5 | 3 |
| the youngest | 3 | - | - | - |
| … | … | … | … | … |

## 3.4 Runtime Transformation of Questions

At run time, a given question $Q$ submitted by a user is converted into one or more keywords and a question pattern, which is subsequently expanded in to a sequence of query terms based on the transformations obtained at training.

We follow the common practice of keyword selection in formulating $Q$ into a query:

- Function words are identified and discarded.
- Proper nouns that are capitalized or quoted are treated as a single search term. We will put quote around a proper noun if it is not already quoted.

Additionally, we expand the question patterns based on alignment and proximity considerations:

- The question pattern $Q_p$ is identified according to the rules described in Section 3.2 and expanded to a disjunction (sequence of OR) of $Q_p$'s headword and $n$ top-ranking bigrams (described in section 3.3).
- The query will be a conjunction (sequence of AND) of expanded $Q_p$, proper names, and remaining keywords. Except for the expanded $Q_p$, all other proper names and keywords will be in the original order in the given question for best results.

Consider the case of formulating a query for the question *“How old was Bruce Lee when he died?”* Its question pattern is simply “*how old.*” There is a proper noun “*Bruce Lee*” in the question and a remaining keyword *“died.”* Therefore, the query is “( ‘*old*’ OR ‘*age of*’ OR ‘*years ol’* ) AND ‘*Bruce Lee’* AND ‘*died.*’” See Table 9 for the example of formulating a query for *“How old was Bruce Lee when he died?”*.

Table 9: An example of transformation from question into query.

<table>
<tr><th colspan="3">Question</th></tr>
<tr><td colspan="3">How old was Bruce Lee when he died?</td></tr>
<tr><th>Question pattern</th><th>Proper noun</th><th>Keyword</th></tr>
<tr><td>how old</td><td rowspan="3">“Bruce Lee”</td><td rowspan="3">died</td></tr>
<tr><th>Transformation</th></tr>
<tr><td>age of, years old</td></tr>
<tr><th colspan="3">Expanded query</th></tr>
<tr><td colspan="3">Boolean query: ( “old” OR “age of” OR “years old” ) AND “Bruce Lee” AND “died”</td></tr>
<tr><td colspan="3">Equivalent Google query: (old || “age of” || “years old”) “Bruce Lee” died</td></tr>
</table>

# Chapter 4 Experiments and Evaluation

In this Chapter, we describe an implementation of the proposed method using the Web search engine, Google, as the underlying information retrieval system. We will also evaluate the experimental results to assess the effectiveness of question classification and query expansion.

We start by introducing the experimental setup in Section 4.1. Then, in Section 4.2, we describe the experimental results and evaluation of question patterns extraction. Finally, we describe the experimental results and evaluation of query expansion method in Section 4.3.

## 4.1 Experimental Setup

We gathered the seed data of questions and answers from QuizZone[3]. This trivia game website provides new quizzes and answers every week on a specific topic. The answers will lately be posted on the Web in a week. The quizzes cover a wide range of subjects, including popular culture, geography, and music, etc.

We collected the questions posted before June, 2004 on QuizZone. After removing redundant ones, we obtained 3,851 distinct question answer pairs. We set aside the first 45 questions for testing purpose and used the rest for training. We also use the 200 questions from TREC-8 QA Track to evaluate the performance of question pattern extraction. For each question, the top 100 summaries returned by Google are stored as the answer passages. In all, we automatically retrieved 95,926 answer passages. See Table 10 for details of the training

[3] QuizZone (http://www.quiz-zone.co.uk)

corpus.

Table 10: The training corpus of questions, answers, and answer passages from the Web.

| Training data set | Distinct (*Q*, *A*) | Distinct (*Q*, *AP*) |
|---|---|---|
| Quiz-Zone | 3,806 | 95,926 |

The tagger and chunker we used for are developed by our laboratory. We used them to perform shallow parsing of the questions and answer passages. The tagger was developed using the Brown corpus and WordNet. The chunker is built from the shared CoNLL-2000 data provided by CoNLL-2000. The shared task CoNLL-2000 provides a set of training and test data for chunks available at http://cnts.uia.ac.be/conll2000/. Our chunker performs at about 94% average precision rate.

## 4.2 The Performance of Question Pattern Extraction

The 200 questions from TREC-8 QA Track provide an independent evaluation of how well the proposed method works for question pattern extraction works. We will also give an error analysis.

Two human judges both majoring in Foreign Languages were asked to assess the results of question pattern extraction and give a label to each extracted question pattern. Apattern will be judged as “good” if it clearly expresses the answer preference of the question; otherwise, it is tagged as “bad.” The precision rate of extraction for these 200 questions is shown in Table 11. The second column indicates the precision rate when two judges agree that an extracted question pattern is “good.” In addition, the third column indicates the rate of those question patterns that are found to be “good” by either judge. The results imply that the proposed pattern extraction rules are general, since they are effective even for questions independent of the training and development data. Table 12 shows evaluation results of the first five questions.

Table 11: Evaluation results of question pattern extraction.

| | Two “good” labels | At least one “good” label |
|---|---|---|
| Precision (%) | 86 | 96 |

Table 12: The first five questions with question patterns and judgment.

| Question | Question pattern | Judgment |
|---|---|---|
| Who is the author of the book, "The Iron Lady: A Biography of Margaret Thatcher"? | Who-author | good |
| What was the monetary value of the Nobel Peace Prize in 1989? | What value | good |
| What does the Peugeot company manufacture? | What do manufacture | good |
| How much did Mercury spend on advertising in 1993? | How much | good |
| What is the name of the managing director of Apricot Computer? | What name | bad |

We summarize the reasons behind these patterns considered as “bad”:

- Incorrect part-of-speech tagging and chunking
- Imperative questions such as “*Name the first private citizen to fly in space.*”
- Question patterns that are not specific enough

For instance, the system produces ‘*what name*” for “*What is the name of the chronic neurological autoimmune disease which attacks the protein sheath that surrounds nerve cells* *causing a gradual loss of movement in the body?*”, while the judges suggested that “what disease” would be more appropriate. Indeed, some of the patterns extracted can be modified to meet the goal of being fine-grained and indicative of a preference to a specific type of proper nouns or terminology.

## 4.3 Experimental results and evaluation of query expansion

We describe the experimental results and evaluation of how *Atlas* does in terms of expanding query and finding answers using the search engine Google. We start by describing the experimental setting in both training and run time. Then, we describe the metrics of evaluating the experimental results. Finally, the results and analysis are presented.

At training time, 338 distinct question patterns are identified from 3,806 questions. We aligned these patterns and keywords with bigrams in the 95,926 answer passages. We also identified the locations of the answers and obtained the bigrams appearing within a distance of 3 of the answers. Combining the results of alignment and proximity rank, we derived the transformations for these question patterns.

At run time, we used the top-ranking bigram to expand each question pattern. If no such bigrams were found, we used the keyword in the question patterns only. The expanded terms for question pattern were placed at the beginning of the query. This ordering seems to produce better results than other ways of placing query terms.

We submitted 45 queries to Google and stored 10 summaries returned for evaluation. In the evaluation, we use three indicators to measure *Atlas*’ performance. The first indicator is the mean reciprocal rank (MRR) of the first relevant document (or summary) returned. If the *r*-th document (summary) returned is the one with the answer, then the reciprocal rank of the document (summary) is 1/*r*. The mean reciprocal rank is the average reciprocal rank of all test questions. The second indicator of effective query is the recall at *R* document retrieved (Recall at R). The last indicator measures the human effort (HE) in finding the answer. HE is defined as the least number of passages needed to be viewed for covering all the answers to be returned from the system.

We used a test set of 45 questions which are set aside from the training corpus. The length of these test questions is short. We believe the proposed question expansion scheme helps those short sentences, which tend to be less effective in mining answers. We evaluated the expanded queries in terms of MRR, Recall at R, and HE with ten summaries returned by Google, against the same measures for summaries returned by simple keyword queries. Meanwhile, the ten batches of returned summaries for the 45 questions were verified by two human judges.

As shown in Table 12, the MRR produced by keyword-based scheme is slightly lower than the one produced by the presented query expansion scheme. Nevertheless, such improvement is encouraging and still indicates the effectiveness of the proposed method.

Table 13: Evaluation results of MRR.

| Performances | MRR |
|---|---|
| GO (Direct keyword query for Google) | 0.64 |
| AT+GO (Atlas expanded query for Google) | 0.69 |

Table 13 lists the comparisons in more details. We observe that *Atlas* is effective in bringing the answers to the top 1 and top 2 summaries as indicated by the high Recall of 0.8 at

R=2. In addition, we also find that our approach can obviously reduce user’s effort. For each question, the average of summaries required to be viewed by human beings goes down from 2.7 to 2.3.

Table 14: Evaluation results of Recall at R and Human Effort.

| Rank | Rank count | | Recall at R | |
|---|---|---|---|---|
| | GO | AT+GO | GO | AT+GO |
| 1 | 25 | 26 | 0.56 | 0.58 |
| 2 | 6 | 10 | 0.69 | 0.80 |
| 3 | 5 | 3 | 0.80 | 0.87 |
| 4 | 0 | 1 | 0.80 | 0.89 |
| 5 | 1 | 1 | 0.82 | 0.91 |
| 6 | 2 | 0 | 0.87 | 0.91 |
| 7 | 1 | 0 | 0.89 | 0.91 |
| 8 | 2 | 0 | 0.93 | 0.91 |
| 9 | 0 | 1 | 0.93 | 0.93 |
| 10 | 0 | 0 | 0.93 | 0.93 |
| No answers | 3 | 3 | | |
| Human Effort | 122 | 105 | | |
| # of questions | 45 | 45 | | |
| HE per question | 2.7 | 2.3 | | |

In conclusion, we found that those bigrams containing a content word and a function word turn out to be very effective. For instance, our method tends to transform the pattern “*who invented*” to bigrams such as “*invented by*,” “*invent the*,” and “*inventor of*.” This contrasts to traditional IR scheme in which function words are treated useless and should be removed from the query. Our experiment also shows a function word as part of a phrasal term seems to indicate implied relation with the answer. We also observe that the reason for higher MRR and Recall at R is due to important arrangements in query formulation. Besides, the identification of question patterns has the side effect of putting a critical keyword and its expanded term before the query. Even if there are no transformation produced at the training phrase due to lack of sufficient data, the movement of keyword still is beneficial.

# Chapter 5 Conclusion and Future Work

## 5.1 Conclusion

In this thesis, we exploit statistical word alignment technique and distance constraint for learning transformations from a natural language question to a query effective for mining the answer. The learning strategies involve several steps. First, we automatically acquire relevant passages from the Web for a set of questions and answer passages. We then align question pattern across from questions to answer passages in order to decide the best bigram transformations for a specific question type. Finally, the selection of ranking strategy is applied in such a way that the transformation is statistically associated and positionally close to the answers. At run time, any user input of natural language questions will be automatically processed and transformed into expanded queries on the basis of the question pattern extracted from its question. The evaluation on a set of questions shows that our prototype in conjunction with a search engine outperforms the underlying search engine used alone.

The effectiveness of the proposed method relies on the following features:

- Using seed questions and answers to automatically gather a large number of answer passages on the Web
- Automatically extracting the question pattern from its question by using linguistic analysis
- Word alignment technique originally developed for statistical machine translation to learn relationship between a type of question and effective query terms
- Combining statistical association and position constraint to filter effective “common sense” phrases which may not be linguistically motivated

## 5.2 Future Work

Many future directions present themselves. First, the patterns learned from answer passages acquired on the Web can be refined and clustered to derive a hierarchical classification of questions. Second, different question patterns such as "*who wrote*" and "*which author*" can be treated as the same in order to cope with data sparseness and further booster the performance. Additionally, an interesting direction to explore is the generation of pattern transformations that contain the answer extraction patterns. These answer extraction patterns can be learned for different types of answers. Yet another direction would be to provide confidence factor for ranking the likelihood of many candidate answers extracted using an answer pattern.

# References

Abney, S., Collins, M., and Singhal A. 2000. Answer Extraction. In *Proceedings of the Applied Natural Language Processing Conference* (ANLPNAACL-00). pp. 296-301. Seattle, WA, 2000.

Agichtein, E., Lawrence, S., and Gravano, L. Learning to find answers to questions on the Web. In *ACM Transactions on Internet Technology (TOIT)*, Volume 4, Issue 2, pp.129-162, 2003.

Berger, A., Caruana, R., Cohn, D., Freitag, D., and Mittal, V. O. Bridging the lexical chasm: statistical approaches to answer-finding. In *Proceedings of the ACM SIGIR Conference*. pp. 192-199. 2000.

Berliner, B., Corey, M., and Ochoa, G. The Book of Answers. Barnes & Noble Books, March 2004.

Buchholz, S. Using grammatical relations, answer frequencies and the World Wide Web for question answering. In *Proceedings of the Tenth Text REtrieval Conference (TREC 2001)*. 2001.

Brill, E., Lin, J., Banko, M., Dumais, S., and NG, A. Data-intensive question answering. In *Proceedings of the TREC-10 Question Answering Track*. pp. 393-400. 2001.

Charles L. A. Clarke, Gordon V. Cormack, and Thomas R. Lynam. Exploiting Redundancy in Question Answerin. *In Proceedings of the 24th annual international ACM SIGIR*. pp. 358-365. New Orleans, Louisiana, USA, 2001.

Clarke, C., Cormack, G., Lynam, T., Li, C., and McLearn, G.. Web Reinforced Question Answering (MultiText Experiments for TREC 2001). In *TREC-10 Notebook Papers*. Gaithesburg, MD, 2001.

Echihabi, A. and Marcu, D. A Noisy-Channel Approach to Question Answering. In *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistic.* pp. 16-23. Sapporo, Japan, July 2003.

Girju, R. Answer fusion with on-line ontology development. In *Proceedings of the North American Chapter of the Association for Computational Linguistics (NAACL) - Student Research Workshop*. 2001.

Hermjakob, U., Echihabi, A., and Marcu, D., Natural Language Based Reformulation Resource and Web Exploitation for Question Answering. In *Proceeding of TREC-2002.* Gaithersburg, Maryland, November 19-22, 2002.

Hildebrandt, W., Katz, B., and Lin, J. Answering definition questions with multiple knowledge sources. In *Proceedings of the 2004 Human Language Technology Conference and the North American Chapter of the Association for Computational Linguistics*. pp. 49-56. May 2004.

Hovy, E., Gerber, L., Hermjakob, U., Junk, M., and Lin, C.-Y. Question answering in Webclopedia. In *Proceedings of the TREC-9 Question Answering Track*. pp. 655–672. Gaithersburg, Maryland, November 13-16, 2000.

Ittycheriah, A., Franz, M., Zhu, W.-J., and Rathaparkhi, A. IBM's statistical question answering system. In *Proceedings of the TREC-9 Question Answering Track*. pp. 231–234. Gaithersburg, Maryland, November 13-16, 2000.

Jian, J.-Y., Chang, Y.-C., and Chang, J.-S. Collocational Translation Memory Extraction Based on Statistical and Linguistic Information. In *ROCLING XV (ROCLING 2004)I.* Taipei, Taiwan, September 2-3, 2004.

Katz, B., Lin, J., Loreto, D., Hildebrandt, W., Bilotti, M., Felshin, S., Fernandes, A., Marton, G., and Mora, F. Integrating Web-based and Corpus-based Techniques for Question Answering. In *Proceedings of the Twelfth Text REtrieval Conference (TREC 2003)*. Gaithersburg, Maryland, November 2003.

Kwok, C. C. T., Etzioni, O., and Weld, D. S. Scaling question answering to the web. In *Proceedings of the World Wide Web Conference (WWW-10)*. pp. 150-161. Hong Kong, May 1-5, 2001.

Lin, J. The Web as a Resource for Question Answering: Perspectives and Challenges. In *Proceedings of the Third International Conference on Language Resources and Evaluation (LREC 2002)*. Canary Islands, Spain, May 2002.

Mann, G. Fine-grained proper noun ontologies for question answering. In *SemaNet'02: Building and Using Semantic Networks*. 2002a.

Mann, G. Learning how to answer questions using trivia games. In *Proceedings of the International Conference on Computational Linguistics (COLING-2002)*. Taipei, Taiwan, 2002b.

Melamed, I. Dan. A Word-to-Word Model of Translational Equivalence. In *Proceedings of the 35st Annual Meeting of the Association for Computational Linguistics*. pp. 490-497. 1997.

Mitra, M., Singhal, A., and Buckley, C. Improving automatic query expansion. In *Proceedings of the ACM SIGIR Conference*. pp. 206-214. Melbourne, Australia, 1998.

Moldovan, D., Pasca, M., Harabagiu, S., and Surdeanu M. Performance Issues and error Analysis in an Open-Domain Question Answering System. In *Proceedings of the 40st Annual Meeting of the Association for Computational Linguistic*. pp. 33-40. Philadelphia, PA, USA, July 2002.

Prager, J.M., Radev, D.R., Brown, E.W., and Coden, A.R. The Use of Predictive Annotation for Question-Answering in TREC8. In *Proceedings of TREC8*. Gaithersburg, MD, 2000.

Radev, D. R., Qi, H., Zheng, Z., Blair-Goldensohn, S., Fan, Z. Z. W., and Prager, J. M. Mining the web for answers to natural language questions. In *Proceedings of the International Conference on Knowledge Management (CIKM-2001)*. pp. 143-150. Atlanta, Georgia, November 5-10, 2001.

Ravichandran, D., and Hovy, E. Learning surface text patterns for a question answering system. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*. Philadelphia, PA, USA, July 2002.

Rocchio, J. Relevance feedback in information retrieval. *G. Salton, editor, The SMART Retrieval System–Experiments in Automatic Document Processing*, 313–323. Englewood Cliffs, NJ. 1971.

Salton, G., and Buckley, C. Term weighting approaches in automatic text retrieval. In *Information Processing and Management*. Volume 24, Issue 5, pp. 513-523, 1988.

Soubbotin M.M., Soubbotin S.M. Patterns of Potential Answer Expressions as Clues to the Right Answers. In *TREC 2001 Notebook*. pp. 175-182. Gaithersburg, USA, 2001.

Voorhees, E. and Tice, D. M. The TREC-8 question answering track evaluation. In *Proceedings of TREC-8*. pp. 84–106. Gaithersburg, Maryland, November 17-19, 1999.

**沈天佐, 林川傑, 陳信希. 以網際網路內容為基礎之問答系統**”Why”**問句研究**. In *Proceedings of Rocling 2003*. pp. 211-229. Hsinchu, Taiwan, 2003.

## Appendix - Test Question Set

| Question | Answer |
|---|---|
| With which sport are the Queensberry Rules associated? | Boxing |
| Vienna is the capital of which country? | Austria |
| Who invented the telephone? | Alexander Graham Bell |
| Which sport do the Chicago Bears play? | American Football |
| What does the Dewey Decimal System classify? | Books |
| What color is the middle stripe on the Irish flag? | White |
| Who wrote "Jungle Book"? | Rudyard Kipling |
| Around which war is the 1986 film "Platoon" based? | Vietnam |
| What is the largest desert in the world? | Sahara |
| Who wrote "Frankenstein"? | Mary Shelley |
| In which Scottish city would you find Holyrood Palace? | Edinburgh |
| Which sign of the zodiac is represented by the Ram? | Aries |
| What is ornithology the study of? | Birds |
| In what sport is the "Fosbury flop" technique used? | High Jump |
| What is most expensive property in the board game Monopoly? | Mayfair |
| What is the chemical symbol for the element Hydrogen? | H |
| In which American state is Hollywood? | California |
| What is the longest river in the world? | Nile |
| What color are the spots on Mr Blobby? | Yellow |
| Which part of the body would be treated by a chiropodist? | Feet |
| Gorgonzola cheese comes from which country? | Italy |
| Which sea does the river Thames flow into? | North Sea |

| | |
|---|---|
| Which country in the world has the highest population? | Chinese |
| In which city is the soap opera "Brookside" set? | Liverpool |
| Which European city is nicknamed the "Eternal City"? | Rome |
| In which European country would you find the city of Strasbourg? | France |
| What color is the circle on the Japanese flag? | Red |
| How old is a quadragenarian? | 40 |
| On which London street is Selfridges? | Oxford Street |
| What was Sarah, the Duchess of York's maiden name? | Ferguson |
| Which country is San Marino surrounded by? | Italy |
| Which comedian has the nickname "the big yin"? | Billy Connolly |
| What is the national flower of Wales? | Daffodil |
| What is the largest desert in the world? | The Sahara |
| Which ocean surrounds Hawaii? | Pacific |
| What is the highest number on a roulette wheel? | 36 |
| From which country does the dish Enchilada originate? | Mexico |
| Who painted "The Haywain"? | Constable |
| Which country was once ruled by Tsars? | Russia |
| What is the only mammal which can fly? | The bat |
| What is the only continent which does not have any reptiles or snakes? | Antarctica |
| What is the bestselling book in the world? | The Bible |
| From which country does the dish Enchilada come? | Mexico |
| What is a group of geese called? | A gaggle |
| In which European country is the city of Strasbourg? | France |