# Automatic detection of descriptive phrases for Question Answering System: A simple pattern matching approach

A study submitted in partial fulfilment
of the requirements for the degree of
Master of Science in Information Management

at

THE UNIVERSITY OF SHEFFIELD

by

Hideo Joho

*September 1999*

# Abstract

The increasing interest in providing various information on the Web has heightened the need for a sophisticated search tool. Most existing information retrieval systems, however, merely provides documents, and this often leaves users to read a relatively large amount of full-text. The study of question answering (QA) systems, which enable people to locate the information they need directly from large free-text databases by utilising their queries, has become an important aspect of information retrieval research.

The purpose of this work was to evaluate a Descriptive Phrase Detection (DPD) system that attempted to detect descriptive phrases from a free-text database. A descriptive phrase was a phrase that explained or described a word/noun phrase. Those detected phrases were expected to be the candidates, that could answer a particular class of question in a QA system. Those questions could be 'What is sushi?', 'Who is Bruce Brown?', 'What job does Steve Jobs do?' or 'What does ISDN stand for?' This system employed only simple pattern matching for detection and term frequency for ranking in order to achieve topic domain independence and to allow the use of free-text as the information source.

As a result of the experiment with 57 queries, the system succeeded in detecting a descriptive phrase in more than 70 percent of the queries, and was able to rank the phrases of 60 percent in the top 5, and 70 percent and 80 percent in the top 10 and 20 respectively. These findings suggest that a system which employs only simple pattern matching and term frequency ranking, has the potential to provide the descriptive information of a word/noun phrase, with the use of a free-text database.

# Acknowledgements

I wish to thank Dr. Mark Sanderson for his supervision and encouragement throughout this project. I also wish to thank Professor Micheline Beaulieu for her guidance in the early work. Dr. Andrew D. Madden helped me in checking my English. Thanks are also due to the students and staffs in the Department for providing a number of interesting queries for the experiment.

# Table of Contents

# CHAPTER 1
# INTRODUCTION

## 1.1 Research background

The increasing interest in providing various information on the Web has heightened the need for a more sophisticated search tool, which can locate desired information. Although existing information retrieval (IR) systems have succeeded in searching for relevant documents, most of them just provide some information about the documents. For example, a typical result of using a search engine, one of the IR systems on the Web, is a ranked list of the data such as URLs, titles and the first few lines of a document. As a result, a user usually has to spend a relatively long time looking through those documents, in order (1) to judge their relevance or (2) to find information which he/she wants (or both).

To address the first point, automatic text summarisation systems, have been developed by investigators such as Earl(1970), Paice(1981), Johnson et al.(1993) and Sanderson (1998), which can be regarded as a supporting tool of IR systems. In fact, Tombros and Sanderson's research shows that an automatic text summarisation system can improve the accuracy of the relevance judgements, reducing the frequency of referral to the full text of the documents (1998). However, those attempts still focus on retrieving documents, and users still need to locate the information they desired information by themselves.

The second point seems to indicate the need for a change of viewpoint, from document retrieval to information retrieval. Mention should be made of Information extraction (IE) at this point. Gaizauskas and Wilks (1998) describe the difference between IR and IE systems as being that, where 'IR retrieves relevant documents from collections, IE extracts relevant information from documents.'

One of the characteristics of IE systems is the use of a template, which specifies the sorts of information to be extracted. This provides a framework, which enables the systems to extract accurate and precise information. It is necessary to decide the framework of the template in advance because the results from IE systems are usually intended for use by another system, rather than by a human user. Gaizauskas and Wilks explain such a situation in the following way.

> '… IE may be seen as the activity of populating a structured information source (or database) from an unstructured, or free text, information source. This structured database is then used for some other purpose: for searching or analysis using conventional database queries or data-mining techniques; for generating a summary; for constructing indexes into the source texts.'

Nevertheless, their work is potentially useful for extracting information rather than documents from a source, especially a free text source. This potential is increased by the growing availability of digitised text.

From these observations, it seems fair to say that a system which enables people to obtain the information they seek in a less time-consuming way, should be developed.

There are, undoubtedly, several ways to achieve such a purpose, and question answering (QA) systems are one of them. Salton and McGill describe QA systems, comparing them with IR systems, as follows.

> 'Automatic question-answering systems might be designed where the system is expected to give explicit answers to incoming search requests ("What is the boiling point of water?" Answer: 100 degrees Celsius), as opposed to merely furnishing bibliographic references expected to contain the answer' (Salton and McGill, 1983: p259).

QA systems started to be developed as an application of the field of Artificial Intelligence (AI) in 1960s (Belkin and Vickery, 1985). However, the research has

tended to focus on using parsers, syntactic or semantic analysis to understand questions as well as texts, rather than on providing answers. Moreover, substantial progress has been made only in limited topic areas, limited vocabularies, and syntactic patterns (Salton and McGill, 1983).

The emergence of various information on the Web, as we have seen, also requires a domain-independent information system. The significance of this viewpoint has also been reflected in the Question Answering track in TREC-8 (WWW002 1998). In order to expand the domain as widely as possible, a system which employs simpler and more flexible methods, rather than relying heavily on linguistic analysis, should be developed.

## 1.2 Aim and scope

The aim of this paper is to investigate the effectiveness of a Descriptive Phrase Detection (DPD) system. The DPD system is a system that detects descriptive phrases from a free-text database, using simple text pattern matching. A descriptive phrase (DP) is defined as 'a phrase that explains or describes a word/noun phrase.' For example, a DP of 'AltaVista' could be 'a search engine' or 'an IR system on the Web.' This system is not expected to detect precise definitions, but to detect various phrases that describe a word or noun phrase.

Those phrases that are detected are expected to be the candidates, that could answer a particular class of question in a QA system. Those questions could be 'What is sushi?', 'Who is Bruce Brown?', 'What job does Steve Jobs do?' and 'What does ISDN stand for?' We believe that an approach, which can provide DPs for questions such as those above, will be helpful in the development of an entire QA system. Moreover, those phrases may be possible candidates for another form of question, since they could have various attributive information about the word or noun phrase.

One may think that this system is similar to online dictionary/encyclopaedias such as whatis.com, which retrieves definitions or explanations of abbreviations or technical terms. The difference is, that while somebody in advance makes the

retrievable definitions of whatis.com, the DPD system will produce DPs from free-text databases by text analysis. As a result, the DPs may be clumsy and less readable, but they should still be meaningful. Most importantly however, they provide the potential to offer far wider and more flexible coverage than could be offered by whatis.com, because the system tries to find the DPs for any word or phrase.

To sum up, therefore: the objective of this project is to produce a DPD system which will be

- independent of topic domain
- capable of using free-text information sources
- applicable to a system which provides a large quantity of information in digitised text format.

## 1.3 Brief description of methodology

In order to achieve the objectives, the system will employ techniques from the field of information retrieval, and other related fields such as information extraction or automatic text summarisation. More specifically, the DPD system will employ simple text pattern matching as the detecting method, and a formula based on inverse document frequency (IDF) as the main ranking method.

Pattern matching techniques have been widely used in automatic indexing, syntactical text analysis and other natural language processing (Salton 1989). Hearst (1998) employed a simple pattern matching in order to extract lexicon-syntactic patterns, which are used for identifying lexicon-semantic relations between the words in WordNet, a lexical database. Her method, the Lexicon-Syntactic Pattern Extraction (LSPE), does not require a knowledge-based algorithm or tool for helping the extraction. Instead, simple text fragments are used. An example text fragment is 'such as', which may be found in a sentence like the following.

> "He used several search engines such as AltaVista, HotBot and goo in order to compare the performance."

Even if a reader has not heard of 'goo' he/she will actually understand that goo is a kind of search engine. Consequently the 'search engines' can be a description of goo. Similarly, several text fragments such as 'and other' and 'or other' will be used as patterns.

The DPs which are detected by the pattern matching will be ranked by a formula based on inverse document frequency (IDF). IDF is a technique which is also commonly used to rank the results of best-match IR systems, or to extract sentences in automatic text summarisation by determining term frequency. 'IDF factor varies inversely with the number of documents *n* to which a term is assigned in a collection of *N* documents. A typical IDF factor may be *log N/n*.' (Salton, 1989). The rationale underlying this technique is

> 'the number of documents relevant to a query is generally small (compared with a whole database), and thus any frequently occurring terms must necessarily occur in many irrelevant documents; infrequently occurring query terms, conversely, have a greater probability of occurring in relevant documents and should thus be considered as being of greater potential importance' (Spark Jones and Willett, 1998: p.307).

The evaluation scheme will be divided into two parts: the overall system and the patterns which form the basis of the matches. Although the DPD system is not an entire QA system, we will use an evaluation method for QA systems as a part of the overall evaluation. It seems worthwhile figuring out the performance of the system as a QA system, because the system is trying to detect candidate answers.

The final point is the criteria of validity for the answers. For instance, take the answer to a question 'What is the sun?' The system may answer that 'The sun is a star' or, the answer may be 'The sun is a vital element for life.' We would like to regard both of them as the correct answers. In other words, by contrast with an IE system, which tends to extract very precise answers the DPD system, as well as a QA system, does not need to make an entirely correct answer, because the second answer may be more appropriate for the user's requirements. This rather lax criterion will

increase the chance that the system will find more candidate answers, and it is for this reason that the system can employ only simple methodology.

## 1.4 Rest of the work

Chapter 2 provides a critical review of the literature concerning DPD systems, the major approaches employed, and the evaluation issues raised. Chapter 3 describes the architecture of the DPD system, the experimental used, and the scheme by which both the system and the patterns were evaluated. Results of the experiments are presented in Chapter 4 and discussed in Chapter 5. Chapter 5 also includes the conclusions of this dissertation, which are followed by several suggestions for future work.

# CHAPTER 2
# LITERATURE REVIEW

As explained in the previous chapter, the objectives of this study is in are to develop a detection system by a simple pattern matching, and to us this to produce a question answering system. Therefore, this chapter will begin with a review of question answering systems, after which, consideration will be given to the techniques that have been employed in the processes of the DPD system. More specifically, the techniques for detecting and ranking will be examined. Finally, issues concerning evaluation of the system will be discussed.

## 2.1 Question Answering System

Salton and McGill (1983) describe question answering (QA) systems as 'specially designed to provide direct answers to questions.' Early QA systems were developed by the researchers on Artificial Intelligence (AI) or knowledge-based techniques as a means of applying the findings of their studies. More recently, there has been growing interest in developing QA systems for use in information retrieval. In this section therefore, those two applications of QA systems will be discussed.

### 2.1.1 Early QA systems

Early QA systems, in common with other expert systems, employed the techniques developed by AI or Knowledge-based studies. Smith (1980) defined research in AI as 'efforts aimed at studying and mechanising information-processing tasks that normally require human intelligence' (cited in Belkin and Vickery 1985). Barr (1982) also sees 'human intelligence' in terms of human behaviour, such as knowing, reasoning, learning, problem-solving, and language-understanding.

From their perspective, QA programs are ones 'that are intended to simulate some or all aspects of human linguistics behaviour', and it is supposed that 'a machine could be built that understood English perfectly, that remembered what it was told about as well as human beings do, that could respond to questions, and so forth' (Kay and Sparck-Jones, 1971 cited in Belkin and Vickery, 1985).

As a result, understanding natural language by machine has been an essential prerequisite for the development of a QA system, as well as other systems in this field. However, processing natural language is particularly difficult because of its varied and complex nature. Salton and McGill (1983) show six levels of language processing as follows.

- *Phonological* level: deals with the treatment of speech sounds as needed, for example, for the handling of speech understanding or speech generation systems.
- *Morphological* level: deals with the processing of individual word forms and of recognisable portions of words.
- *Lexical* level: deals with the procedures that operate on full words.
- *Syntactic* level: deals with grouping the words of a sentence into structural units such as prepositional phrases, and subject-verb-object groupings that collectively represent the grammatical structure of the sentence.
- *Pragmatic* level: to help in the text interpretation, additional information is used about the social environment in which a given document exists, about the relationships that normally prevail in the world between various entities, and about the world-at-large.

In addition, understanding of natural language requires the integration of the processes shown above. However, Salton and McGill say that 'it is unclear which levels of language processing are most important and how the corresponding techniques are best applied.' These difficulties in understanding language have consequently led the development of QA systems by AI or knowledge-based techniques, into restricted topic domains. For example, BASEBALL developed by Green et al. (1961), one of the first QA systems, covered only the games played during one season of the American League.

Much of the significant early research in QA systems involved modelling the question forms. Kearsley (1976) developed a taxonomy of question forms and functions. Table 2.1 shows his model of wh-questions (i.e. questions formed by interrogative words, such as who, what, when, why).

| | | |
|---|---|---|
| Who (Whom) | 1. Unique person specification<br>2. Role specification | Who is that? |
| Where | 1. Geographical/common knowledge<br>2. Relative location<br>3. Shared private knowledge | Where does he live? |
| When | 1. Objective date<br>2. Relative time<br>3. Personal age<br>4. Shared private knowledge | When were you there? |
| How | 1. Evaluative (ascriptive)<br>2. Evaluative (nonascriptive)<br>3. Explanation of procedure<br>4. Justification | How are you?<br>How many are there?<br>How do you play this?<br>How come I always lose? |
| Why | 1. Justification of reasons<br>2. Puzzlement<br>3. Information<br>4. Explanation | Why did you do that?<br>Why doesn't it work?<br>Why do you ask?<br>Why did it happen? |
| What | 1. Specification of objects, activity, definition | What kind is that? |
| Which | 1. Specification of objects, attributes | Which book do you want? |
| Whose | 1. Specification of ownership | Whose car is it? |

**Table 2.1** Kearsley's model of wh-questions (source: Vickery and Vickery (1987), p184)

Kupiec (1993) employed similar but rather simpler wh-question models to build a QA system. He used the interrogative words for informing the kinds of information required by the system.

| | | |
|---|---|---|
| Who/Whose: | *Person* | |
| What/Which: | *Thing,* | *Person, Location* |
| Where: | *Location* | |
| When: | *Time* | |
| How Many: | *Number* | |

**Table 2.2** Kupiec's model of wh-questions

Needless to say, such systematic understandings of the nature of questions are crucial for those who are involved in the study and development of QA systems. More detailed reviews of QA systems with AI techniques and question forms can be found in the chapter 3 and chapter 8 of the report written by Belkin and Vickery (1985).

### 2.1.2 QA system as search tool

As was stated above, in addition to the use of QA systems for research into artificial intelligence, interest has focused on their possible applications in the field of Information Retrieval (IR). Research in this area focuses on the development of QA systems as another form of search tool. A considerable amount of information is currently available in digitised form, so with the aid of networked computers, we have, in theory, ready access to such information via the Internet. However, although such access is theoretically available, in practice, because most information on the Web is not structured, most of it only becomes readily available by means of sophisticated search tools.

IR systems have been developed to help people to find information, and as such, have played a very important role in the information society. In most cases, the 'information' is synonymous with a 'document' or 'information about documents' in IR. Therefore, IR has often been called 'document retrieval.' In other words, 'information retrieval deals with the representation, storage, and access to documents or representatives of documents' (Salton and McGill 1983). However, this 'document' oriented interaction between the users often leaves them to locate the 'information' they desire by themselves. More specifically, a user is required to (1)

judge the relevance of documents provided by an IR system, and (2) locate the information in full text; both of which are time-consuming processes.

To address the first point, automatic text summarisation systems, have been developed by investigators such as Earl(1970), Paice(1981), Johnson et al.(1993) and Sanderson (1998), which can be regarded as a supporting tool of IR systems. In fact, Tombros and Sanderson's research shows that an automatic text summarisation system can improve the accuracy of the relevance judgements, reducing the frequency of referral to the full text of the documents (1998). However, those attempts still focus on retrieving documents, and users still need to locate the information they desired information by themselves.

From these observations, it is obvious that a search tool, which enable users to locate information directly from documents, should be developed. In addition, we can intuitively see that the form of question-answering is suitable for the interaction between such a search tool and the users. The queries will be changed from a set of keywords to questions, and the response of the system will be changed from documents to answers. Consequently, a QA system can be another form of the search tools.

It should also be noted that IR systems usually require structured databases as the information source. The advantage of the use of structured information source is that can enhance the accuracy or processing speed of the system, and the disadvantage is the needs of another software for building such a database. In this context, information extraction (IE) systems are interesting because not only of the attempting of extracting various kinds of information (rather than documents), but also of the use of free-text as the information source.

Gaizauskas and Wilks(1998) describe IE as 'a term which has come to be applied to the activity of automatically extracting pre-specified sorts of information from short, natural language texts', and the contrast between IR and IE systems as 'IR retrieve relevant documents from collections, IE extracts relevant information from documents.' One of the characteristics of IE systems is use of a template, which specifies the sorts of information extracted. This enable the systems to extract

accurate and precise information, while one will be required to decide a framework of the template in advance.

However, as can be seen in the use of the templates, IE systems are often intended to be used for another system, rather than human activity. Gaizauskas and Wilks explain such a situation in the following way.

> 'IE may be seen as the activity of populating a structured information source (or database) from an unstructured, or free text, information source. This structured database is then used for some other purpose: for searching or analysis using conventional database queries or data-mining techniques; for generating a summary; for constructing indexes into the source texts.' (Gaizauskas and Wilks 1998: p.70).

Up to here in the subsection, we have seen another stream of QA systems as a search tool, and other related works such as IR, text summarisation and IE. The emphasis was on the needs of a search tool, which can locate information desired from free text information source. In addition, the development of the QA system without domain restriction could also be a considerable point, from the observation of the early QA systems in the previous subsection. In this context, the integrated techniques which are derived from these related field will be worth employing for developing a QA system. This perspective is also reflected in the international conference of IR, Text Retrieval Conference (TREC), by the QA track (WWW002).

## 2.2 Relative techniques to the DPD system

Although various types of QA systems may exist, the systems that we assume have some hypotheses as follows.

- Relatively large amount of free-text database is available for the information source of the system.
- In principle, the minimum requirement of the system is that it provide a single answer, though in many cases it may return more than one.
- The answers to be returned can be vaguer than the data processed by other

information systems, since the user is human.

As a result, the DPD system will also be developed on these assumptions.

### 2.2.1 Detecting method

Based on the hypotheses, we will employ *pattern matching* as the main technique for detecting the descriptive phrases. Pattern matching techniques have been widely used in automatic indexing, especially in which syntactic analysis determines the words or phrases to be indexed (Salton 1966, Dillon and Gray 1983). More recently, a number of researchers has attempted to extract semantic information from free text databases (Alshawi 1987, Nakamura and Nagao 1988, Wilks et al. 1990). Hayes et al. (1988) employed the pattern matching for categorising news stories.

In the context of some aspects of natural language processing (i.e. semantic or syntactical), pattern matching techniques sometime are compared with *parsing*. Ahlswede and Evens (1988), who compared an approach based on parsing with one based on pattern matching, describe a parsing as 'a computational technique of text analysis drawing on an extensive database of linguistic knowledge, e.g., the lexicon, syntax and/or semantic of English' and a pattern matching ('text processing' in their paper) as 'any computational technique that involves little or not such knowledge.'

There is little agreement which method is more suitable for the works related to text analysis. Ahlswede and Evens (1988) concluded that full natural language parsing is not an efficient procedure for gathering lexical information, while a number of researchers attempted to text analysis by parsing (Cutting et al. 1992, Kim and Moldovan 1995, Charniak et al. 1996). However, there seems some consensus of pros and cons of both approaches. That is, pattern matching is simple and fast, but not so suitable for complex text analysis, while parsing is able to analyse such a complex text, but is slow.

Whereas we recognise of importance of the accuracy in text analysing, it seems that very accurate text analysing, based on extensive linguistic knowledge is not required for our purpose. We aware of processing speed as well as accuracy since

the DPD system will use a large free text database as information source. Radev (1998) comments that 'rules at such a detailed syntactic level take too long to process on a 180 MB corpus' and notes that 'using syntactic information on such a large corpus does not appear particularly feasible.' Although there are few report that shows actual data size and its processing time, such an approach does not seem suitable for the DPD system. Nevertheless, it should be noted that there are cases which make use of the advantages of these two approaches (e.g., Jacobs and Rau 1990, Kupiec 1993, Black et al. 1998).

In such a context, Hearst's work, which employed pattern matching for discovering lexico-semantic relation from WordNet, seems interesting and useful for the DPD system. WordNet is an on-line hierarchical lexical database which contains semantic information about English words (see Miller (1995) and Fellbaum (1998) for detail). Her method, the Lexicon-Syntactic Pattern Extraction (LSPE), does not require a knowledge-based algorithm or tool for helping the extraction. Instead, simple text fragments are used. An example text fragment is 'such as', which may be found in a sentence like the following.

> "He used several search engines such as AltaVista, HotBot and goo in order to compare the performance."

Even if a reader has not heard of 'goo' he/she will actually understand that goo is a kind of search engines. Consequently the 'search engines' can be a description of goo. Similarly, several text fragments such as 'and other' and 'or other' will be used as patterns. The significance of her approach can be found in which these text fragment can locate both a word and its description, and in which those *identifiers* are available in free text.

In the linguistic point of view, her method can be regarded as describing a word by extracting its *super-ordinate* concept (*hyponym* in her paper). That is, the goo is *a kind of* search engines, in the above example. Radev (1998), on the other hand, attempted to describe a word (especially people's name) by extracting its *appositive* concept. This example may be found in the following sentence.

"U.S. president Bill Clinton president met Japanese prime minister Keizo Obuchi …"
or
"Sun Microsystems, the leading workstation manufacturer, appealed … "

'U.S. president' and 'Japanese prime minister' are the appositive concepts of 'Bill Clinton' and 'Keizo Obuchi' respectively, and 'the leading workstation manufacturer' is of 'Sun Microsystems.' As can be seen, 'appositives are used in English to further specify the meaning of the noun they follow' (Gershman, 1982 cited in Coates-Stephens, 1993). Coates-Stephens also claims the significance of appositives as follows.

'Anything can be described by an appositive, and the method of deriving the descriptive information is the same in every case' (1993: p.447).

Up to here, we have been focused on pattern matching and its feasibility in the DPD systems. However, as mentioned before, most pattern matching approaches to text analysis has lack of real understanding, and this causes 'the lack of discrimination in the output' (Salton 1989). In other word, less accurate pattern matching sometimes brings unwilling outputs, which is often call 'noise.' As a result, the process of *ranking* becomes important to decrease the noise from the results of pattern matching process.

### 2.2.2 Ranking method

The point about ranking is what can be the key for sorting. The notion of *similarity* of the documents to a query has been used for such a key in IR since the early age. A similarity is usually determined by *term weighting*. The weighting of a term defines 'the significance of a term for an individual document or query as some function of its frequency in the document itself and its frequency in the document set' (Spark Jones and Willett 1998). The IR systems, which employ several processes based on the term weighting, are often called 'vector model', as opposed to Boolean model, which is based on set theory and Boolean algebra (Baeza-Yate and Ribeiro-Neto 1999).

The term weighting comprises two major components: term frequency (TF) and inverse document frequency (IDF). TF is the raw frequency of a term inside a document. IDF is the inverse of the frequency of a term among the documents in the collection. Baeza-Yate and Ribeiro-Neto (1999) define TF and IDF as follows.

Let $N$ be the total number of documents in the system and $n_i$ be the number of documents in which the index term $k_i$ appears. Let $freq_{i,j}$ be the raw frequency of term $k_i$ in the document $d_j$ (i.e., the number of times the term $k_i$ is mentioned in the text of the documents $d_j$). Then, the normalised frequency $f_{i,j}$ of term $k_i$ in document $d_j$ is given by

$$f_{i,j} = freq_{i,j} / max_l \ freq_{l,j}$$

where the maximum is computed over all terms which are mentioned in the text of the document $d_j$. Further, let $idf_i$, inverse document frequency for $k_i$, be given by

$$idf_i = \log N / n_i$$

Spark Jones and Willett explain that 'the basis for IDF weighting is that the observation that people tend to express their information needs using rather broadly defined, frequently occurring terms, whereas it is the more specific, i.e., low-frequency, terms that are likely to be of particular importance in identifying relevant material.' Thus, 'the number of documents relevant to a query is generally small, and thus any frequently occurring terms must necessarily occur in many irrelevant document; infrequently occurring query terms, conversely, have a greater probability of occurring in relevant documents and should thus be considered as being of greater potential importance.'

Furthermore, a term weighting scheme is determined by the balance of these two factors. The best known term weighting is given by

$$w_{i,j} = f_{i,j} \times \log N / n_i$$

although various kinds of combination can be found in Salton and Buckley (1988).

The significance of this ranking method can be found in the fact that the vector model IR systems show the same or better performance than other model, despite of its simplicity. Moreover, the wide applicants of this method, such as automatic indexing, relevant feedback (e.g., Yu and Salton 1976, van Rijsbergen 1979, Salton and Buckley 1988), proves the flexibility.

## 2.3 Evaluation issues

Up to here, the two techniques, pattern matching and ranking, have been reviewed. They are important since the DPD system will employ them as the main processing. At the same time, evaluation of the system must also be considered. However, as described in Harter and Hert (1997), the current evaluation issues are rapidly expanded. Therefore, we will focus on the measures of evaluation in this section.

### 2.3.1 Measures for IR systems

'Evaluation means assessing performance of value of a system, process (technique, procedure…), product, or policy' (Saracevic 1995). In order to evaluate such a performance, several measures have been considered. In IR, for instance, *recall* and *precision* are broadly used for the evaluation. Given set *R* of relevant documents to a query in a collection, and set *A* of the documents retrieved by a system to the query, and set *Ra* of the relevant documents in the set *A*, recall and precision are given by

$$\text{Recall} = Ra / R$$
$$\text{Precision} = Ra / A$$

In other word, recall is the ratio of relevant items retrieved to all relevant items in a collection and precision is the ratio of relevant items retrieved to all retrieved items (Saracevic 1995). These two measures have been used not only for IR systems, but also for evaluating other systems, such as Information Extraction (e.g. in Message Understanding Conference).

Recall and Precision are based on the number of relevant documents. Determining a document whether relevant or not is usually referred to as *relevance judgement*. The relevance judgement is one of the fundamental issues in the research of IR, and also seems to be relevant to this project.

Cuadra and Katter (1967, cited in Salton & McGill, 1983) define *relevance* as 'the correspondence in context between an information requirement statement (i.e., a query) and an article (a document), that is, the extent to which the article covers the material that is appropriate to the requirement statement.' Such relevance is often called *stated* relevance. On the other hand, the relevance, which is based on the value of the document for a particular user at a particular point in time, is often called *user* relevance (Tague-Sutcliffe 1996). In both cases, however, a person judges the relevance of a document. Thus, one may say that relevance judgement are a function of one's mental state at the time a reference is read. 'Therefore, they are not fixed, but dynamic' (Harter 1992). Belkin (1981) comments that 'they are subjective and intangible. Thus, most performances of the effectiveness of IR systems are based on such an *ineffable* concepts.'

This fundamental nature of relevance judgement is well described as the dilemma of IR research by Ellis (1996). He introduces one episode that describes such a situation of relevance judgement. This was about an early experiment of IR systems in 1953, called the Uniterm test. The test was a contest between two organisations, concerning the subject heading system, using 98 questions applied to a test collection of 15,000 technical documents. Before the results could evaluated, it was to decide jointly which among the retrieved documents were relevant or not.

> 'Both team agreed that 1390 documents were relevant to one or more of the 98 questions, but there were another 1577 documents that one team or the other, but not both, considered to be relevant - a colossal disagreement was never resolved' (Swanson 1988: p.555).

This will not be often the cases with the current situation. Nevertheless, we can see how difficult people have consensus agreement on the relevance judgement from this small episode. This situation may also occur to the judgement of validity of the

answers provided by a QA system. In other words, different people may find an valid answer from different descriptions.

### 2.3.2 Measures for QA systems

Turning now to the measures for QA systems. Mean reciprocal answer rank (MRAR) is the measure adopted by the question answering track of Text Retrieval Conference (TREC). MRAR is given by the mean of reciprocal answer ranks. A reciprocal answer rank of a query is given by

Reciprocal answer rank = 1 / Answer rank

In this measure, a system is required to rank the answers in advance, and the top 5 ranked answers are used. If the answer is found at multiple ranks, the best (lowest) rank will be used. If an answer is not found in the top 5 ranks, the score for that query is zero (WWW 002).

Mani et al. (to be print) employ a method that consists of three measures such as Answer Recall Lenient (ARL), Answer Recall Strict (ARS) and Answer Recall Average (ARA), for evaluating the summaries by a question answering task. The first two measures are given by

$$ARL = (n1+(0.5*n2)) / n3$$
$$ARS = n1 / n3$$

Where $n1$ is the number of correct answer, $n2$ is the number of partially correct answer, and $n3$ is the number of questions answered. ARA is the average of ARL and ARS.

One of the significant differences between MRAR and Mani's method can be seen in the degree of validity of answers. In other words, MRAR requires the judgement whether the provided answers are correct or not, while Hani's method requires correct, partially correct or missing. In this context, Mani's method presumes one to be able to judge the answers in more detail than MRAR. This may

be possible since the summaries are available for judging the validity of answers in Mani's case.

Up to here, we have focused on the measures for evaluation, especially in IR and QA systems. We also mentioned the validity of answers in the context of relevance judgement in IR. However, as mentioned before, the evaluation in IR involve a number of issues that was not addressed here, and little is known the evaluation scheme of QA systems. Reader may want to refer to Harter and Hert (1997) for a comprehensive discussion of the evaluation issues in information retrieval.

# CHAPTER 3
# METHODOLOGY

In this chapter, the architecture of the DPD system and its main processes, design of the experiment, and the evaluation scheme of the system and the patterns employed will be described.

## 3.1 Descriptive Phrase Detection(DPD) System

The descriptive phrase detection (DPD) system was that detected descriptive phrases from a free text database. A descriptive phrase (DP) was a text fragment or a sentence which explains or describes a word/noun phrase. The DPs detected were expected to happen to answer a particular class of question.

In this section, the architecture of the system will be shown, and the main processes, such as detection of DPs and ranking, will be then described in the following two subsections.

### 3.1.1 Architecture of the system

The flow of main process in the system was shown in Figure 3.1 as below, followed by the explanations.

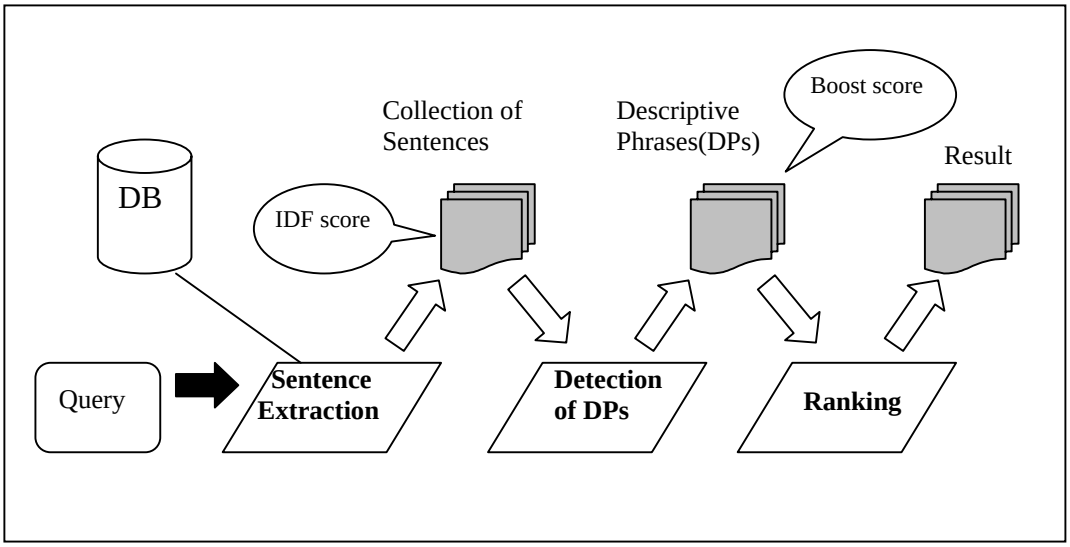


**Figure 3.1** Flow of main process

**Extraction of sentences**: all sentences which contain a query were extracted from a free text database. As a result, a document, which we shall call a *relative sentence collection*, was generated. Each sentence was then given a score that indicates its significance, which we shall call *sentence weighting score*, based on inverse documents frequency (IDF). The score was used for a factor of ranking. (See 3.1.3 for detail)

**Detection of descriptive phrases**: descriptive phrases of a query were detected by pattern matching from the relative sentence collection. A sentence that contained a descriptive phrase was then given another score, which we shall call a *boost score*, determined by the pattern matched. A sentence that did not match any pattern was also given 'one' as the score. This score was used for ranking as well as the score given in the previous process.

**Ranking**: the system employed three ranking methods. One was by the sentence weighting score. Another was by the boost score. And the last method was by the combination of these two scores.

### 3.1.2 Detecting descriptive phrases

Descriptive phrases were detected by a simple pattern matching. The patterns can be

divided into three groups: text fragments, appositives, and acronyms.

**Text fragment**: this group detected a descriptive phrase using a certain text fragments. We shall call this text fragment an *identifier*. An example of the first group was as follows.

" … large computer company such as IBM …"

The identifier of this example was *such as*. The identifier located the descriptive phrase about IBM as well as its position in a sentence. This enabled the system to detect the descriptive phrase automatically. The first group had seven such identifiers as shown in Table 3.1, with their conditions to be matched, where DP stood for a descriptive phrase and X indicated a query.

| Identifier | Condition |
|---|---|
| Such as(1) | … DP such as X … |
| Such as(2) | … such DP as X … |
| And other | … X and other DP … |
| or other | … X or other DP … |
| Including | … DP, including X … |
| Especially | … DP, especially X … |
| Is a | X is a DP … |

**Table 3.1** Identifier

**Appositive**: this group detected a descriptive phrase by use of appositive text of a query. Therefore, this group did not have text fragments as the identifier, but the query itself and two commas between a descriptive phrase were the identifiers. The example was as follows.

" … Yukio Mishima, the great Japanese novelist, was …"

'The great Japanese novelist' was the descriptive phrase about 'Yukio Mishima'. However, since this pattern matching rule seemed too lax, further factors were added to the identifier as shown in Table 3.2, where DP stood for a descriptive phrase and X indicated a query.

| | Condition |
|---|---|
| 1 | … X, DP, is/was … |
| 2 | … X, DP, was/were … |
| 3 | … X, which DP, … |
| 4 | … X, the DP, … |
| 5 | … X, a DP, … |

**Table 3.2** Appositive matching rules

**Acronyms**: this group attempted to extract the original name of an acronym or abbreviation, rather than detecting descriptive phrases. However, we believed that such original names would be meaningful, since it is often the case that a person wonders what an acronym symbolises. The examples of this group were as follows.

" … acquired immune deficiency syndrome (AIDS) … "

or

" … Fed (Federal Reserve Board) …"

Therefore, the matching conditions were as follows, where DP stands for a descriptive phrase and X stands for a query.

… DP(X) … or … X(DP) …

These three groups of pattern matching were executed to a relative sentence collection that gathered from a free text database. All sentences in the collection were then given a boost score depending on the type of pattern matched. The boost score shown in Table 3.3 was decided by an ad hoc observation. This boost score was used for ranking described in the following subsection. In addition, the sentences that did not match any pattern were also given 1 as the boost score.

| Pattern | Boost score |
|---|---|
| is a | 9 |
| Acronym | 9 |
| Such as(1) | 7 |
| Appositive | 7 |
| Such as(2) | 5 |
| Especially | 5 |
| Including | 5 |
| And other | 3 |
| Or other | 3 |

**Table 3.3** Boost score

### 3.1.3 Ranking

The system employed three ranking methods. One was based on sentence weighting scores, and another was based on boost scores, and the last was based on the combination of the previous two scores. All scores were given to a sentence in a relative sentence collection. The followings were the description of each scoring.

**Sentence weighting score**

A sentence weighting(SW) score was the average of inverse document frequency (IDF) scores of all terms in a sentence. Each term in the sentence had two IDF scores; one was of a whole free text database(1), and another was of a relative sentence collection(2). An IDF of a term was figured by the following way.

IDF = log N/n

Where N was number of documents in a collection, and n was the number of document that contain the term. The collection was the DB in the case of (1), and a relative sentence collection in (2). The number of *sentence* was regarded as that of *document* in the case of (2). The IDF score of each term was then figured as follows.

IDF score = IDF(1) - IDF(2)

This procedure was intended to indicate the difference of frequency between in the database and in a relative sentence collection. In other words, if a term that did not occur frequently in the database but occurred frequently in the relative sentence collections, then this could show the significance of such a term.

This was based on a hyphosis that there might be a number of descriptions which used some particular terms. For example, 'Tony Blair' is often associated with 'prime minister', 'U.K.', and the combination. Therefore, a sentence that contains much of those terms seemed to have a descriptive phrase.

Thus, a sentence weighting score was figured by the average of the IDF score of all terms contained as follows. This enables the sentences with the most number of

unusually frequent terms to receive the highest score.

Sentence weighting score = sum of all IDF scores/number of terms

As can be seen, this scoring method was intended to rank all sentences in a relative sentence collection, by their significance. Therefore, the result was not determined whether to match a pattern or not.

**Boost score**

The sentences that matched any pattern, which was described in the previous subsection, were given a boost score depending on a type of the pattern matched. The score ranged from three to nine (see 3.1.2 for detail). The sentences that did not match any pattern, were given '1' as their boost score. This scoring method was intended to reflect the result of pattern matching strongly, as opposed to sentence weighting scoring method.

**Sentence weighting & boost score**

The last method of ranking was based on the combined score of the previous two scores. The combined score was figured by the following way.

Combined score = sentence weighting score * boost score

This scoring method was intended to maximise the characteristics of two scores, such as sentence weighting score and boost score. In other words, this method attempted to give the priority to the sentences, that matched a descriptive phrase by the pattern matching , as well as respecting the significance of each sentence. Therefore, this method was expected to show the best performance among the three methods.

## 3.2 Description of experiment

The experiment conducted was quite straightforward. Giving a query to the system and gathered the result. In this section, therefore, the queries and information source

will be described. Nevertheless, the reflections of the objectives of the system, such as domain independent and use of free text as information source, can be seen in some of the point described in this section.

### 3.2.1 Query

A query in this project was a word or noun phrase or acronyms/abbreviation. No restriction on the topic of queries was specified. Particularly, we were interested in detecting the names of people, companies, or associations, as well as specific terms. This was because ordinary vocabularies seemed less meaningful to be described by the DPD system.

All queries were not confirmed their existence in the database in advance, nor the presence of their descriptive phrases. Therefore, a *valid query* was determined by the following two conditions.

(1) There was at least one sentence that contains the query in the database,
as well as
(2) there was at least one sentence that contains the descriptive phrase about a query.

### 3.2.2 Information source

As an information source, we chose the full-text database of LA Times in 1989 to 1990, which is available on TREC (WWW001 1998). The database contained approximately 460 MB of news articles. For the experiment, the database was divided almost evenly into two parts: training set and testing set. The training data was used for developing the system and ad hoc testing for ensuring the work, while the testing data was used only for the experiment. In other words, the system did not do any training on the testing set.

In addition, it should be noted that this database might be unfairly suitable for the system, since most of the documents were written by journalists, and they tend to explain unfamiliar terms or specific phrases in the articles. Nevertheless, we believed that news articles could be enough for evaluating the system, in terms of use of free text information source.

## 3.3 Evaluation scheme (1) Effectiveness of the system

The evaluation scheme was mainly divided into two part: effectiveness of the system and effectiveness of the patterns employed. In this section, the first part will be described. The evaluation of the system was compared by the three ranking methods described in 3.2.3.

### 3.3.1 Mean Reciprocal Answer Rank (MRAR)

This evaluation method, which was called Mean Reciprocal Answer Rank, is adopted by the QA track of coming TREC-8 (WWW002, 1998). The score was calculated as follows.

Mean Reciprocal Answer Rank = 1 / Answer rank for query

Ranking the top five answers was a prerequisite for this method. If an answer was not found in top five ranks, the score for that query was zero. This approach gave a score to each query. The sum of all scores was then divided by the number of queries, and each ranking method was given this divided score.

There were two main reasons for using this evaluation method, despite the DPD system is not an entire question answering(QA) system. One was to figure out the performance of the system as a QA system. In other words, it was expected to show how well the system works in responding to the needs of a QA system. Another was to use a well-established method. In general, little is known about the evaluation scheme of QA system. Nevertheless, we regarded this TREC's method as one of the most established method at the moment.

### 3.3.2 Detection rate in top 20

The previous method, MRAR score was, however, rather restricted and too specific for evaluating the DPD system. Therefore, we prepared another criteria for the evaluation: detection rate in top 20. This attempted to figure out the number of the sentence that contained descriptive phrases, and its percentage in the top 20 ranks.

The percentage was given both at the query level and at the sentence level, in the top five, ten, and twenty, by each ranking method. In addition, this criteria was intended to show the system's performance in wider view, and to be complementary to MRAR.

## 3.4 Evaluation scheme (2) Effectiveness of the patterns

Apart from the overall evaluation of the system, performing the effectiveness of the patterns in detail, also seemed to be meaningful since this was the key function of the system. The evaluation scheme of effectiveness of the patterns were divided into three aspects: overall, coverage, accuracy.

### 3.4.1 Overall

Overall effectiveness of the pattern matching was performed by two aspects: one was by the number of the query that the patterns succeeded to detect a descriptive phrase, and another was by the detected rate depending on the size of a relative sentence collection. The former was aimed to show general effectiveness of the pattern matching, and the latter was aimed to show the effectiveness in terms of size of source.

### 3.4.2 Coverage

Each pattern was figured out its coverage and accuracy. The coverage were based on the number of the query or sentence that matched a pattern (i.e. not on the query that had a descriptive phrase). For example, if a pattern succeeded to detect a correct answer to 10 queries in a whole 50 queries. The coverage was 20 percent.

Another aspect of the coverage was the distribution rate of the patterns. This was the ratio of the sentences detected by a pattern in a whole sentences detected by all patterns. This ratio was intended to reveal the relative amount of the sentence that detected by each pattern.

### 3.4.3 Accuracy

Like the measure of coverage, an accuracy of a pattern was also computed at two levels. At the query level, the accuracy was the number of query that had a descriptive phrase in the number of query that matched a pattern. At the sentence level, the accuracy was the number of sentence that contained a descriptive phrase in the number of sentence that matched a pattern.

# CHAPTER 4
# RESULT

In this chapter, the results from the experiments are presented. As mentioned in the previous chapter, the results will be divided into two parts: effectiveness of the system and effectiveness of the patterns.

## 4.1 DPD System

Seventy six queries were used for the experiment. Of the 76 queries, ten did not exist in the database at all. Of the remainder, 57 queries had at least one sentence that contained a descriptive phrase, and nine did not. Therefore, the 57 queries were regarded as the *valid queries* for the evaluation of the system.

The evaluation scheme for the system was based on three ranking methods: sentence weighting rank; boost score rank; and combined rank. All three methods were performed by Mean Reciprocal Answer Rank score and detection rate in the top 20.

### 4.1.1 Mean Reciprocal Answer Rank (MRAR) score

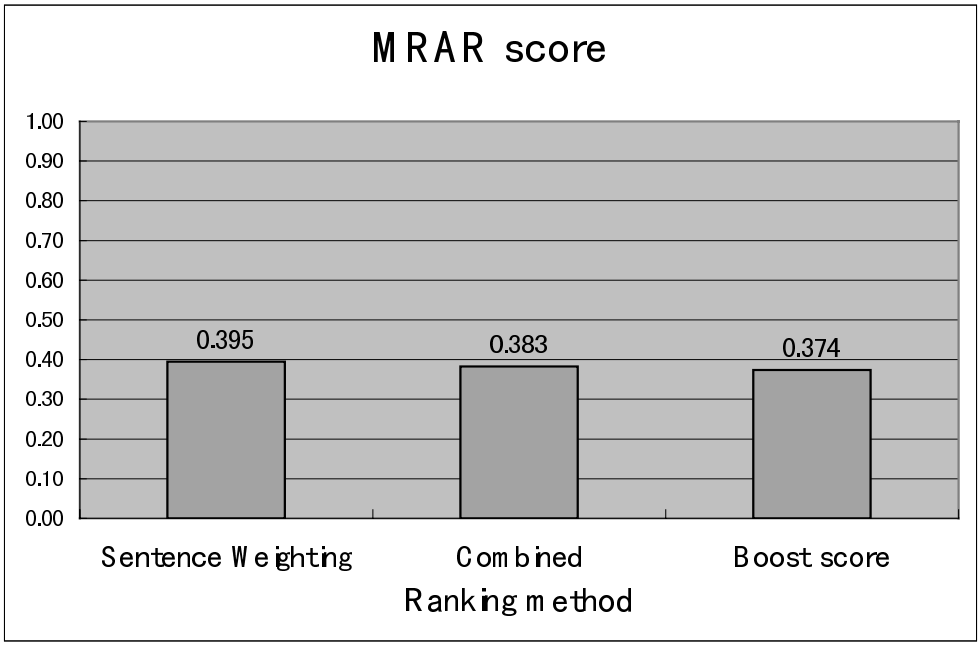


**Figure 4.1** MRAR score

Figure 4.1 illustrates the Mean Reciprocal Answer Rank (MRAR) score of the three ranking methods. The MRAR scores are designed to focus on the top 5 ranks. Theoretically, the score ranges from 1 to zero, with the best being score 1. As can be seen, the system shows the system being tested to be similarly effective through all ranking methods, although the ranking assigned according to sentence weighting slightly exceeds the other two.

### 4.1.2 Detection rate in top 20

Figure 4.2 provides the detection rates in the top 5, 10 and 20 of the three ranking methods. A detection rate was calculated from the number of the query, whose descriptive phrase was ranked in top 5, 10, 20, divided by the number of the valid query (i.e. 57).

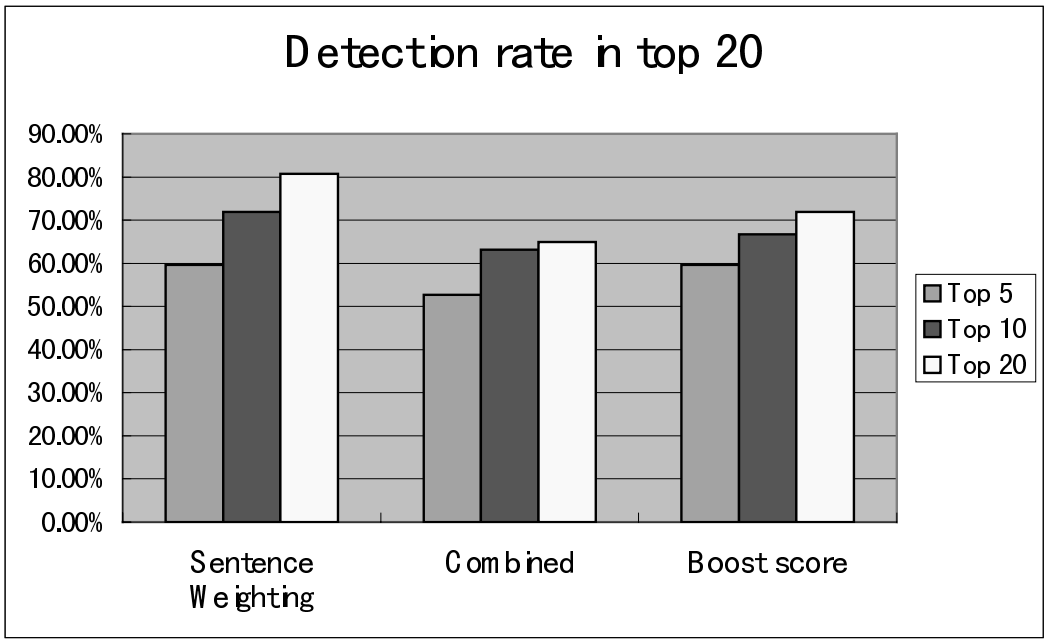


**Figure 4.2** Detection rate in top 20

As indicated in the previous MRAR score, the best performance in all groupings (top 5%, top 10 % and top 20%) was obtained using the sentence weighting method. Moreover, the detection rate of the sentence weighting method shows a greater increment in percentage retrieval between successive groups than is the case with results obtained by the other two methods.

Note that there were two queries whose collection of sentences was less than 20, as shown in Table 4.1 with brackets. Their descriptive phrases were ranked in the top 20 regardless of the ranking method used.

| | Top 5 | Top 10 | Top 20 |
|---|---|---|---|
| Sentence Weighting | 34 (1) | 41 (1) | 46 (2) |
| Combined | 30 (1) | 36 (1) | 37 (2) |
| Boost score | 34 (1) | 38 (1) | 41 (2) |

**Table 4.1** Detected queries in top 20 (N = 57)

## 4.2 Patterns

So far, we have seen the effectiveness of the system as assessed by the two measures discussed above. Now, attention will be paid to the effectiveness of the pattern matching. In this section, the term 'match' means that a sentence is matched to any pattern: it does not always mean that the pattern succeeded in detecting a descriptive phrase. On the other hand, the term 'detect' means that a sentence is matched to a pattern, and the pattern was successful in detecting a descriptive phrase.

### 4.2.1 Overall

The pattern matching for detection DPs was conducted on the valid 57 queries. Of the 57 queries, 48 queries had at least one sentence that matched one of the patterns. The number of sentences that matched a pattern was 482. Of these 482 sentences, the system succeeded in detecting at least one DP in 172. These 172 sentences came from 41 of the 48 queries.

From these facts, the detection rate of the system by the pattern matching was calculated to be 71.93% against the 57 valid queries, and 85.42%, against the matched 48 queries.

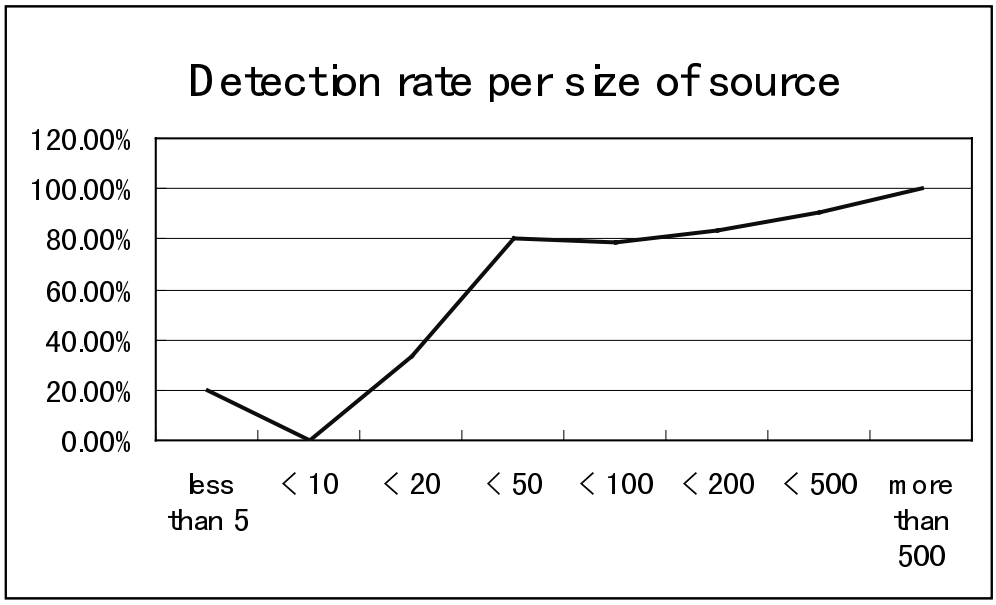


**Figure 4.3** Detection rate per size of source

Figure 4.3 shows the detection rates of the system depending on the size of the collection of sentences in a query. As can be seen from the data, the detection rates in cases where the collection had more than 20 sentences were almost all above 80%. Moreover, as shown in table 4.2, 20 sentences is not an unrealistic size for a collection, as demonstrated by the fact that 46 of the 57 queries had more than 20 sentences: in fact the average size was 170 sentences.

| Size of source (Sentence) | Valid query | Match | Detect | Rate |
|---|---|---|---|---|
| Less than 5 | 5 | 1 | 1 | 20.00% |
| < 10 | 3 | 0 | 0 | 0.00% |
| < 20 | 3 | 1 | 1 | 33.33% |
| < 50 | 10 | 10 | 8 | 80.00% |
| < 100 | 14 | 14 | 11 | 78.57% |
| < 200 | 6 | 6 | 5 | 83.33% |
| < 500 | 11 | 11 | 10 | 90.91% |
| More than 500 | 5 | 5 | 5 | 100.00% |
| Total | 57 | 48 | 41 | |

**Table 4.2** Detection rate per collection size

### 4.2.2 Coverage

Table 4.3 shows the coverage of the patterns. Coverage of a pattern was calculated by the ratio of the detected query in the 57 valid queries. The coverage indicates how widely a pattern can be applied to detection.

| Pattern | Detect (query) | Coverage |
|---|---|---|
| appositive | 29 | 50.88% |
| such as | 18 | 31.58% |
| and other | 20 | 35.09% |
| including | 13 | 22.81% |
| such x as | 10 | 17.54% |
| abbreviation | 8 | 14.04% |
| is a | 7 | 12.28% |
| especially | 1 | 1.75% |
| or other | 1 | 1.75% |

**Table 4.3** Coverage of pattern (N = 57)

As can be seen, the pattern 'appositive' succeeded in detecting a descriptive phrase in more than half of all the valid queries. The patterns 'such as' and 'and other' also show high coverage, followed by 'including.'

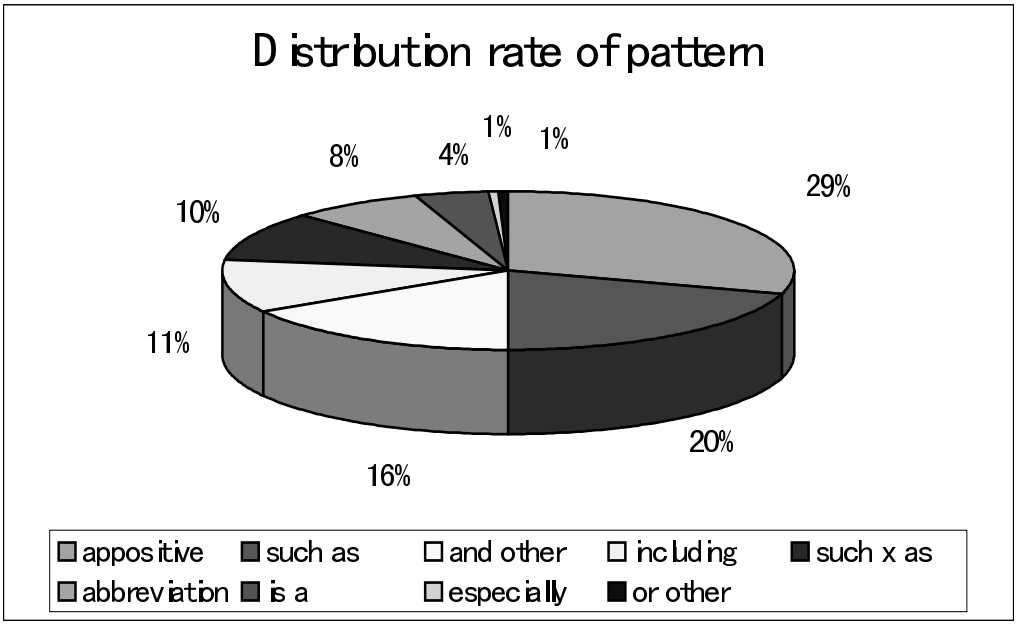


**Figure 4.4** Distribution rate of pattern

Figure 4.4 illustrates the distribution rate in the 172 sentences detected. The three patterns, mentioned in the previous paragraph, account for approximately 65% of all the sentences. See Table 4.4 for the real number of the distribution.

| Pattern | Detect (sentence) | Rate |
|---|---|---|
| Appositive | 51 | 29.65% |
| Such as | 35 | 20.35% |
| and other | 28 | 16.28% |
| Including | 19 | 11.05% |
| Such x as | 17 | 9.88% |
| Abbreviation | 13 | 7.56% |
| is a | 7 | 4.07% |
| Especially | 1 | 0.58% |
| or other | 1 | 0.58% |

**Table 4.4** Distribution rate in matched172 sentences

### 4.2.3 Accuracy

The accuracy of the patterns was determined by the number of detection divided by the number of matches. Table 4.5 shows the levels of accuracy at both query level and sentence level. Of the two levels, the values at the sentence level should probably be regarded as better measures of the accuracy of the patterns.

| Pattern | Match (query) | Detect (query) | Accuracy | Match (sentence) | Detect (sentence) | Accuracy |
|---|---|---|---|---|---|---|
| Appositive | 37 | 29 | 78.38% | 116 | 51 | 43.97% |
| Such as | 34 | 18 | 52.94% | 79 | 35 | 44.30% |
| and other | 28 | 20 | 71.43% | 57 | 28 | 49.12% |
| Including | 29 | 13 | 44.83% | 71 | 19 | 26.76% |
| Such x as | 25 | 10 | 40.00% | 41 | 17 | 41.46% |
| Abbreviation | 24 | 8 | 33.33% | 62 | 13 | 20.97% |
| is a | 12 | 7 | 58.33% | 16 | 7 | 43.75% |
| Especially | 8 | 1 | 12.50% | 8 | 1 | 12.50% |
| or other | 5 | 1 | 20.00% | 5 | 1 | 20.00% |

**Table 4.5** Accuracy of pattern

As can be seen, 'appositive' and 'and other' performed with exceptional accuracy at the query level, which indicates their reliability as detection clues. Moreover, 'is a' also showed high accuracy despite its few matches. Similar results can be seen at sentence level. Five patterns that had over 40% accuracy at the sentence level, might be regarded as the very reliable detection clues.

# CHAPTER 5
# DISCUSSION

Significant points arising from the results of the experiments presented in the previous chapter will be discussed in this chapter. The overall findings will then be presented in the conclusion of this work, along with a summary of each of the chapters in this thesis. Finally, recommendations will be made for future work.

## 5.1 Points for discussion

In this section, we will focus on three significant findings from the experiments: the performance of the pattern matching approach for detection, correlation of ranking method with the system's performance, and the validity of descriptive phrases.

### 5.1.1 Pattern matching for detecting descriptive phrases

The DPD system employed a pattern matching approach in its detection. The patterns were quite simple syntactically and did not require extensive linguistic knowledge or additional supporting software. Instead, the system used a query and several text fragments as the cues. By means of this simple technique, it attempted to detect descriptive phrases from a free text database. Despite the simplicity of the technique, as presented in the previous chapter, the overall performance of the pattern matching was 71.93% against the valid 57 queries, and 85.42% against the 48 matched queries. Thus, it is fair to say that the pattern matching performed well in detecting the descriptive phrases from unstructured texts. In addition, since the database consisted of news articles, these figures may indicate that this pattern matching approach can be successfully applied independently of topic domain: something which has not been achieved by most AI or knowledge-based techniques.

This significance of the pattern matching was also enhanced by the more than 80% detection rate achieved when the collection of queries is bigger than 20 sentences. Moreover, as mentioned in the previous chapter, the figure of more than 20 sentences as the source of a query is not unrealistic, since over 80% of all the valid queries had more than 20 sentences. However, due to the limitations of time we were unable to collect a large number of queries for the experiment. Therefore, one may assume that the high detection rate would be decreased if more queries had been used. Nevertheless, from the data presented in Figure 4.3, it is fair to say that the performance of the pattern matching improves as the size of source is increased.

As for the patterns, the 'appositive' pattern proved to be the most successful in terms of both coverage and accuracy. It succeeded in detecting a descriptive phrase in more than half of all the valid queries, and had an accuracy of over 75%. This indicates that the appositive phrases of a query can be used as the descriptive phrase, and are relatively easy to detect without error. Other patterns such as 'and other' and 'such as' also showed high coverage and accuracy. Apart from these three well performed patterns, the 'such as' and 'is a' showed high accuracy despite the narrow coverage. From these observations, one can see that each pattern has its own tendencies in terms of accuracy and coverage.

In addition, it is important to use patterns together in order to detect a descriptive phrase, hence more detailed observation of these patterns will be required. Also, where a large free-text database is available, the accuracy of a pattern should be determined prior to the coverage, since the minimum requirement of the system is to find a single correct phrase.

### 5.1.2 Ranking method

The DPD system employed three methods of ranking. One was based on sentence weighting score, which was derived from IDF score of terms in a sentence. One was based on boost score, which was determined by the types of pattern matched. Another was based on a combination of the other two scores. The two measures, which provided an indication of the effectiveness of the overall system, that is, the

MRAR score and detection rate in top 20, were strongly affected by the performance of the ranking methods.

In both measures, the system showed better performance with the sentence weighting ranking, than with the others. It succeeded in ranking more than one descriptive phrase of 60% of the valid queries, into the top 5, 70% into the top 10, and 80% into the top 20. These high sentence weighting rankings justified our assumption that the term weighting for ranking descriptive phrases would be significant. Since the sentence weighting ranking method was not supported by the information from the pattern matching at all, one may say that the sentence that consists of many 'significant' terms tends to have a more descriptive phrase than that of less significant terms.

On the other hand, the boost score ranking was strongly determined by the results of pattern matching. The score of each sentence was given depending on the type of pattern matched. This pattern-biased ranking method also showed reasonable results, with the support of the high performance of the pattern matching.

Despite the high performance with the previous two methods, the performance with the combined ranking was unexpected. This could be due to the way in which the scores were combined, which was simply to multiply the two together. However, there were cases in which the weighting score of a sentence that matched any pattern, was negative. In those cases, this score, multiplied by the boost score generated a combined score which was also negative number. Because all sentences were arranged in order of decreasing score, those sentences that were given a negative sentence weighting score merely ranked high. This may explain the worse performance of the combined ranking method.

### 5.1.3 What can be a descriptive phrase?

Up to now, we have focused on findings from the results of the experiment. All those results were based on judgements of whether a detected phrase is descriptive or not. Although those judgement were made by the author, it was not always a straightforward task. Take the query 'START', for instance. The DPD system found

two phrases for this query: ’ the Strategic Arms Reduction Talks’ and ’issues.’ Although the latter phrase is not wrong, one may regard it as too general and hence not valid as a descriptive phrase. Another example is ’Nike.’ The detected phrases were ’ the nation’s top sneaker firm ’ and ’companies.’ Both could be the descriptive phrases of the query. However, it seems that the former describes the query better than the latter. Note that some detected phrases are plurals because of the structure of pattern matching.

From these observations, it is fair to say that

- A phrase, even if correct, may not be suitable as a descriptive phrase for a variety of reasons (e.g., generality).
- Valid phrases may contain either poor descriptions or rich descriptions.
- A framework is needed which in order to determine the validity of descriptive phrases.

However, we concluded that it would be premature to discuss such a framework using only the results from this experiment: further research is clearly needed on this point. Instead, we attempted to identify several factors that seem to influence the validity of descriptive phrases. These factors are discussed below.

**Information source**

Given that the descriptive phrases are detected from a free-text database, the phrases are certainly influenced by some attributions of the information source. Even the existence of a query depends on the information source. In fact, some of the queries used in our experiment did not exist in the database (LA Times in 1989-90), due to its being out of date.

In addition, the aim of a document and of its author will strongly affect the quality of descriptive phrases. For example, it will be easier to detect descriptive phrases in an explanatory document such as a news article or a journal than in other types of document, since those documents are intended to inform on events or facts to people who are not familiar with them.

**Popularity of query**

We found that very popular queries tend to have metaphorical or indirect descriptions. For example, ' the worst thing to happen in the 20th Century' was retrieved as a description for AIDS and ' the genetic blueprint of life' for DNA. Less popular queries in turn had the tendency to be described in detail. For example, Noboru Takeshita was 'the old-guard politician who resigned as prime minister because of his links to the Recruit Co.'. Again, the popularity may also be influenced by the date, domain, and other cultural factors of the information source. Therefore, a wide range of dates and information source domains will be needed, in addition to a range of sizes, in order to obtain useful descriptive phrases.

**Different attribution of query**

Some queries had two (or more) attributions to be described. For example, NATO is the acronym of at least two organisations: 'National Association of Theater Owners' and 'North Atlantic Treaty Organization.' In this case, both descriptions must be detected since they are different in essence. Another case may be that the attribution of a query has changed. For example, the age of person, the population of a country, the length of a river and the like. In this case, the descriptive phrases may have different information despite referring to the same object.

**User**

As is well-known in the issues of relevance judgement in information retrieval, the validity of descriptive phrases may also be objective and determined by an individual perspective.

As can be seen, several factors may affect the validity of descriptive phrases. Descriptions of the various attributes of a query will be especially important, since such descriptions could allow different users to specify their needs more precisely. The DPD system could then succeed in detecting the different descriptions of a query, which would be a useful novelty of this system.

Up to this point, we have discussed some of the remarkable features of this work. The next section will summarise the work and provide conclusions, followed by recommendations.

## 5.2 Conclusion

The aim of this work was to evaluate the effectiveness of the descriptive phrase detection (DPD) system. The evaluation was mainly divided into two part: the first part considered the effectiveness of the system, while the second part dealt with the pattern matching approach for detecting the descriptive phrases. These detected phrases were expected to answer several specific types of questions, such as 'What is X?', 'Who is X?', 'What does X mean?' or 'What job does X do?' Questions of this type have not been answered directly by existing information retrieval (IR) systems, and hence the DPD system was expected to contribute toward satisfying the need for another type of information seeking behaviour, which could be provided through the form of question answering.

Chapter 1 provided the research background, the aim and scope of this work motivated by the background, and a brief description of methodology. As the social context of the research background, the growth of digitised information on the web and the needs of another form of search tools were mentioned. As the technical context, the limitation of existing IR systems and the significance of question answering (QA) systems as a kind of search tools were examined. It was noted that few attempts have been made to build a QA system using the techniques in IR and other related fields, despite their use for testing AI and Knowledge based techniques. From these observations, the aim of the work and the objectives of the system were described: i.e., that it should be domain independent and use free text as its information source.

Chapter 2 reviewed the literature related to this work. The aim and scope of the previous chapter, and the methodology in the next chapter were based on the findings from the literature review. This chapter was divided into three parts according to topic. The first focussed on question answering systems in general, and considered the use of DPD in such systems. The findings from the first part were that

- the early QA systems were often employed by AI or knowledge-based related techniques, and were usually restricted in domain,
- there has recently been growing interest in the use of QA systems as another form of IR, due to the limitation of existing IR systems.

The second part of Chapter 2 focused on techniques related to the main processes of the DPD system: detection and ranking. In considering detection methods, the pattern matching approach was reviewed and compared with parsing. The advantages were seen as being simple algorithm and fast processing, while the main disadvantage in comparison to parsing was the lower accuracy of pattern matching. Here, the significance of Heast's pattern matching was especially emphasised because of its flexibility as well as its efficiency. Inverse document frequency (IDF) was focussed on as the means of ranking, and its uses in IR were examined. As a result, the information on the efficiency of this method was found in studies of IR and automatic text summarisation. The last part of Chapter 2 was concerned with evaluation issues.

The architecture of the DPD system, experimental resources, and evaluation schemes were presented in chapter 3. In the architecture, a detailed description was provided of the patterns employed. These were based on Heast's technique, and the use of appositive phrases was shown with illustrative examples. Also, three ranking methods based on IDF and boost score were described. Emphasis was placed on achieving domain independence by those methods. The nature of the query and information sources used by the system were then explained. The use of a free-text database as the information source was one of the objectives of the system. As mentioned before, the evaluation scheme was divided into two parts. The system's overall effectiveness was evaluated by MRAR score and detection rates in the top 5, 10, 20. In order to evaluate the effectiveness of the detection method, the patterns were evaluated separately in terms of accuracy and coverage.

Chapter 4 presented the results from the experiment through the evaluation scheme. The system proved to be effective, with a 60% detection rate in the top 5, and with rates of 70% and 80% in the top 10 and 20 respectively. The best performance was given by sentence weighting ranking. As for overall pattern effectiveness, the patterns succeeded in detecting a descriptive phrase in more than 70% of the queries. This figure was increased to nearly 80% in queries with more than 20 sentences. The patterns which were most effective were 'appositive', 'such as' and 'and other'. These performed significantly better than the other patterns, in terms

of accuracy and coverage. The patterns 'such x as' and 'is a' also showed high accuracy despite their narrow coverage.

Some significant points arising from this work were discussed in the beginning of this chapter. At the same time, limitations were identified, and recommendations were made for improvements to the system. These points were then summarised in the section that followed.

In conclusion, the following points can be made.

- The DPD system has the potential to provide the descriptive information of a word/noun phrase.
- Simple pattern matching can detect descriptive phrases at a high rate.
- Detection rates are influenced by the size of a collection of related sentences.
- Each pattern has its own tendency in terms of accuracy and coverage.
- Appositives are particularly useful for detecting descriptive phrases.
- Simple IDF-based term weighting can be useful for ranking descriptive phrases.
- The quality of a descriptive phrase may be influenced by the information source, and by some attributes of queries and users.

## 5.3 Future work

Suggestions for future work will focus on the improvement of detection process, managing descriptive phrases, and other aspects of implementation.

### 5.3.1 Improvement of detection

We will try to find more simple text fragments or other such clues for detection. During this experiment, we found that 'known as' seems to be useful, for example in '… Telmex, also *known as* Telefonos de Mexico …'

Hearst (1992) suggests a way to discover new patterns as follows:

(1) Decide on a lexical relation of interest, e.g., 'group/member.'

(2) Gather a list of terms for which this relation is known to hold, e.g., 'England-country.'

(3) Find places in the corpus where these expressions occur syntactically near one another and record the environment.
(4) Find the commonalties among these environments and hypothesise that common ones yield patterns that indicate the lexical relation of interest (see (1)).
(5) Once a new pattern has been positively identified, use it to gather more instances of the target relation.

In order to improve detection algorithms, more sophisticated statistical and phrase techniques could be used, including exploiting simple IE and NLP techniques, such as named entity recognisers, and co-reference resolving. Co-reference resolving is a means of dealing with anaphora in texts (see Lappin and Leass, 1994), as in the following example:

*'Bill Evans* died in 1961. *He* was one of the most famous jazz pianists.'

The 'he' refers to 'Bill Evans', and consequently 'one of the most …' can be detected as a description of 'Bill Evans.' Pronouns such as 'he', 'she', 'they' will be the cues for this technique. Although some syntactic analysis may be required for a complex text, the degree of analysis can be decreased by using this method in combination with the text fragments used in our project.

### 5.3.2 Managing descriptive phrases

As the size of database is increased, a number of issues will arise from the fact that too many descriptions will be returned. For example, Tony Blair may be described as a politician, a father, a prime minister and many other things. There may be two people called Tony Blair, just as there are two NATOs. How should the system choose the descriptive phrase? Similarly, in one instance in the course of this project, a query was described in many ways all of which had similar meanings. 'AIDS' was referred to variously as 'a life-threatening disease', 'dreadful disease', and 'a virus infection.' Although the first two phrases look very similar, they are quite different from the last one. From these observations, a process of *foldering*, i.e., identifying a representative phrase from the group of similar descriptions, may be required.

This could be achieved by spotting *co-occurrence* of certain pairs of terms within a collection of descriptive phrases. Doyle (1962) explains co-occurrence as follows: 'If authors writing on special topics use certain words with unusual frequency, a consequence of this should be unusual co-occurrence of certain pairs of words within the text of the same documents.' Such information based on term frequency may be used for characterising and grouping similar descriptions.

Currently the system only ranks sentences, therefore, the process of extracting descriptive phrases and displaying them to the user should be developed. Due to the detection algorithm, the descriptive phrases are sometimes plurals. Thus, if appropriate, a process that alter a phrase to singular form may also be required.

### 5.3.3 Others

In this work, we were interested in detecting descriptive phrases based on a query, especially a noun word or phrase such as a name. This will help to provide answers to questions such as 'Who is Tony Blair?' -British prime minister. Alternatively, one may want to extract some information based on a piece of descriptive phrase. For example, 'Who is the British prime minister?' -Tony Blair. It is interesting to examine how well the simple patterns work for the latter case.

Recently, a number of researchers have focused on the World Wide Web (WWW) as an information source for a system (e.g., Katz (1997), Craven et al. (1998), Gaizauskas and Robertson (1997)). Although a number of technical problems concerning the use of the WWW were identified by Baeza-Yates and Ribeiro-Nero (1999), the one that relates to our system is that regarding the quality of documents on the Web. As mentioned before, authors on the Web are not necessarily professionals, so the quality of documents is far more variable than that of news articles or academic papers. Therefore the WWW represents a more genuine free-text information source than the database used here, so we are interested in seeing how the DPD system would perform in such an environment.

# REFERENCES


Ahlswede, T. & Evens, M. (1988). "Parsing vs. Text Processing in the Analysis of Dictionary Definitions", In: *Proceedings of the 26th Annual Meeting of the Association for Computational Linguistics*, 217-224. Association for Computational Linguistics.

Alshawi, H. (1987). "Processing Dictionary Definitions with Phrasal Pattern Hierarchies", *Computational Linguistics*, **13**(3-4), 195-202.

Baeza-Yates, R. & Ribeiro-Neto, B. (1999). *Modern Information Retrieval*. New York: ACM Press.

Barr, A. (1982). *Artificial intelligence: cognition as computation*. Department of Computer Science, Stanford University.

Belkin, N. J. (1981). "Ineffiable concepts in information retrieval". In: Sparck Jones, K. (editor). *Information Retrieval Experiment*, 44-58. London: Butterworths.

Belkin, N. J. & Vickery, A. (1985). *Interaction in Information Systems: A review of research from document retrieval to knowledge-based systems*. London: British Library Board.

Black, W. J., Gilardoni, L., Rinaldi, F. & Dressel, R. (1997). "Intergrated text categorisation and information extraction using pattern matching and linguistic processing", In: *Proceedings of the Computer-Assisted Information Searching on Internet (RIAO 97)*, 321-335. Montreal, Canada. McGill University.

Charniak, E., Carroll, G., Adcock, J., Cassandra, A., Gotoh, Y., Katz, J., Littman, M. & McCann, J. (1996). "Taggers for parsers", *Artificial Intelligence*, **85**, 45-57.

Coates-Stephens, S. (1993). "The Analysis and Acquisition of Proper Names for the Understanding of Free Text", *Computers and the Humanities*, **26**, 441-456.

Craven, M., DiPasquo, D., Freitag, D., McCallum, A., Mitchell, T., Nigam, K. & Slattery, S. (1998). "Learning to Extract Symbolic Knowledge from the World Wide Web", In: *Proceedings of 15th National Conference on Artificial Intelligence (AAAI-98),* American Association for Artificial Intelligence.

Cuadra, C. A. & Katter, R. V. (1967). "Opening the black box of relevance", *Journal of Documentation*. **23**, 291-303.

Cutting, D., Kupiec, J., Pedersen, J. & Sibun, P. (1992). "A Practical Part-of-Speech Tagger", In: *Proceedings of the 3rd conference on Applied Natural Language Processing*, 133-140. Trento, Italy.

Dillon, M. & Gray, A. (1983). "Fully Automatic Syntax-based Indexing", *Journal of the ASIS*, **34**(2), 99-108.

Doyle, L. B. (1962). "Indexing and Abstracting by Association. Part 1." SP-718/001/00. Santa Monica, CA: System Development Corporation. Also, In: Sparck Jones, K. & Willett, P. (editors.). (1998). *Readings in Information Retrieval*, 25-38. San Francisco: Morgan kaufmann.

Earl, L. L. (1970). "Experiments in automatic abstracting and indexing", *Information Storage and Retrieval*, **6**(4)**,** 313-334.

Ellis, D. (1996). "The Dilemma of Measurement in Information Retrieval Research", *Journal of the American Society for Information Science*. **47**(1), 23-36.

Fellbaum, C. (editor). (1998). *WordNet: An Electronic Lexical Database*. Cambridge: MIT Press.

Gaizauskas, R. & Robertson, A. M. (1997). "Coupling Information Retrieval and Information Extraction: A New Text Technology for Gathering Information from the Web", In: *Computer-Assisted Information Searching on Internet (RIAO 97)*, 356-370. Montreal, Canada.

Gaizauskas, R. & Wilks, Y. (1998). "Information Extraction: Beyond Document Retrieval", *Journal of Documentation,* **54**(1)**,** 70-105.

Gershman, A. (1982). "A Framework for Conceptual Analyzers". In: Lehnert, W. G. & Ringle, M. H. (editors), *Strategies for Natural Language Processing*, 177-197. Hillsdale: Lawrence Erlbaum.

Green, B., Wolf, A., Chomsky, C. & Laughery, K. (1961). "BASEBALL: An Automatic Question Answerer", In: *Proceedings of the Western Joint Computer Conference 19,* 219-224. American Federation of Information Processing Societies. Also, In: Grosz, B. J., Sparck Jones, K. & Webber, B. L. (editors). (1986). *Readings in Natural Language Processing*, 545-549. California: Morgan Kaufmann.

Grosz, B. J., Sparck Jones, K. & Webber, B. L. (editors). (1986). *Readings in Natural Language Processing*. California: Morgan Kaufmann.

Haas, S. W. (1996). "Natural Language Processing: Toward Large-Scale, Robust Systems", *Annual Review of Information Science and Technology*, **31**, 83-119.

Harter, S. P. (1996). "Variations in Relevance Assessments and the Measurement of Retrieval Effectiveness", *Journal of the American Society for Information Science*, **47**(1), 37-49.

Harter, S. P. & Hert, C. A. (1997). "Evaluation of Information Retrieval Systems: Approaches, Issues, and Methods", *Annual Review of Information Science and Technology*, **32**, 3-94.

Hayes, P. J., Knecht, L. E. & Cellio, M. J. (1988). "A news story categorisation system", In: *Proceedings of the Second Conference on Applied Natural Language Processing, Association for Computational Linguistics*, 9-17. San Francisco: Morgan Kaufmann.

Hearst, M. A. (1992). "Automatic Acquisition of Hyponyms from Large Text Corpora", In: *Proceedings of The 14th International Conference on Computational Linguistics*, 539-545. Nantes: ACL.

Hearst, M. A. (1998). "Automated Discovery of WordNet Relations". In: Fellbaum, C. (editor), *WordNet: An Electronic Lexical Database and Some of its Applications*, 131-151. Massachusetts: MIT Press.

Jacobs, P. S., Krupka, G. R. & Rau, L. F. (1991). "Lexico-Semantic Pattern Matching as a Companion to parsing in Text Understanding", In: *Proceedings of the fourth DARPA Workshop on Speech and Natural Language*, 337-342. California: Pacific Grove.

Jacobs, P. S. & Rau, L. F. (1990). "SCISOR: Extracting Information from On-line News", *Communications of the ACM*, **33**(11), 88-97.

Johnson, F. C., Paice, C. D., Black, W. J. & Neal, A. P. (1993). "The application of linguistic processing to automatic abstract generation", *Journal of Documentation and Text Management*, **1**(3), 215-241.

Katz, B. (1997). "Annotating the World Wide Web using Natural Language", In: *Proceedings of the Computer-Assisted Information Searching on Internet (RIAO 97)*, 136-155. Montreal, Canada.: McGill University.

Kay, M. & Sparck Jones, K. (1971). "Automated language processing", *Annual review of information science and technology*, **6**, 141-166.

Kearsley, G. P. (1976). "Questions and question asking in verbal disourse", *Journal of Psychological Research*, **5**, 355-375.

Kim, J.-T. & Moldovan, D. I. (1995). "Acquisition of Linguistic Patterns for Knowledge-Based Information Extraction", *IEEE Transactions on Knowledge and Data Engineering*, **7**(5), 713-724.

Kupiec, J. (1993). "MURAX: A Robust Linguistic Approach For Question Answering Using An On-Line Encyclopedia", In: *Proceedings of 16th Annual ACM SIGIR conference on Research and Development in Information Retrieve*, 181-190. Pittsgurgh: ACM.

Lappin, S. & Leass, H. (1994). "An algorithm for prnominal anaphora resolution", *Computational Linguistics*, **29**(4), 535-561.

Mani, I., House, D., Klein, G., Hirschman, L., Obrst, L., Firmin, T., Chrzanowski, M. & Sundheim, B. (to be print). *The TIPSTER SUMMAC Text Summarization Evaluation: Final Report*. Maryland: The MITRE Corporation.

Mani, I. & Maybury, M. T. (editors). (1999). *Advances in Automatic Text Summarization*. Cambridge: MIT Press.

Miller, G. A. (1995). "WordNet: A lexical database for English", *Communications of the ACM*, **38**(11), 39-41.

Nakamura, J. & Nagao, M. (1988). "Extraction of Semantic Information from an Ordinary English Dictionary and its Evaluation", In: *Proceedings of the 12th International Conference on Computational Linguistics*, 459-464.

Paice, C. D. (1981). "The automatic generation of literature abstracts: an approach based on the identification of self-indicating phrases". In: Oddy, R. N. (editors), *Information retrieval research*, 172-191. London: Butterworths.

Paice, C. D. (1990). "Constructing Literature Abstracts by Computer: Techniques and Prospects", *Information Processing & Management*, **26**(1), 171-186.

Poesio, M. & Vieira, R. (1998). "A Corpus-based Investigation of Definite Description Use", *Computational Linguistics*, **24**(2), 183-216.

Radev, D. R. (1998). "Learning correlations between linguistic indicators and semantic constraints: Reuse of context-dependent descriptions of entities", In: *Proceedings of the Joint 17th International Conference on Computational Linguistics 36th Annual Meeting of the Association for Computational Linguistics (COLING-ACL'98)*, Montreal, Canada.

Rau, L. F., Jocobs, P. S. & Zernik, U. (1989). "Information Extraction and Text Summarization Using Linguistic Knowledge Acquisition", *Information Processing and Management*, **25**(4), 419-428.

Salton, G. (1966). "Automatic Phrase Matching". In: Hays, D. G. (editor), *Readings in Computational Linguistics*, 169-188. New York: American Elsevier.

Salton, G. (1989). *Automatic Text Processing: The Transformation, Analysis, and Retrieval of information by Computer*. Massachusetts: Addison-Wesley.

Salton, G. & Buckley, C. (1988). "Term-weighting approaches in automatic text retrieval", *Information Processing and Management*, **24**, 513-523.

Salton, G. & McGill, M. J. (1983). *Introduction to Modern Information Retrieval*. New York: McGraw-Hill.

Sanderson, M. (1998). "Accurate user directed sumarization from existing tools", In: *Proceedings of the 7th International Conference on Information and Knowledge Management (CIKM 98)*, 45-51, Bethesda.

Saracevic, T. (1975). "Relevance: A review of and a framework for thinking on the notion in information science", *Journal of the American Society for Information Science*, **26**, 321-343. Also, In: Sparck Jones, K. & Willett, P. (editors). (1998). *Readings in Information Retrieval*, 143-165. San Francisco: Morgan Kaufmann.

Saracevic, T. (1995). "Evaluation of Evaluation in Information Retrieval", In: Fox, E. A., Ingwersen, P. & Fidel, R. (editors), *Proceedings of the 18th Annural International ACM SIGIR Conference on Research and Development in Information Retrieval*, 138-146. Seattle, Wahington: ACM Press.

Smith, L. C. (1980). "Artificial intelligence: applications in information systems", *Annual review of information science and technology*, **15**, 67-105.

Sparck Jones, K. & Willett, P. (editors). (1997). *Readings in Information Retrieval*. San Francisco: Morgan Kaufmann.

Swanson, D. R. (1988). "Historical Note: Information Retrieval and the Future of an Illusion", *Journal of the American Society for Information Science*, **39**, 92-98.

Tague-Sutcliffe, J. M. (1996). "Some Perspectives on the Evaluation of Information Retrieval System", *Journal of the American Society for Information Science*, **47**(1), 1-3.

Tombros, A. & Sanderson, M. (1998). "Advantages of Query Biased Summaries in Information Retrieval", In: *Proceedings of 21st Annual ACM SIGIR conference on Research and Development in Information Retrieve*, 2-10, 1998, Melbourne: ACM.

van Rijsbergen, C. J. (1979). *Inormation Retrieval*. London:Butterworths.

Vickery, B. C. & Vickery, A. (1987). *Information Science in Theory and Practice*. London: Butterworths.

Wendlandt, E. B. L. & Driscoll, J. R. (1991). "Incorporating a Semantic Analysis into a Document Retrieval Strategy", In: *Proceedings of 14th Annual ACM SIGIR conference on Research and Development in Information Retrieve*, 270-279. Illinois: ACM.

Wilks, Y. A., Fass, D. C., Guo, C. M., McDonald, J. E., Plate, T. & Slator, B. M. (1990). "Providing machine tractable dictionary tools", *Machine Translation*, **5**(2), 99-151.

WWW001. (1998). *Text Retrieval Conference (TREC) Home Page*. [http://trec.nist.gov/]. Site visited at: 22/02/99.

WWW002. (1998). *Question Answering Track at TREC-8*. [http://www.research.att.com/~singhal/qa-track.html]. Site visited at: 22/02/99.

Yu, C. T. & Salton, G. (1976). "Precision weighing - an effective automatic indexing method", *Journal of the ACM*, **23**(1), 76-88.

# APPENDIX:
# SAMPLE DESCRIPTIVE PHRASES

---

**Query**: Aerosmith
**Descriptive phrase (DP)**: power rock precursors

**Query**: Agent Orange
**DP**: chemicals deformed the land and people; a defoliant containing dioxin

**Query**: AIDS
**DP**: a life-threatening disease; the worst thing to happen in the 20th Century; a human disaster; a virus infection; acquired immune deficiency syndrome

**Query**: Bob Dylan
**DP**: giants; artists;

**Query**: Cold War
**DP**: issues; the arms race and a policy of military confrontation depleted so much of their restticted resources; an era whose onset leaves world powers with the luxury of joining in collective action without having to worry about guarding their own back doors

**Query**: cookies
**DP**: International holiday backed goods; snack foods; date concoctions; luxuries

**Query**: Diane Sawyer
**DP**: reporters; media and fashion stars

**Query**: DNA

**DP**: deoxyribonucleic acid; the repository of genetic information; the genetic blueprint of life; the master regulator of the cell

**Query**: entreprenues

**DP**: experts; a diverse and fascinating lot whose common trait is the ability to view old problems with new perspectives;

**Query**: Fed

**DP**: Federal Reserve Board; the nation's central bank; the nation's fourth-largest thrift with $23 billion in assets; European central banks

**Query**: Hitachi

**DP**: Japanese semiconductor giants; large Japanese electronics companies; industrial powerhouses

**Query**: IBM

**DP**: the world's largest manufacturer and consumer of memory chips; mainframe computers; Technology stocks; the country's largest businesses

**Query**: IRA

**DP**: your tax-deferred retirement program; demand-increasing schemes; Irish Republican Army; the guerrilla group battling to oust Britain from Northern Ireland

**Query**: IRS

**DP**: Internal Revenue Service; company duties;

**Query**: Kabuki

**DP**: performing arts/cultural exhibits

**Query**: Manson

**DP**: 90 factory-direct and off-price outlets; the razor-sliced swastika in his forehead emphasized in blue-black ink

**Query**: Marlboro

**DP**: well-known Philip Morris brands; the biggest-bucks sponsor in Indy car racing; widely know brands

**Query**: Mavericks

**DP**: contending for one of the eight playoff spots in the Western Conference; the team's previous owner(Dallas Mavericks); post-modern coach(Dallas Mavericks)

**Query**: Microsoft

**DP**: The world's largest PC software publishing house; software companies

**Query**: moratorium

**DP**: an extention of shorter and less stringent demolition bans

**Query**: NATO

**DP**: the National Association of Theater Owners; North Atlantic Treaty Organization; international organization

**Query**: Nike

**DP**: companies; the nation's top sneaker firm

**Query**: Nissan

**DP**: companies; the leading Japanese auto makers; Japan's second largest auto maker

**Query**: Noboru Takeshita

**DP**: the sixth Cabinet member; Administration officials; the old-guard politician who resigned as prime minister because of his links to the Recruit Co.

**Query**: Rolling Stones

**DP**: rockers; artists; Anglo or American acts

**Query**: Safeway

**DP**: Major supermarket chains

**Query**: Samurai
**DP**: a warrior class with no wars to fight; a popular product of American Suzuki Motor Corp.

**Query**: Sony
**DP**: the world's major entertainment and computer companies; the Japanese electronics maker that owns CBS Records and Columbia Pictures; the first Japanese company to license Bell Labs' transistor in the 1950s

**Query**: Sosuke Uno
**DP**: the sixth Cabinet member

**Query**: Star Wars
**DP**: strategic programs in fiscal 1991; high-technology weaponry

**Query**: START
**DP**: the Strategic Arms Reduction Talks; issues

**Query**: Sun Microsystems
**DP**: the nation's largest manufacturer of powerful computer workstations; the darling of Silicon Valley; entrepreneurial upstarts

**Query**: tofu
**DP**: bean curd

**Query**: Toshiba
**DP**: Japanese semiconductor giants; the leading Japanese chip maker; big Japanese electronics companies

**Query**: Toshiki Kaifu
**DP**: leaders; a 59-year-old Japanese politician trying to ride the tiger of a surging Japan while balancing the demands of his own constituency and that of an increasingly wary United States

**Query**: Toyota

**DP**: Japan's largest auto maker and the third largest worldwide; industrial powerhouses; Camaro's import competitors

**Query**: UNIX

**DP**: developed operating systems

**Query**: Walkman

**DP**: pioneered products; a series of new consumer electronics; the personal music machine that started us wearing those funny little head phonesis

**Query**: Yamaha

**DP**: the world's major entertainment and computer companies; four-cylinder variations of Japanese sportbikes

**Query**: Yukio Mishima

**DP**: the great Japanese novelist