Université de Montréal

Domain-specific Question Answering System – An Application to the Construction Sector

par

Zhuo Zhang

Département d'informatique et de recherche opérationnelle

Faculté des arts et des sciences

Mémoire présenté à la Faculté des études supérieures

en vue de l'obtention du grade de

Maître en Informatique

Avril, 2003

Université de Montréal

Faculté des études supérieures

Ce mémoire intitulé:

Domain-specific Question Answering System – An Application to the Construction Sector

présenté par:

Zhuo Zhang

a été évalué par un jury composé des personnes suivantes:

Peter Kropf	(président-rapporteur)
Jian-Yun Nie	(directeur de recherche)
Philippe Langlais	(membre du jury)

Mémoire accepté le 3 septembre 2003

Abstract

Question Answering (QA) aims to provide precise answers to user's questions. This task becomes more and more important because of the information explosion. People are not satisfied with the traditional Information Retrieval (IR) systems which identify a large set of documents which may contain an answer. QA needs more refined processing on top of the IR results. Up to now, many approaches have been proposed for general-domain QA. No particular attention has been paid to domain-specific QA. In this thesis, we explore QA in a specific domain — construction sector.

Domain-specific QA implies all the aspects of general-domain QA. Therefore, we implemented a mechanism for general domain QA following the approaches described in the literature. In addition, we also deal with questions related to specialized concepts of the domain, i.e.,to deal with domain-specific QA. This constitutes the main original contribution of this thesis. To extend the existing QA approaches to these questions, we consider categories of concepts in construction as special named entities (NEs) on which one may ask questions. To do this we make use of a thesaurus in the construction sector.

In this thesis, we propose methods to recognize special units in documents and questions: common NEs [1], categories of concepts and compound terms. We also define different search strategies for different types of questions: questions asking for an NE, for a concept of a semantic category, and for a definition.

We have tested our approaches on a set of specialized documents and a set of

[1] A common NE type refers to a type of NE that is domain independant, such as date, person name, organization and so on. A domain-specific NE type refers to a particular semantic category in a specific area.

questions. Our results show that the system performance (i.e., the quality of the answers found by the system.) by using Category-based search strategy is improved by 7.11% in comparison with the baseline approach based on keyword search. By using NE and Definition search strategies, it is improved by 10.35%. Therefore, we can conclude that our domain-specific QA methods are more effective than the baseline method. The final conclusion of this study is that it is beneficial to integrate domain knowledge in specialized QA.

Keywords: Question Answering (QA), Information Retrieval (IR), Information Extraction (IE), Named Entity (NE), Thesaurus, Domain-specific QA.

Résumé

La Question-Réponse (QR) vise à trouver des réponses précises aux questions d'utilisateurs. Cette tâche devient de plus en plus importante étant donné l'explosion d'information actuelle. Les utilisateurs ne sont plus satisfaits des systèmes de recherche d'information (RI) traditionnels qui fournissent un grand ensemble de documents pouvant contenir une réponse. La QR nécessite des traitements plus raffinés sur les résultats de la RI.

Jusqu'à maintenant, beaucoup d'approches ont été proposées pour la QR dans des domaines généraux. Il n'y a pas eu d'étude spécifique pour la QR dans des domaines spécialisés. Dans ce mémoire, nous explorons la QR dans un domaine spécifique, le secteur de la construction.

La QR dans un domaine spécifique implique tous les aspects de la QR dans des domaines généraux. Ainsi, dans notre travail, nous avons aussi implanté un mécanisme pour la QR dans le domaine général, en suivant, les approches décrites dans la littérature. En plus, nous devons aussi traiter des questions reliées aux concepts spécialisés du domaine, c'est-à-dire de traiter la QR du domaine spécifique. C'est sur cet aspect que ce travail apporte une contribution originale. Afin d'étendre les approches de la QR existantes à ce type de question, nous considérons les catégories de concepts en construction comme des entités nommées (EN) spéciales, sur lesquelles les questions peuvent porter. Pour faire cela, nous utilisons un thésaurus dans le domaine de la construction.

Dans ce mémoire, nous proposons des méthodes pour reconnaître des unités spéciales dans des textes et des questions, telles que les EN commune [2], les catégories des concepts et les termes composés. Nous définissons aussi des stratégies de recherche pour différents types de question: questions demandant une EN, un concept d'une catégorie sémantique et une définition.

Nous avons testé nos approches sur un ensemble de documents spécialisés et un ensemble de questions. Nos résultats expérimentaux montrent que la performance du système (i.e., la qualité des réponses trouvées par le systéme) en utilisant la stratégie de recherche basée sur les catégories est améliorée de 7.11%, en comparaison avec une approche de base utilisant seulement des mots clés. En utilisant la recherche basée sur les EN et la définition, la performance est améliorée de 10.35%. Ces résultats montrent clairement que nos approche à QR spécialisée sont plus performante que l'approche de base. La conclusion finale de cette étude est qu'il est bénéfique d'intégrer des connaissances du domaine dans la QR spécialisée.

Mots-clés: Question-Réponse (QR), Recherche d'Information (RI), Thésaurus, Extraction d'Information (EI) , Entités Nommées (EN), QR spécialisée.

[2]Une EN commune est une EN indépendante du domaine, tel que la date, le nom personnel, l'organisation etc. Une EN spécifique du domaine correspond à une catégorie sémantique spécifique au domaine.

Contents

List of Figures

List of Tables

Acknowledgement

First of all, I would like to express my gratitute to Prof. Jian-Yun Nie, my director, for his accurate and patient direction. Without his help, this thesis would not have been available.

I owe my gratitude to Prof. Colin Davidson and Gonzalo Lizarralde of Faculté de l'Aménagement for providing the necessary resources for this work.

I also owe my gratitude to RALI group for providing me with POS tagger, which is an important tool that I used in this work.

This work is part of a research project supported by the Bell-LUB. Iwould like to thank Bell-LUB for providing me with a scholarship.

I would like to thank my classmates Lifang Liu, Qi Zhang, Haiying Xiong, Song Zhang, Xiangqian Ni and Cuihua Lu as well as my friends Fuman Jin and Ngoc Tran Nguyen, for their help and useful discussions.

Finally, thanks to my husband and my parents for their constant support and encouragement.

The completion of this research was made possible thanks to Bell Canada's support through its Bell University Laboratories R & D program.

Chapter 1

Introduction

We live in an information age, where information is crucial for the success of businesses or individuals. The fast growth of information and the development of the World Wide Web (WWW) have given people potential access to more information than they have ever had before. Thus, how to obtain timely and precise information has become an important problem in modern society. More and more, people are not satisfied with retrieving a long list of documents which can potentially contain an answer to their question. They want to obtain a precise answer to it. As a result, Question Answering (QA) has gained a key place among the information access methods. This thesis is about QA. We will develop methods for QA inspired from existing methods. Different from the latter, our QA is carried out in a specific domain — the construction sector. Therefore, we also benefit from the domain knowledge available. An important contribution of this thesis is that we show that the use of domain knowledge for domain-specific QA is highly beneficial for improving the quality of the answers found by the ststem.

In order to better introduce our problems, we will start by describing some general concepts in the following section.

1.1 IR and QA

Traditionally, Information Retrieval (IR) systems are used to find relevant documents in response to a user's query, which specifies the information need of the user. Once the documents are returned, the user needs to read the documents returned by the IR system and find the required information from them. The existing search engines on the Web are examples of IR systems. If the number of relevant documents is small and the user's information requirement is general rather than specific, then this step of extracting information from returned documents may be acceptable. However, if there is a huge amount of documents or if the information requirement is specific, then this step of locating the required information from the returned documents might become unacceptable [GH00].

Currently, although the techniques of IR have much improved, no IR system can understand the meaning of the documents and the user's question. Most IR systems retrieve documents according to keyword matching. It is known that keyword cannot express the full meanings of natural language. An example is given below. The following sentences or phrases contain similar keywords, but they have different meanings [Lin01]:

- He interfaced the design
- He designed the interface
- the designs interface
- the interfaces design

From these sentences or phrases, the current IR systems often extract the same keywords "design" and "interface". Then for a question related to these two keywords, all the documents containing one of these sentences will be returned, and many of them are unrelated to the user's question. Therefore, these limitations in IR make the IR techniques alone unsuitable for certain specific applications [ABH98]. For example, there is no easy way to find an answer to a question such as "who was

the President of the USA in 1999?". Clearly, the user would prefer a person name as an answer such as "Bill Clinton", perhaps with a small amount of context (e.g., a sentence), instead of a ranked list of documents or paragraphs which they must read to discover the answer. In fact, many of the returned documents may not contain a person name at all. It is clear that current IR techniques do not yet enable a system to give precise answers to precise questions. In order to provide precise answers to precise questions, we adopt a new approach that involves IR, Natural Language Process (NLP), Information Extraction (IE), knowledge representation and reasoning techniques together. This is what Question Answering is about.

Question answering aims to return information that directly answers the user's question. The earliest QA system was built in the 1970s. However, because of the lack of advanced techniques, such as parsing, named entity recognition, information extraction and so on, the system performance in terms of quality of the responses was not very good. With the appearance of the related techniques, QA field has been developed rapidly. In particular, the domain has been boosted by the creation of a question answering track in the eighth Text Retrieval Conference (TREC-8) in 1999. Since then, many methods have been proposed and tested on real data for QA. For example, one can combine IR techniques and Named Entity (NE) recognition techniques in a QA system. This combination has often been used for the following reasons: IR has advanced techniques for indexing and retrieving texts in large collections of texts, but lacks sophisticated methods to deal with the semantics of the query and the documents [RPS00]. On the other hand, NE recognition extracts certain types of semantic information, but lacks efficient techniques for indexing and retrieval. Hence, a reasonable combination of them can be beneficial. This combination usually works in the following way: the IR techniques treat the question as a query and return a set of top-ranked documents or passages; then, the NE techniques are used to process the question and analyse the top-ranked documents or passages returned by the IR system and give the precise answer. So far, many QA systems combining IR and NE technologies have been built in such a way (e.g., [ACS00]).

Currently, most of the existing QA systems try to answer open-domain questions.

In principle, this kind of QA system can be established by first, creating a large knowledge base with the information extracted from documents; and then, querying such knowledge base. However, the knowledge is infinite and researchers cannot establish a large enough knowledge base to cover all the world knowledge. In addition, there are limitations on the advanced techniques of NLP, IE, knowledge representation and reasoning [AB00], so that it is impossible to answer correctly all the open-domain questions. The types of questions which one is able to answer are limited. They usually concern named entities such as time, persons and places. In order to enlarge the types of questions, one has to use more knowledge. This is only possible for a domain-specific application because in a specific domain, there is often an existing domain knowledge base available.

1.2 Our project

In our study described in this thesis, we will build a domain-specific question answering system for the construction sector. The goal of this project is to provide a precise answer for the professional user's question in the construction sector.

Our system takes a natural language question as input and identifies short passages, which may contain an answer. For our project, we use a general-purpose IR system – Okapi – to identify a small set of passages that may contain an answer. The identification of this small set of passages has been implemented by another MSc. student [Zha03] by using Okapi [Oka]. Our work starts with the identified passages and tries to verify if there is indeed a possible answer in each of these passages.

Our work involves two main parts. The first part concerns the common QA problems – analyzing questions and documents to extract common named entities from them. This part is similar to most of the current methods on QA. The second part, domain-dependent QA is new. In our application area – construction – there is a thesaurus, the Canadian Thesaurus of Construction Science and Technology, which contains a large network of approximately 15,000 concepts with approximately 26,000 links between them. We will exploit this thesaurus to answer domain-specific

questions. In particular, we will consider categories of concepts in this thesaurus as special NEs on which one may ask questions. The extension of general domain NEs to domain-specific NEs constitutes our main contribution.

Our main purposes in this project are:

- to develop an extended QA method for domain-specific NEs or categories of concepts and compound terms.
- to experiment our method on a test collection.

For these purposes, we need to solve three problems in domain-specific QA:

1. how to extract common and extended NEs based on a thesaurus;
2. how to determine compound terms to create a more precise representation than with single keywords, with the help of the thesaurus;
3. how to deploy search strategies for utilizing the extended NEs and domain compound terms in QA processes.

On the extraction of extended NEs (categories henceforth), we first implement a static method: we choose some fixed concepts in the thesaurus as extended NE categories, on which users may ask questions. For example, "material" is identified as such a concept. Then users can ask questions such as " What material ...? ". Unfortunately, this method results in a decrease of 6.1% in the system performance in comparison with the Keyword-based search, in which no extended NEs are identified. We have thus to abandon this idea.

Through analyzing the failure reasons, we design a dynamic method for choosing categories. This method brings an improvement of 7.11% to the system performance compared to the result returned by Okapi directly. The main idea of this dynamic method is that all the higher level categories of a concept in the thesaurus are considered as possible extended NE categories.

In our system, we use three different search strategies: Category-based search, NE search and Definition search. Establishing a search strategy includes determining

parameters, formulas, patterns that may be matched in identifying possible answer locations, as well as weight calculation methods taking into account all kinds of parameters. Our experimental results show that, the system performance by using Category-based search strategy is improved by 7.11% compared to the results returned by Okapi directly and by using NE and Definition search strategies, it is improved by 10.35%. These results clearly show that the methods we propose in this thesis are appropriate for domain-specific QA.

This thesis is structured as follows. In Chapter II, previous related work is reviewed. In Chapter III, we concentrate on describing our approach and techniques for QA as well as some implementation details. In Chapter IV, our experimental results are presented and analyzed. In Chapter V, we will draw some conclusions and describe some future research issues related to a domain-specific QA system.

Chapter 2

Related work

Question Answering (QA) combines techniques from Information Retrieval (IR), Information Extraction (IE) and Natural Language Process (NLP) techniques. IR provides methods for indexing and searching documents in large collections. IE aims to recognize more specific types of information. NLP aims to develop techniques for dealing with all the aspects of natural language such as syntax and semantics. The goal of QA is to combine all these techniques in order to identify precise answers for user's natural language questions. In this chapter, we will describe the IR, IE and QA techniques related to our work.

2.1 Information Retrieval

In this section, we will describe what IR is, and its current state of the art and its future.

2.1.1 Basic concepts of IR

IR studies the retrieval of information from a collection of documents in order to satisfy a user's information need, usually expressed as a query in natural language. Salton and Mcgill defined it as follows:

> Information retrieval is concerned with the representation, storage, or-

> ganization, and accessing of information items. Items found in retrieval systems are characterized by an emphasis on narrative information. Such narrative information must be analysed to determine the information content and to assess the role each item may play in satisfying the information needs of the users [SM83].

The primary goal of an IR system is to retrieve quickly all relevant documents to a user's query while retrieving as few non-relevant documents as possible. There are three basic concepts concerning IR: document, query and relevance [Nie03].

- Document: A document can be a text, a piece of text, a Web page, an image, a video and so on. All document units can constitute a response for a user's query.
- Query: A query expresses the information that the user needs.
- Relevance: Relevance is the central concept in the IR because the goal of the IR is to find the relevant documents. All the evaluations of IR systems are based on this concept. However, the concept of relevance is also very complex, because the users of IR system have greatly different needs and they also have very different criteria to judge if a document is relevant. Therefore, the concept of relevance always covers a very vast range of criteria and relations. In the relevant documents, the user should be able to find information that he needs. According to an estimation of relevance, the system must judge if a document should be given to the user as a response.

In order to determine the documents to be retrieved, the general approach is to carry out an indexing process on both documents and queries. This process produces a set of weighted indexes for each document and query, which constitutes an internal representation of them. The degree of relevance of a document to a query is determined by the correspondence of their internal representation. This degree is determined during the retrieval process. We will give more details about indexing and retrieval methods in the next two sections respectively.

2.1.2 How are documents and questions indexed?

In order to speed up the search, one should index the text of the documents in the documents collection. As not all words are equally significant for representing the semantics of a document, it is necessary to preprocess the text of the documents in the collection to determine the terms to be used as index terms. Usually, the document preprocessing can be divided into the following steps [BYRN99]:

- Tokenization: It is the process of converting a stream of characters (the text of the documents) into a stream of words (the candidate words to be adopted as index terms). Normally, it recognizes spaces and punctuation marks as word separators.

- Stoplist: Words which are too frequent among the documents in the collection are not good discriminators. Such words are frequently referred to as stopwords and are normally filtered out from potential index terms. Articles, prepositions, and conjunctions are natural candidates for a list of stopwords. For example, the terms like "*the* ", "*on* ", or "*and*" have no meanings by themselves and might lead to the retrieval of various documents which are unrelated to the query.

- Stemming: Stemming of the remaining words has the objective of removing prefixes and suffixes and allowing the retrieval of documents containing syntactic variations of query terms, for example, build, building, built, etc.

Once a set of index terms for a document is determined, we notice further that not all terms are equally useful for reprensenting the document contents. Clearly, the distinct index terms should have varying relevance when used to describe document contents. This effect is captured through the assignment of numerical weights to each index term of a document. Among the term-weighting schemes, the approach based on *tf*idf* is the best known in IR. Here, "*tf*" means "term frequency" and "*idf*" means "inverted document frequency". "*tf*" indicates the importance of a

term for a document. In general, this value is determined by the frequency of the term in the document. "*idf*" measures if the term is discriminating or specific to some documents[1].

Once the indexing process has been carried out, one usually constructs an inverted file to store the indexing result. The structure of inverted file is in the following form:

$$Word = \{..., Doc_i, ...\}$$

That is, each index term is related to a list of documents which contain the word. The advantage of using an inverted file is that the retrieval process can be very fast: we only need to identify the lists of documents related to the words in a query, then the lists are combined.

2.1.3 How are documents retrieved?

Once indexing has been done, the next question is to determine the degree of correspondance between a document and a query. The way of doing this is determined by a retrieval model. There are three classical models in information retrieval, namely, Boolean model, vector space model, and probabilistic model [BYRN99]. We will briefly present them below.

Boolean model

This is a simple retrieval model based on set theory and Boolean algebra. The index term weights are all binary. It means the weight of each index term is 0 or 1. The queries are specified as Boolean expressions, which have precise semantics. Then one can calculate the similarity of a document to the query according to whether the Boolean expression of the query is satisfied by the set of terms of the document. If the value of similarity is 1, it means that the document is relevant to the query. Otherwise,

[1] idf = log(N/n), where N is the number of documents in the corpus, and n is the number of documents that contain the term. The higher is n, the less is the term specific to some documents, and the lower is idf.

the document is not relevant to the query. The Boolean model was adopted by many of the early commercial bibliographic systems.

Its main advantages are: First, it has the clear formalism behind the model. Second, it is simple to implement.

The main disadvantages of the Boolean model are: First, its retrieval strategy is based on a binary decision criterion without any notion of a grading scale, which prevents good retrieval performance. As a matter of fact, the classical Boolean model without term weighting adopts exact matching for retrieval. This may lead to retrieving too many documents (if the query is a long OR-ed expression) or too few documents (if the query is a long AND-ed expression). Second, while Boolean expressions have precise semantics, generally it is not easy to translate an information need expressed in natrual language into a Boolean expression. In fact, most users find it difficult and awkward to describe their requests in terms of Boolean expressions.

Vector space model

It proposes a framework in which partial matching is possible. This is accomplished by assigning non-binary weights to index terms in queries and in documents. The document and query are represented as t-dimensional vectors where t is the number of all the indexed words. The vector space model evaluates the degree of similarity between each document and the user's query, for example, by the cosine of the angle between these two vectors (see Figure 2.1). Since the value of similarity varies from 0 to 1, the vector space model can rank the documents according to their degrees of similarity instead of answering whether a document is relevant or not.

The main advantages of the vector space model are: first, the non-binary term-weighting scheme improves retrieval performance; second, the partial matching strategy allows the retrieval of documents that contain part of the terms of the query; and third, the cosine ranking formula ranks the documents in terms of their degree of similarity to the query.

The main disadvantage of the vector space model is that index terms are assumed

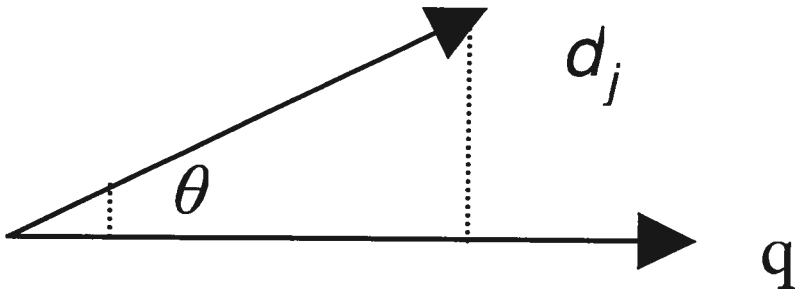


Figure 2.1: The cosine of θ is adopted as $sim(d_j, q)$.

to be mutually independent. However, in practice, it is difficult to consider term dependencies because it is difficult to determine whether two terms are dependent.

Probabilistic model

This model attempts to capture the IR problem within a probabilistic framework. This framework considers the appearance or absence of terms as the basic events. A document and a query are all formed by a set of such events. The basic probabilistic model tries to determine how probable each event is characteristic of a relevent or irrelevant document through an analysis of a set of sample documents. Then given a query, the correspondence degree of a document is determined according to the extent to which the characteristic events of the documents correspond to those of the query.

The main advantage of the probabilistic model is that documents can be ranked in descending order of their probability of being relevant instead of answer ing whether a document is relevant or not.

The disadvantages of the probabilistic model are: first, the model needs to have a set of relevant and non-relevant documents for the estimation of probabilities; second, this model in its classical form does not take directly into account the frequency that an index term occurs inside a document; and third, the model usually adopts the independence assumption for index terms. However, as discussed in the vector space model, the consideration of term dependencies might be a problematic.

2.1.4 Current state of IR [BYRN99]

Recently, the area of information retrieval has grown rapidly beyond its primary goals of indexing text and searching for useful documents in a collection. Nowadays, research in IR includes modelling, document classification and categorization, system architecture, user interfaces, data visualization, filtering, etc. In the past, IR was seen as a narrow area used only by librarians and information experts. This situation lasted for many years. Since the beginning of the 1990s, along with the development of the World Wide Web (WWW) and the emergence of mass storage devices, this situation has changed. As a result, IR has gained a key place in the information processing field.

Currently, the research and development in IR is extending beyond its original area of library. Active research is being pursued in several directions. First, one tries to develop techniques that allow us to retrieve higher quality information in the dynamic world of the Web and from large information resources. Second, people are developing techniques that yield faster indexes and shorter query response time. This point is more necessary now than ever before because of the continually increasing demand for access. Third, we try to develop techniques that can better understand the users' behaviours, because the quality of the retrieval task is greatly affected by the users' interaction with the system.

The Web is becoming a universal repository of human knowledge and culture which has allowed unprecedented sharing of ideas and information in a scale never seen before. Basically, low cost, greater access, and publishing freedom have allowed people to use the Web as a highly interactive medium. Meanwhile, people always hope the system to return accurate results quickly. However, in fact, it is difficult to satisfy this requirement because of the limitation of IR techniques for recognizing the semantic contents of texts. In order to better recognize the contents of a text, information extraction is often employed.

2.2 Information Extraction

In this section, we will present the concept of IE and its development and application, as well as the relationship with IR.

2.2.1 Information Extraction

Information Extraction analyzes unrestricted text in order to extract information about pre-specified types of events, entities or relationships [Gro96]. For example, a CIA agent who tracks terrorist activities organized by international terrorism may use an IE system to gather the needed information. News articles may be the input to the IE system. This IE system may classify the types of terrorist event, and record the identified or suspected perpetrators, dead or injured victims, and any damage to buildings or the infrastructure, as well as the time and location of the event. IE also can be regarded as the activity of generating a structured information source (or database) from an unstructured or free text information source. Then, this structured data can be used for: 1) searching or analyzing data using conventional database queries or data-mining techniques; 2) generating a summary; 3) constructing indices of the source texts [GW98].

[Sag81] presented a survey on IE techniques. Early work on IE was on template filling, which aims to feed structured records with information extracted from natural language source texts. The Linguistic String Project at New York University and FRUMP system [DeJ82], which was designed and implemented by Roger Schank and Gerald De Jong at Yale University, are good examples using this approach. After that, many IE systems have adopted a similar approach.

IE has been developed rapidly since the late 1980's when the DARPA (Defense Advanced Research Projects Agency) led government effort to make progress in text processing technologies through the cooperation of researchers and developers in government, industry and academia. The research results were provided to analysts in the intelligence community with improved operational tools. This program was ended in the fall of 1998 because of shortage of funding . Message understanding conferences

(MUCs) are main activities in driving the field of IE forward. In the mid-1980's, the US Navy sponsored projects aiming to construct systems for understanding all kinds of naval messages including those about terrorism. Some systems were constructed for understanding the newspaper articles about terrorism and answering the related questions. In order to better understand and compare their systems' performance, a number of these message understanding (MU) projects decided to work on a set of common messages and then to see how their systems would perform when given some new, unseen messages. In this case, the message understanding conferences were constituted. Information extraction in the sense of the Message Understanding Conferences has been defined as the extraction of information from a text in the form of text strings and processed text strings that are placed into slots labelled to indicate the kind of information that can fill them [MUC03].

MUC examines evaluations of information extraction system in terms of pre-established tasks. The evaluation metrics have evolved along with each MUC. The starting points were the standard IR metrics of recall and precision. In MUC-6, the evaluation emphasized finer-grained evaluation and portability issues and comprised four subtasks – named entity recognition, coreference identification, and template element and scenario template extraction tasks [GW98]. The Named Entity and Coreference tasks entailed Standard Generalized Markup Language (SGML) annotation of texts and were being performed for the first time. The other two tasks, Template Element and Scenario Template, were information extraction tasks that followed on from previous MUC evaluations. Participants were invited to enter their systems in four different task-oriented evaluations. In MUC-7, another subtask for evaluation – template relation was added on top of the four subtasks in MUC-6.

Along with MUCs, many new techniques have been brought in.

- **Named entity recognition.** This task requires the recognition and classification of definite named entities such as organisations, persons, locations, dates and monetary amounts.

- **Coreference resolution.** This task requires the identification of expressions

in the text that referred to the same object, set or activity. These include variant forms of name expression (Ford Motor Company . . . Ford), definite noun phrases and their antecedents (Ford ... the American car manufacturer), and pronouns and their antecedents (President Clinton ... he).

- **Template element filling.** This task requires the filling of small scale templates wherever they occurre in the texts. There are only two such template elements, one for organizations and one for persons in MUC-6. In MUC-7, such as organizations, persons, certain artifacts, and locations, with slots such as name (plus name variants), description as supplied in the text, and subtype. This task has been carried out successfully with a reported accuracy of over 95% for the best systems.

- **Scenario template filling.** The task requires the detection of specific relations holding between template elements relevant to a particular information need and the construction of an object-oriented structure recording the entities and details of the relation.

- **Template Relation filling.** Template Relation (TR) evaluation identifies general relational objects which point to Template Element (TE) objects. This task is viewed as the next step up from the TE task and the beginning of a compilation of scenario-independent facts about TEs. The three relations included in MUC-7 are LOCATION_OF, EMPLOYEE_OF, and PRODUCT_OF.

The evaluation results from MUC-3 to MUC-7 by tasks are presented in Table 2.1 [MUC03].

Evaluation/Tasks	NE	CO	TE	TR	ST
MUC-3					$R < 50\%$
					$P < 70\%$
MUC-4					$F < 56\%$
MUC-5					$EJVF < 53\%$
					$EMEF < 50\%$
MUC-6	$F < 97\%$	$R < 63\%$	$F < 80\%$		$F < 57\%$
		$P < 72\%$			
MUC-7	$F < 94\%$	$F < 62\%$	$F < 87\%$	$F < 76\%$	$F < 51\%$

Table 2.1: Maximum results reported in MUC-3 through MUC-7 by task [MUC03].

CO: Coreference, TE: Template Element,

TR: Template Relation, ST: Scenario Template,

R: Recall: proportion of relavant material actually retrieved,

P: Precision: proportion of retrieved material actually relevant,

F: F-Measure[2] with Recall and Precision Weighted Equally,

EJVF: English Joint Venture F-Measure (an F-measure for documents in a particular area),

EMEF: English Microelectronics F-Measure (an F-measure for documents in another area).

[2]F-Measure: It combines precision and recall into one number [Hea02] as follows:.

$$F_b = \frac{(b^2+1)PR}{b^2P+R}$$

We set b to 1 in our work.

Since the MUCs, several significant IE projects have been developed, such as LaSIE, AVENTINUS, ECRAN, GATE, and so on (see [Gro96] for more details). We know that information extraction is a difficult task, because there are many ways of expressing the same fact and information may need to be combined across several sentences in natural language. IE is not an isolated domain and it has a close relation with natural language processing and computational linguistics. Up to now, there are still some limitations in natural language processing and computational linguistics techniques so that IE is also limited. In addition, Templates are usually handcrafted by human experts to suit a particular domain and therefore template filling cannot be easily transferred to a new domain. So, one of the developing trends in IE is to seek automatically learning methods to extract templates.

There are wide application areas of information extraction. IE technology has already been applied to Finance, Military intelligence, Medicine, Law, Police, Technology/product tracking, Academic research, Employment, Fault Diagnosis, Software system requirements specification and so on.

2.2.2 IR and IE

Information extraction adopts many mature technologies from information retrieval, which selects a more relevant subset of documents from a large collection has a given user query. On the other hand, IR can also benefit from IE in selecting more meaningful indices. In this subsection, we describe some of their relationships.

Differences between IR and IE

First, the basic functions of IR and IE systems are different: IR retrieves relevant documents from a document collection while IE extracts relevant information from documents [GW98, Gro96]. Therefore, the two techniques are complementary, and their combination has the potential to create powerful new tools in text processing. [GW98] gives examples to show the differences and complementary roles of IR and

IE. For example[3], one might scan business newswire texts for announcements of management succession events (retirement, promotions, etc.), extract the names of the participating companies and individuals, the post involved, the vacancy reason, and so on. This management succession event scenario was part of the DARPA MUC-6 information system evaluation. For this evaluation texts pertaining to management succession were required. To obtain them, a corpus of Wall Street journal articles was searched using an IR system with the query shown in Figure 2.2 a). The query was deliberately not fine-tuned, as it was expected to obtain some proportion of irrelevant texts. A sample of a relevant text retrieved by this query is shown in Figure 2.2 b). Such texts were then run through IE systems whose task was to fill in a template whose structure is shown in Figure 2.2 c) to produce results as (partially) shown in Figure 2.2 d). As secondary output the system used here is able to generate a natural language summary of the information in the template as shown in Figure 2.2 e).

[3]This example is from [GW98]

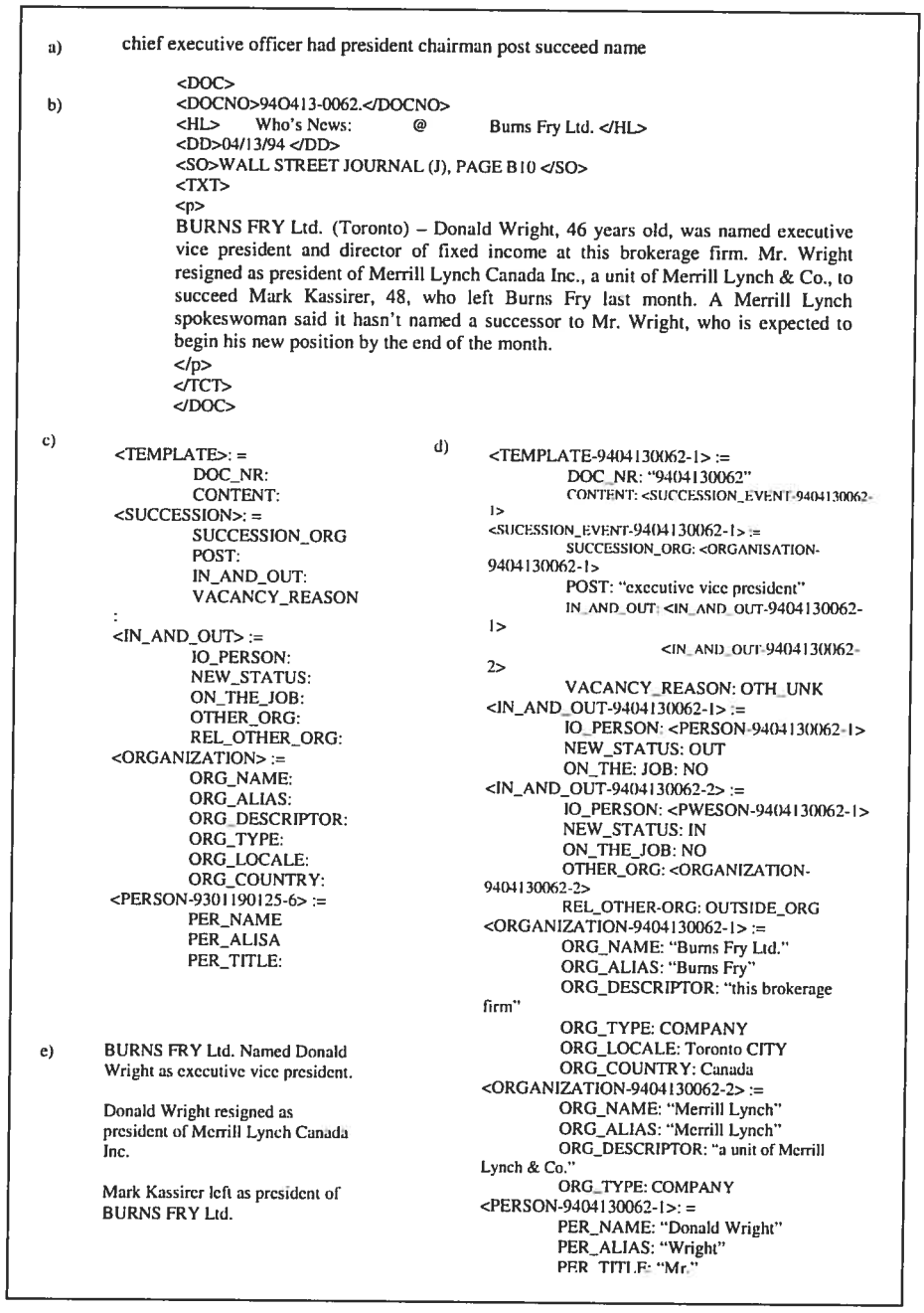


Figure 2.2: IR and IE: a) an IR query. b) a retrieved text. c) an empty template. d) a fragment of the filled template. e) a summary generated from the filled template.

Second, the techniques they have deployed are also different. Most work in IE has focused on rule-based systems in computational linguistics and natural language processing. IE needs to parse texts for structural or syntactic properties in order to identify the information to extract. Here is an example in [GW98], "Carnegie

hired Mellon" is not the same as "Mellon hired Carnegie" which differs again from "Mellon was hired by Carnegie". To extract the correct information, some level of linguistic analysis is necessary. Here are some examples from [GW98]:

1. BNC Holdings Inc. named Ms. G. Torretta to succeed Mr. N. Andrew as its new chair-person;

2. Nicholas Andrew was succeeded by Gina Torretta as chair-person of BNC Holdings Inc.;

3. Ms. Gina Torretta took the helm at BNC Holdings Inc. She succeeds Nick Andrews.

To extract a canonical fact such as "G. Torretta succeeds N. Andrews as chair-person of BNC Holdings Inc." from each of these alternative formulations, we need to cope with grammatical variations (active/passive, was succeeded by vs. succeed), lexical variations (named to vs. took the helm) and cross-sentence phenomena (anaphora, Ms Gina Torretta vs. She).

IR usually exploits little linguistic analysis of texts. It employs statistics to determine the important indexes for texts. While a query is submitted, a degree of correspondence is calculated between the query and each document according to the importance of the indexes in the document, which occur in the query.

Given the complementary of IE and IR, it is possible to combine them. This has been investigated by several researchers [Gro02]. The advantage of such a combination is they take into account not only the content words of a document but also some semantic information obtained by IE. It can improve the precision of IR system.

However, such a combination also has some limitations. One is that it needs to work out reasonable schemes for deploying the semantic information into IR system. Otherwise, it will creat undesirable effects for IR system. Another one is that the simple combination cannot satisfy the user's needs since it doesn't provide the direct answers for the user's questions.

The combination of IR and IE is particularly interesting for finding specialized information on the Web. Although there is a huge amount of information on the web, people still find that it is difficult to obtain proper information relevant to their information needs. Often, users want quick and direct responses to their questions. For example, for a factual question such as " *Who is the current President of the USA?*", they desire to obtain the precise answer *George W. Bush.* Present IR systems cannot answer such question directly, but only give an indication of where answer will probably be found. The user has to do a further search in the documents to find the answer. Clearly, just the simple combination of IR and IE still cannot satisfy application needs. What is needed is a system that can pinpoint the exact candidate answers in a document collection from which we can infer the answer to a specific question. This leads to a new type of system – "question answering". This system is much more in accordance with the idea of user-driven information extraction, accepting natural language questions, then generating information contained either directly in the text or inferred from it and finally returning the precise answer to the user[IE001]. Despite the name difference (Question Answering v.s. Information Extraction), many researchers in QA believe that the most important influencing element to question answering is still information extraction technology. QA is an ideal test bed for demonstrating the power of IE. There is a natural co-operation between IE and IR for the purpose of QA.

In the next section, we will describe the problem of QA.

2.3 Question Answering

In this section, we will present the concept of QA, and main methods that have been adopted in the existing QA system.

2.3.1 Definition

Question answering is a field on information process domain. It tries to retrieve a direct answer to a user's question. The goal is to implement a system that can automatically find answers from a vast amount of underlying text. QA is a promising area related to information retrieval as it takes a step closer to information retrieval rather than document retrieval [Uni].

Research in QA has received a strong boost by the QA track at the TREC conferences (TREC-8 QA track (1999) and TREC-9 QA track (2000)), with a wide range of participating research groups, both from industry (e.g. IBM, Sun, Microsoft) and academia (with groups from the US, Europe and Asia).

START (SynTactic Analysis using Reversible Transformations)[Inf], developed by Boris Katz and his associates in the Infolab Group, is an example of a question answering system that uses natural language annotations. It has been available to users on the World Wide Web since December 1993. It is one of the earliest QA systems.

Recently, a large number of QA systems have emerged. Primarily, they follow two directions: one is to use the TREC QA [Lin01] data as the test corpus and develop their own search engines and answer extraction techniques on top of the corpus; the other direction is to use the Internet as a potential answer source and use generic search engines, such as Okapi, to retrieve information related to question and do further post-processing to extract answer for the question[RFQ+02]. Techniques that have been adopted are almost the same for both directions. From another point of view, QA systems may be divided into two types, i.e., open-domain and domain-specific. We will review some recent work of these two types in the following section.

2.3.2 Open-domain QA

In the early studies, several approaches to QA have been developed, such as conceptual theory of QA with associated question taxonomy [Leh78], and the mechanisms for generating questions [GG91]. However, these approaches did not apply parsing,

named entity recognizing and information extraction techniques. Recently, QA researchers use various techniques to find precise answer to user's question. There are mainly 5 types of approach:

- based on IR and NLP [GH00, HGH+00];
- based on NE [ACS00, SL00];
- based on semantic match as well as term weighting and coverage [CCKL00];
- based on integrated NLP resources [HMM+00];
- based on scenarios techniques [Leh75];

In this section, we will describe the methods that have been proposed for open-domain QA [RFQ+02].

2.3.2.1 QA based on IR and NLP techniques.

The main idea of this approach is to establish the template sets of question types and answer types. The users question can then be indexed by its type, from which all equivalent forms of the answer can be determined. These QA equivalence types can help with both query expansion (for IR) and answer pinpointing (for NLP).

The steps of this approach are approximately the following ones:

First, question templates and answer templates are constructed. Template examples are shown in Figure 2.3.

Second, a given question is first parsed to create a query to retrieve the top ranked documents. These top-ranked documents are then split into segments and further ranked.

Third, the ranked segments are input into a parser, which is trained on a corpus to return both syntactic and semantic information.

Finally, according to the syntactic and semantic information returned by the parser, the potential answers are then extracted and sorted according to a ranking function involving the match with the question type and patterns.

Examples of this approach are [GH00, HGH+00].

Question examples	Question templates
Who was Johnny Mathis' high school track coach?	who be <entity>'s <role>
Who was Lincoln's Secretary of State?	
Who was President of Turkmenistan in 1994?	who be <role> of <entity>
Who is the composer of Eugene Onegin?	
Who is the CEO of General Electric?	
Actual answers	**Answer templates**
Lou Vasquez, track coach of ... and Johnny Mathis	<person>,<role> of <entity>
Signed Saparmurad Turkmenbachy [Niyazov], president of Turkmenistan	<person> <role-title> of <entity>
	<entity>'s <role> <person>
... Turkmenistan's President Saparmurad Niyazov ...	<person>'s <entity>
... in Tchaikovsky's Eugene Onegin ...	<role-title><person> ... <entity>
Mr. Jack Welch, GE chairman ...	<role>
...Chairman John Welch said ... GE's	<subject> | <psv object> of related role-verb

Figure 2.3: Templates examples for proper-person.

2.3.2.2 QA based on IR and NE

This approach is used to process a question whose answer is a common NE or an extended NE in a specific domain. As this approach is closely related to ours, we will go into it in more details. For each question, a set of relevant passages that most likely contain the answer is first identified. Then, a candidate set of named entities is extracted from these retrieved passages as potential answers to the question. From the question, the expected answer type is also identified. Sometimes, named entities are first extracted from the documents collection, and then relevant passages are filtered. There isn't a fixed order for these two steps. The order varies from a system to another. Both the expected answer type [4] and these extracted entities are compared. Only those entities that match the type required by the question are retained. Then these passages are re-ranked according to how well its types match the expected answer type. Some related frequency and position information are applied in this stage. Examples of this approach are [ACS00, SL00]. In order to know well about this method, we will further introduce named entity and named entity recognition.

In [MUC95], named entities refer to entities (such as, organizations, persons, lo-

[4]The expected answer type should be either a common NE or a domain-specific category.

cations), times (auch as, dates, times), and quantities (such as, monetary values, percentages). For example, suppose the following passage:

> Iraqi President **Saddam Hussein** demanded **Saturday** that the **U.N. Security Council** remove sanctions imposed after **Iraq's 1990** invasion of **Kuwait**, saying it was complying with **U.N.** disarmament demands.

This passage contains 7 named entities:

> "**Saddam Hussein**" is a PERSON; "**Iraq**" and "**Kuwait**" are LOCATIONs; "**U.N.**" and the "**U.N. Security Council**" are ORGANIZATIONs; "**1990**" and "**Saturday**" are DATEs.

The recognition of NE was introduced as a part of the Sixth Message Understanding Conference in 1995 (MUC6). Actually, Named Entity Recognition (NER) is a subtask of IE, which is typically designed to extract fixed types of information in specific domains and languages.

In [SL00], the author points out that the NE technology is an important component for QA. Domain independent IE can result in a QA breakthrough as it can recognize the nature of some concepts. However, high-level IE technology beyond NE has not been in the stage of possible application until recently. Clearly, many researchers working on QA regard named entity extraction as a core technology for obtaining semantics of texts [NIS03]. Up to now, a lot of researchers have worked on NE recognition and many approaches have been proposed in the CoNLL-2002 [CoN02] and CoNLL-2003 [CoN03] shared tasks. CoNLL is an international forum for discussion and presenation of research on natural language learning. It is a yearly meeting organized by SIGNLL, the Association for Computational Linguistics Special Interest Group on Natural Language Learning [CoN02]. Roughly, the methods of NE recognition can be divided into three types: based on gazetteers, based on heuristics or based on machine learning.

1. **Based on gazetteers**

   A gazetteer is a list of geographic names (country, province, city and so on) or person names (family names, male first names and female first names) or others. This method is to include gazetteers in the system and then through gazetteers lookup to find named entities. It's usually used by combining with other methods. Examples of this method are [BON03, ML03].

2. **Based on heuristics**

   Heuristics-based methods use rules written by human experts after inspecting examples and common knowledge bases. Examples of such methods are [Gro01], [EFO+02] and [WNC03]. In CoNLL-2002 shared task, researchers found out that choice of features is important for recognizing named entities [SM03]. The main tasks involved in this approach are as follows:

   First, constructing some rules in connection with a knowledge base. These rules are constructed according to observations on examples.

   Second, tagging feature terms, of which the words describe the characteristics and function of an entity. For example, features are used for distinguishing money, time, date, types of capitalization and so on.

   Third, using syntax analysis, gazetteer, and some feature information to identify some NEs or tag more feature information.

   Forth, one use rules, feature information, contextual information and some NE taggers to recognize other NEs. For example, "Jun., 1999" is tagged as one NE (DATE) instead of two NEs (DATE (month) and DATE (year)). In this step, we should pay more attention on rules priority. It is based on pattern length, rule status and rule ordering.

   Fifth, by applying a set of filters, one gets rid of false hits. This step aims to improve the precision of NE tagging.

   The advantage of this method is simple and easy to implement. The performance of this method is acceptable. The disadvantage is that one has to write

a new set of rules for every new language and new entity.

3. **Based on machine learning**

   Learning-based methods include a machine learning component. To develop such a system, one has to provide training data, development data and test data. The NE recognition methods will be trained with the training data. The parameters of the methods are tuned by the development data. Finally, the performance of system will be tested on the test data [San02]. Sixteen systems [BON03, CMP03a, CMP03b, CN03, CC03, MD03, FIJZ03, Ham03, HvdB03, KSNM03, MMP03, ML03, MLP03, WP03, WNC03, ZJ03] have participated in the CoNLL-2003 shared task. These systems used a great variety of machine learning techniques for implementing named entity recognition. The results for the test data for English and German are shown in Table 2.2 and Table2.3, respectively.

References	Precision	Recall	F-Measure
[FIJZ03]	88.99%	88.54%	88.76
[CN03]	88.12%	88.51%	88.31
[KSNM03]	85.93%	86.21%	86.07
[ZJ03]	86.13%	84.88%	85.50
[CMP03b]	84.05%	85.96%	85.00
[CC03]	84.29%	85.50%	84.89
[MMP03]	84.45%	84.90%	84.67
[CMP03a]	85.81%	82.84%	84.30
[ML03]	84.52%	83.55%	84.04
[BON03]	84.68%	83.18%	83.92
[MLP03]	80.87%	84.21%	82.50
[WNC03]	82.02%	81.39%	81.70
[WP03]	81.60%	78.05%	79.78
[HvdB03]	76.33%	80.17%	78.20
[MD03]	75.84%	78.13%	76.97
[Ham03]	69.09%	53.26%	60.15
baseline	71.91%	50.90%	59.61

Table 2.2: The results for English test data.

In [SM03], it gives a simple description for methods deployed in these systems. An excerpt is as below:

> The most frequently applied technique in the CoNLL-2003 shared task is the Maximum Entropy Model. Five systems used this statistical learning method. Three systems [BON03, CN03, CC03] used Maximum Entropy Models in isolation. Two more systems [FIJZ03, KSNM03] used them in combination with other techniques. Maxmum Entropy Models seem to be a good choice for this kind of task: the

References	Precision	Recall	F-Measure
[FIJZ03]	83.87%	63.71%	72.41
[KSNM03]	80.38%	65.04%	71.90
[ZJ03]	82.00%	63.03%	71.27
[MMP03]	75.97%	64.82%	69.96
[CMP03b]	75.47%	63.82%	69.15
[BON03]	74.82%	63.82%	68.88
[CC03]	75.61%	62.46%	68.41
[ML03]	75.97%	61.72%	68.11
[MLP03]	69.37%	66.21%	67.75
[CMP03a]	77.83%	58.02%	66.48
[WNC03]	75.20%	59.35%	66.34
[CN03]	76.83%	57.34%	65.67
[HvdB03]	71.15%	56.55%	63.02
[MD03]	63.93%	51.86%	57.27
[WP03]	71.05%	44.11%	54.43
[Ham03]	63.49%	38.25%	47.74
baseline	31.86%	28.89%	30.30

Table 2.3: The results for German test data.

top three results for English and the top two results for German were obtained by participants who employed them in one way or another.

Hidden Markov Models were employed by four of the systems [FIJZ03, KSNM03, MMP03, WP03] that took part in the shared task. However, they were always used in combination with other learning techniques. [KSNM03] also applied the related Conditional Markov Models for combining classifiers.

Learning methods that were based on connectionist approaches were applied by four systems. [ZJ03] used robust risk minimization,

which is a Winnow technique. [FIJZ03] employed the same technique in a combination of learners. Voted perceptrons were applied to the shared task data by [CMP03a] and [Ham03] used a recurrent neural network (Long Short-Term Memory) for finding named entities. Other learning approaches were employed less frequently. Two teams [CMP03b, WNC03] used AdaBoost.MH and two other groups [MD03, HvdB03] employed memory-based learning. Transformation-based learning [FIJZ03], Support Vector Machines [MMP03] and Conditional Random Fields [ML03] were applied by one system each.

Combination of different learning systems has proven to be a good method for obtaining excellent results. Five participating groups have applied system combination. [FIJZ03] tested different methods for combining the results of four systems and found that robust risk minimization worked best. [KSNM03] employed a stacked learning system which contains Hidden Markov Models, Maximum Entropy Models and Conditional Markov Models. [MMP03] stacked two learners and obtained better performance. [WNC03] applied both stacking and voting to three learners. [MLP03] employed both voting and bagging for combining classifiers.

From the point of view of training examples, learning methods can be divided into two types, namely, supervised methods and non-supervised methods.

- Supervised methods, such as [BMSW97], use labelled training examples. One of the important questions for this method is how much training data is required to get acceptable performance. Usually, a fairly large number of labelled examples should be required to train an extractor. This method is adopted by most QA systems based on learning.
- Non-supervised methods use unlabeled examples for named entity extraction. First, a few hand-coded name elements and patterns are given. Then an unsupervised algorithm will learn new entities and their components

[QBW02]. [CS99] shows that the use of unlabeled data can reduce the requirements for supervision to just 7 simple seed rules. In addition, this approach also considers other features such as spelling of the name and the context. As many named-entity instances both the spelling of the name and the context in which it appears are sufficient to determine its type. More details on unsupervised algorithms are described in [CS99].

2.3.2.3 QA based on semantic match as well as term weighting and coverage

This method uses semantic match between the query type and terms, the *idf*-like term weighting of each term and also the coverage of these query related terms in the passage itself [CCKL00]. In this approach, they propose the technique that locates high-scoring passages, where the score of a passage is based on its length and the weights of the terms occurring within it. Passage boundaries are determined by the query, and can start and end at any term position. Here, we give a brief description about this method.

For passage retrieval purpose, they use the following concepts:

- Each document D in the corpus is treated as an ordered sequence of words:

$$D = (d_1, d_2, \ldots, d_m)$$

- A query is treated as a set of terms:

$$Q = (q_1, q_2, q_3, \ldots)$$

- An extent (u, v), with $1 \leq u \leq v \leq m$ is used to represent a subsequence of D beginning at position u and ending at position v:

$$d_u, d_{u+1}, d_{u+2}, \ldots, d_v$$

- A term t is assigned an idf-like weight:

$$w_t = \log(N/f_t)$$

where f_t is the number that t is matched in the corpus and N is the sum of the lengths of all the documents in the corpus.

- The weight W assigned to a set of terms $T \subseteq Q$ is the sum of the weights W assigned to each term in T:

$$W_T = \sum_{t \in T} w_t$$

- If an extent (u, v) is a cover for the term set T then it can be assigned a score combining the length of the extent and the weight of its matching terms:

$$C(T, u, v) = W(T) - |T| \log(v - u + 1)$$

Once the highest-scoring extents from distinct documents are determined, the centerpoint of each extent is computed as $(u + v)/2$ and a passage of fixed length (in this case, it is set to 200 words.) centered at this point is retrieved from the corpus. Then these ten highest-scoring passages are passed to the post-processor, which consults external databases containing lists of countries, states, cities, proper names, etc. The post-processing proceeds with the following steps:

1 Determine the answer category from the parser, which is a statistical context-free grammar parser based on WordNet.

2 Scan the passages for patterns matching the answer category.

3 Assign each possible answer term an initial score based on its rarity.

4 Decrease or increase the term scores depending on various quality heuristics.

5 Select from the passages the (50-byte or 250-byte)[5] answer that maximizes the sum of the term scores it contains.

6 Set the scores of all terms appearing in the selected answer to zero.

7 Repeat steps 5 and 6 until five answers are selected.

[5]The required outputs of TREC are of two kinds: 50 bytes and 250 bytes.

For example, suppose the question is: " Who is the leader of India? ", the top five 50-byte passages returned by the post-processor are:

1. Indian Prime Minister Vishwanath Pratap Singh f
2. Front. INDIA LEADER URGES SIKHS' PARTICI
3. PUNJAB PEACE. From Times Staff and Wire Report
4. Unist Party of India) leader, Mr M. Farooqui. bu
5. D Monday. J. N. Dixit said Velupillai Prabhakaran,

2.3.2.4 QA based on integrating NLP resources and NE

This method integrates different forms of syntactic, semantic and pragmatic knowledge as well as NE techniques [HMM+00]. In this system, question reformulation is used to construct a query that contains more information than the original question. A shallow parser is used to extract semantic information based on WordNet. Named entity recognition techniques are employed to ensure high quality passage retrieval. Potential answers are extracted from the semantically rich passages that match the question type, and then these candidate answers are further justified by using abductive reasoning and only those that pass the test are retrieved. Figure 2.4 illustrates the detailed processing steps in the system. This system scored very high in the recent TREC QA evaluation contest.

2.3.2.5 QA based on script techniques

The basic theoretical construct of this method is the notion of a script [Leh75]. Script-based knowledge is mundane information which tends to lie in the periphery of consciousness. The acts that define a script are things which people automatically do or expect to occur. Going to a restaurant, watching a football, and paying bills are examples of script activities. This method is mainly used in story understanding. Suppose the following story:

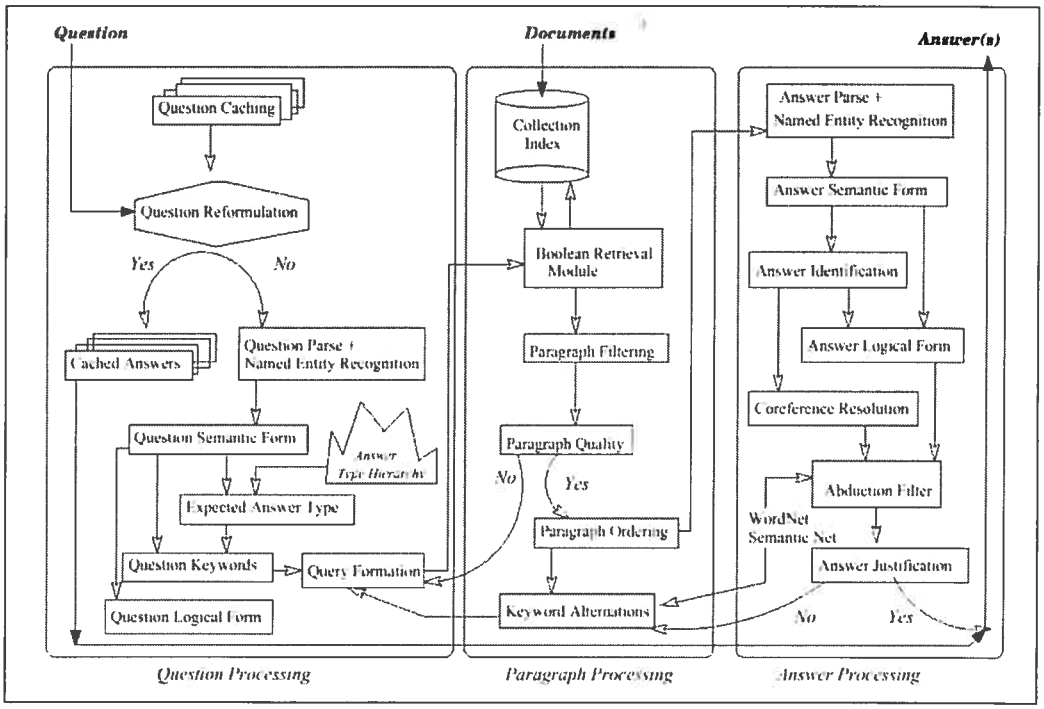


Figure 2.4: Question, paragraph and processing in FALCON [HMM+00]

John went to a restaurant. The hostess seated John. The hostess gave John a menu. The waiter came to the table. John ordered lobster. John was served quickly. John left a large tip. John left the restaurant.

The desired system would respond in a sample question answering session as follows:

Q. Why did John go to a restaurant?

A. So John could eat.

Q. Did the waiter give John a menu?

A. No, the hostess gave John a menu.

Q. What happened when the hostess gave John a menu?

A. John read the menu. The waiter saw that John was at the table. The waiter went to the table.

Q. What did John eat?

A. Lobster.

In order to answer these questions, implicit information is required. For example, one has to know implicitly that the purpose of going to a restaurant is to eat. Such implicit information is encoded into scripts. However, such an approach can only be used in a very limited application area in which there are typical scenarios.

2.3.3 Domain-specific QA

As above-mentioned, there have been some methods proposed for open-domain QA system. However, as we know, an open domain is infinite and one cannot establish a large enough knowledge base to cover it. On the other hand, there is an increasing need for domain-specific QA systems for professionals working in different areas. For example, professionals in the construction sector want to ask domain-specific questions. Therefore, the development of domain-specific QA is an urgent task. However, domain-specific QA system is not isolated and it is atop of an open-domain QA system. Thus, all the methods for open-domain QA system can be adopted in domain-specific QA system. The domain limitation makes it possible to acquire domain knowledge and to integrate it into QA system. In our case, our application area is the construction sector, in which there is a great deal of domain knowledge that we can exploit.

The integration of domain-specific knowledge into QA system means to exploit semantic information from domain-specific knowledge for identifying possible answers. This is a complex problem. Some studies have been done in this direction.

- Some systems do query expansion by using domain-specific knowledge, e.g., [JC94]. In query expansion, related terms and broader terms are used to expand the original query. These terms are added into the query.

- Some systems use concepts (unambiguous denotations of the entities) obtained from domain knowledge rather than words, to reduce the ambiguity problem.

[CTHD00] and [ARB94] are such systems. As a word can represent several concepts and a concept can be represented by several words, it is difficult to represent what the user is really interested in just by words. [CTHD00] points out that, to conduct concept-based search by using domain thesaurus, three main tasks have to be done:

1) building a concept index for target resources,

2) reformulating user's query in terms of concepts,

3) giving a concept-based search algorithm to match the user's concept query with the concept index of resources.

The quality of domain thesaurus is a key factor affecting the performance of this approach. In this approach, concepts are only used for the first stage of passages selection (an IR process). They are not used in the post-processing, i.e., the verification of answer type, the selection of different weighting schemes, the reordering of the candidate passages and so on. However, concepts are also highly useful for post-processing.

2.4 Summary of existing QA approaches

In the last three sections of this chapter, we have introduced some concepts and techniques on IR, IE and QA. Meanwhile, we also describe some QA approaches adopted by the existing QA systems. In this section, we will conclude the existing QA approaches.

Most of the QA systems are implemented as two steps: pre-processing and post-processing. The pre-processing uses IR techniques for a first document or passage selection. The IR system will take the question as a query and returns a set of top ranked documents or passages. Its main purpose is to select the highly potential passages that may contain an answer. A limited number of passages is usually selected at this step in order to avoid performing the costly post-processing on too many passages.

The post-processing aims to extract the information that the user seeks from the documents or passages returned by the pre-processing. In this step, some IE or NLP techniques are employed. In particular, Named Entity tagging is an important component in information extraction. Usually, There are two methods to tag NE: based on rules/gazetteers and based on machine learning. The former is simple to implement and its performance is acceptable. The latter technique may result in better performance than the former. However, it needs complex training process and asset of training data. In our project, we adopt the first method based on rules/gazetteers because we do not have training data for the second approach.

2.5 Our project

In the first three sections of this chapter, we have introduced some concepts and techniques on IR, IE and QA. Meanwhile, we also describe some QA methods adopted by the existing QA systems. Especially we give more details about the method based on NE because this method is more related to our project.

Our project aims to construct a QA system for the construction sector. It is a domain-specific QA system. We assume that all the documents in which we try to locate answers are related to construction. In our project, we first use an existing IR system - Okapi - for the basic passage retrieval. The techniques we will develop are either integrated into the Okapi indexing and search process, or used in a post-processing of the retrieval results. Our approach combines several existing methods described in the literature. First, we try to locate passages in the local text collection which may contain an answer to a question. If no satisfactory answer is identified, search is extended to the Web. As our QA system is specific to the field of construction and experts have already constructed a domain thesaurus, we can benefit from the thesaurus. This thesaurus will be deployed for query expansion, concept-based search as described earlier as well as in the post-processing. The new aspect of our approach is that we expand the common named entity concept to domain-specific named entity. A domain-specific NE is indeed a semantic category of concepts that is identified in

the thesaurus. We consider such categories as special types of NE, and questions can be asked on them.

For example, we will be able to deal with questions such as "what material is the most suitable to the constructions in the Northern areas of Quebec?", in which "material" is considered as a type of special NE. Notice that for open-domain QA, one can only ask question on common types of NE such as "what is the date of independence of the USA?".

In the next chapter, we will describe details of our approach.

Chapter 3

Our approach to domain-specific QA

In chapter II, we have introduced general QA problems and QA approaches. However, as we mentioned, the particularity of the QA system that we want to implement is that it is domain-specific. This means that we want to answer questions related to a specific domain, which is the construction area in our case. However, a domain-specific QA also involves a general-domain QA. Thus, we should not only solve the problems that appear in general-domain QA system, but also deal with problems that appear in domain-specific QA.

3.1 Problems in domain-specific QA

General QA systems focus on answering common sense questions. Namely, it tries to answer the questions whose answer types belong to common NE types, i.e., an NE type that is domain independant, such as date, person name, organization and so on. For example,

Question 1: "When was Trec-10 held?"

Question 2: "Who is the President of USA?"

The expected answer types for these two questions are DATE and PERSON respectively. These types are added into the questions so that the general QA system can return the precise answers for this kind of questions. However, sometimes, we also want to ask questions in a specialized area. For example, a professional in construction sector may ask the following question:

Question 3: "What materials are the best suited for houses in Montreal area?"

General QA system cannot determine an expected answer type for this question and have to adopt the general IR search. The problem is that NEs used in the previous research are general-domain NEs. They are not enough for dealing with domain-specific concepts and question types. To answer more complex questions than those on general NEs, one has to use more knowledge. However, because the world knowledge is infinite and no matter how large a knowledge base becomes, it is impossible to store all the concepts and technical terms for all domains. Even the largest knowledge base can only store a part of them. Clearly, no general QA can provide precise answers for professional questions in all the areas. Our approach tries to use domain-specific knowledge, which is more available than general knowledge.

In order to extend the general QA approach based on NE to a specialized area, the key is to extend general NE types to specialized NE types, so that questions can also be asked on the latter. Just as common NE types, specialized NE types are also types of (specialized) concepts. We use sometimes NEs to refer to them because the techniques we will use to deal with them is similar to those used for common NEs. In fact, they are semantic categories of concepts, such as "material" in Question 3. So we will also call the specialized NEs "categories".

In addition, in a specific domain, a lot of technical terms are compound terms. Traditionally, single words are used as indexes for the first-step passage selection with an IR system. This is not precise enough. The problem of compound terms is

also an important factor affecting the quality of QA systems. Thus, recognizing the compound terms is an important part in a domain-specific QA system.

In order to implement a domain-specific QA system, a domain knowledge base, or at least a thesaurus is necessary. In our case, we have a thesaurus in the construction sector. We will exploit it in our work. Namely, we will work out methods to extract categories and compound terms from the domain thesaurus. Further more, we will also determine the expected answer type of professional question in terms of categories organized in the thesaurus.

In the following sections, first, we will give an overview of our domain-specific QA system. Second, we will describe how to tag common NEs. It includes document tagging and question tagging. Third, we will represent how to tag domain-specific categories and compound terms by utilizing the thesaurus. Fourth, we will describe the search strategies for applying this semantic information. Fifth, we will summary how to deploy these methods and techniques into the system. Finally, we describe some details on our implementation.

3.2 Overview of the system

In our system, the approach that we have adopted is similar to the second method explained in chapter II, namely, the method based on named entity identification (see section 2.3.2.2). The reason of our choice is due to the simplicity of this approach and its effectiveness as reported by the previous experiments. In fact, this is the most commonly used approach in QA.

The system consists of some modules, each of which is an independent component. Figure 3.1 gives the workflow of the system.

- **Document Collection:** It downloads domain-specific documents from the Web with the assistance of a Webmaster. This step is used to establish a collection of texts. This step has been implemented in another MSc. project [Zha03].

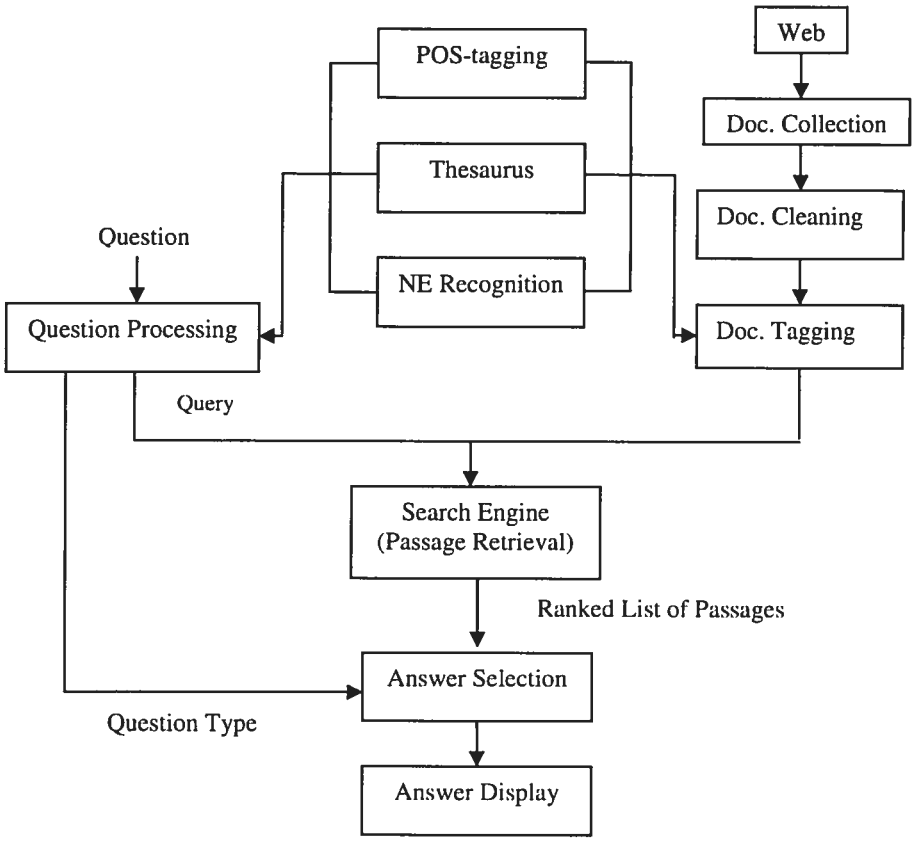


Figure 3.1: Workflow of the system

- **Documents Cleaning:** This module is used to transform the downloaded documents into a suitable form for our processing. In our system, we implemented structure recognition (such as title, passage and sentence), changing the format of documents (from HTML document to text document). As for the other process, such as the elimination of stopwords (such as articles and connectives), the use of stemming (which reduces distinct words to their common grammatical root), the identification of noun groups, they all are done by the Okapi search engine, or implemented in another project [Zha03].

- **Document Tagging:** In this module, we deal with the problems of extracting common named entities (e.g. person name, address, organization, etc.), some specialized named entities or semantic categories (for example, material, building, etc.), and compound terms (for example, winter concrete, etc.) in the

construction domain. The system takes a text, associated with some semantic information as input and produces as output a text containing more markers. This involves two essential tasks: Pos-tagging and NE recognition, which also require a thesaurus.

  - **NE Recognition:** This step aims to identify NEs in documents and questions through recognition rules for each NE class. In our system, the named entities that have been tagged include person, organization, location, date, time, season, percentage, monetary amounts and so on.
  - **POS-tagging:** This step aims to tag the Part-Of-Speech (POS) of each word in sentences. We used an existing package of POS-tagging from the RALI lab for this task. This is a statistical tagger.
  - **Thesaurus:** The identification of semantic categories and compound terms are based on a thesaurus. The principal advantage of using thesaurus is that we can obtain more semantic information of terms by means of hierarchical information and relationships between terms. The purpose of obtaining these semantic information is to deduce the silence rate due to the fact that a document doesn't mention the same concept as the one required by a question, but a related one or an implied one.

- **Question Processing:** It has two purposes. One is to form a query for the first selection of candidate answers. The formation of query will directly influence the recall and precision of the system. Thus, we need to pay attention on it. Another purpose is to determine the expected answer type. The expected answer type should be either a common NE or a domain-specific category that have been tagged in document processing. We deal with three question types in our study: Definition, common NE, and Category. For the questions that do not belong to these three question types, no post-processing is used for them.

- **Search Engine (Passage Retrieval):** Like most current QA systems, our system is also built on top of a retrieval system. An IR system (Okapi) was

employed to select a set of passages (paragraphs) that contain potential answers to the question. The retrieval system we use is the Okapi search engine (built at City University, London)[Oka]. Okapi search engine is not document-oriented but passage-oriented. In our case, a passage is a paragraph. What we have done is to first pre-process the documents to attach semantic information to the original documents, so that it is also indexed by Okapi search engine. In so doing, it is also possible to exploit this additional information during retrieval. For example, the expected NE type will be considered as an additional index (or keyword). So the candidate passages identified by Okapi will more likely contain an answer of this type. The top 50 passages [1] are returned by Okapi search engine in our system.

- **Answer Selection (Search Strategy):** We use the ranked list of passages containing the possible answer as the input of the answer selection module. At this stage, a special retrieval form is used, in which we consider not only the question keywords occurred in the passage, but also the tags that we added in the passage such as NE types, categories, and so on. The reason to do this is that we not only want the selected passages containing the required keywords, but also the required types of element (e.g., NEs, categories, etc.). This will avoid the retrieval of passages containing the required keywords (e.g., president, USA), but not the required answer (the answer to "who"). We then use some additional constraints to further verify if the passage contains an answer. One of them is that the candidate passages must contain at least one identical NE type or semantic category to the expected answer type. Here, the expected answer type is one or more named entities (e.g., person, organization, etc.), or are some extended named entities or categories (for example, building, material, etc.). According to the question type obtained from the question processing

[1] We did experiments and found that answers of 96% of questions appeared in the first 50 passages. In addition, if we chose more passages (more than 50 passages), the post-processing would need more time to deal with them and the expected improvement is small.

module, we work out three search strategies: Definition search, common NE search and Category search. For each search strategy, we will use different formula to compute the score of each sentence or passage to the question, and the passages are then re-ranked.

3.3 NE tagging

Named Entity tagging plays an important role in question answering system based on NE. The quality of NE tagging influences directly the performance of QA system.

We adopt a heuristic method for tagging NE as most of the other IE and QA system. The set of common named entity types that we have tagged is shown in Table 3.1. Some of them are further divided into subtypes. There are some differences in tagging document NE and question NE. So, we will describe them separately.

Notice that an existing package for NE tagging from [Gat] has been used in our group - RALI. However, when we started this project, the package was not yet available to us. Therefore, we constructed our own NE tagging tool following the approaches described in the literature.

3.3.1 Document tagging

It is usually believed that for many named-entity instances, both the spelling of the name and the context in which it appears are sufficient to determine its type [CS99]. Thus, the tagging approach we used is of two kinds. One is through word matching by using some gazetteers. Another one is to use some rules.

- Using Gazetteers

  The name Gazetteer originated from its use by English newspapers ("gazette") for its list of authoritative forms of place names. Now, the gazetteer concept has applications beyond the representation of places. This approach is mainly used for identifying PERSON and LOCATION types, which have fixed spelling.

Type	Subtype	Examples
ORGANIZATION		Educational Facilities Laboratories Inc.
PERSON		Mary Young, Prof. Smith
LOCATION	COUNTRY	Canada, China
	PROVINCE	Quebec, Ontario
	CITY	Montreal, Ottowa
TIME		2:30 pm, 7 o'clock
DATE		in 1999, Jun.1
ADDRESS		3000 Sand Hill Road, Building 1, Suite 120, Melon Park, California 94025
SEASON		spring,summer,autumn, winter
NUMBER	Number	20.3, 3004, eight
	TEMPERATURE	20 degree
	PERCENTAGE	90 percent, 50%
	MONEY	23 dollar, 25 cent

Table 3.1: Types for named entity annotation.

Here, a gazetteer is a list of geographic names (country, province, city and so on) or person names (family names, male first names and female first names).

Several gazetteers have been employed in our system. For the identification of person names, we used a gazetteer, which is the U.S. census list of the 15,024 most frequent last names, 4275 most frequent female first names, and 1219 most frequent male first names in the U.S.A. [Bur]. As for tagging cities, countries and provinces, we also found some gazetteers and used them as our tagging basis for these types [Lib02, Edu, Gaz]. Table 3.2 shows the numbers of entries in each gazetteer. The use of gazetteers for NEs tagging is simple. We only need to compare the input sentence with the entries of gazetteers. For example, we may have a gazetteer that stores "James Johnson" as a PERSON

and "Montreal" as a CITY. Thus, if "James Johnson" appears in a sentence, we can tag it as a PERSON. In the same way, "Montreal" can be tagged as a CITY. Clearly, if an NE is stored in such a gazetteer, it is easy to tag its occurrence in a document by a simple lookup into the gazetteer.

NE Type	Sub-Type	Number	Examples
PERSON	Family name	15024	SMITH, JOHNSON, WILLIAMS, ...
	Female name	4275	JAMES, JOHN, ROBERT, ...
	Male name	1219	MARY, PATRICIA, LINDA, ...
LOCATION	Country	243	Zambia (zm), United States (us), ...
	Province	12	British Columbia(BC), Quebec (PQ), ...
	City	374	Yellowknife, Woodstock, Waterloo, ...

Table 3.2: The numbers of entries in each gazetteer.

- Using Rules

  Using rules is another method for NE tagging. This approach is mainly used for identifying ORGANIZATION, NUMBER, DATE, TIME, PERCENTAGE, ADDRESS and TEMPERATURE types. For these types, it is impossible to store all the possible forms in a dictionary or a gazetteer. However, they usually follow some writing rules.

  In this approach, first, some rule expressions have to be defined to recognize the named entities. Then we analyse the words surrounding the feature word in the sentence and try to find more features of these words by feature word, which means a word that can determine the NE type of a word or a word sequence. Finally we compare these features with some rule expressions and check whether they match or not.

  In order to identify ORGANIZATION type, we compiled a list of feature words that occur frequently in ORGANIZATION type. For instance, '&' , 'Inc.' ,

'Ltd.' , 'Administration' , 'Department' , 'Committee' , and so on, are likely to be used within names of organization. In order to identify PERSON type, we also look for particular indicators for a person name, such as 'Mrs.' , 'President' , 'Dr.' and so on. Therefore, it is clear that we need to define some feature rules to identify important features. This feature information is integrated in rule expressions used for identifying named entity type. In our approach, we define the following types of feature:

- **FeatureInternalWord:** This feature is associated to the elements (characters, strings) that may appear within a type of word. For example, *NumString* is one of such features. It means Arabic numbers (1, 2, 3, ...). Other features in this category include: *NumString* (one, two, ..., hundred, thousand, million, ...), *NumLetter* (25th, 3rd, ...), *NumSymbol* (9:30, 09-08-2002, ...), *Uppercase* (A, B, ...), *Lowercase* (a, b, c, ...), *CapAll* (MR, LTD, ..., PEOPLE, ...), *CapFirst* (Li, John, ...), *StringSymbol* (part-of-speech, ...), ...

- **FeatureWordType:** This feature is associated to some special words corresponding to a special type. For example, *OrganizationSym* (..., Inc., Ltd., ...), *TitleSym* (Mrs., President, Dr., ...), *MonthSym*(January, February, Jan., Feb., ...), *FunctionalWords* (functional words are determiners and prepositions which typically appear in NEs, for example, a, an, the, of, in, ...), ...

The main difference between the two types is that the first type is more related to characters, while the second to complete words. Table 3.3 shows a part of such feature rules we created.

Non-terminal(Left)	Terminal and Non-terminal (Right)
NumString	0|1|2|3|4|5|6|7|8|9|
Delimiter	([" , "|" – "])" "+
Titlesym	$Mr\|Dr\|Prof\|President\|Sir\|Ms$
Uppercase	$A\|B\|C\|D\|E\|F\|G\|H\|I\|J\|K\|L\|M\|N\|O\|P\|Q\|R\|S\|T\|U\|V\|W\|X\|Y\|Z$
Lowercase	$a\|b\|c\|d\|e\|f\|g\|h\|i\|j\|k\|l\|m\|n\|o\|p\|q\|r\|s\|t\|u\|v\|w\|x\|y\|z$
Letter	$@Uppercase\|@Lowercase$
Word	$@Letter+$
CapFirst	$@Uppercase@Lowercase+$
Company	$Co.\|Corp.\|Company\|Inc.$
OrganizationSym	$Academy\|Administration\|Association\|Democratic\|$ $University\|Institute\|College\|@Company\|Fedaral\|$ $Municipal\|Democratic\|Christian\|Municipal$
FunctionalWords	$n\|on\|the\|a\|an\|of\|at\|null\|$
Year	$@NumString@NumString(@NumString@NumString)$
MonthSym	$January\|February\|March\|April\|May\|June\|July\|August\|$ $September\|October\|November\|December\|Jan.\|Feb.\|Mar.\|$ $Apr.\|May.\|Jun.\|Jul.\|Aug.\|Sep.\|Oct.\|Nov.\|Dec.$
NumLetter	$@NumRoman@Letter+$
NumRoman	$@NumString+$

Table 3.3: Feature rules.

Once these features are defined, we should now define rule expressions. We call these rules as tagging rules, which are used to determine NE types in an input string. Table 3.4 shows some of such tagging rules. We now show how the rules we define are used in NEs tagging.

Non-terminal(Left)	Terminal and Non-terminal(Right)	Examples
Person	$(@Titlesym(".")) @Uppercase@Word$ $(@uppercase@word)$	President Bush
Date	$@MonthSym@Delimiter@NumString$ $(@NumString)@Delimiter@Year\|$ $@MonthSym@Delimiter@NumLetter$ $@Delimiter@Year\|$ $@MonthSym@Delimiter@NumString$ $(@NumString)\|$ $@MonthSym@Delimiter@NumLetter\|$ $@Year" - "@NumString(@NumString)" - "$ $@NumString(@NumString)\|$ $@NumString(@NumString)" - "@NumString$ $(@NumString)" - "@Year$	Jan. 12, 1999 05-04-2000
Organization	$@CapFirst + @FunctionalWords$ $@CapFirst+$ $@OrganizationSym@FunctinalWords$ $@CapFirst + \| + @OrganizationSym$ $@FunctionalWords@CapFirst + \|$ $@CapFirst + @FunctionalWords@CapFirst$ $+@OrganizationSym$	Educational Inc.

Table 3.4: Tagging rules.

There are mainly two steps in NEs tagging by using rules. The first step consists of tagging the features of words by using feature rules. The second step is

to apply tagging rules for locating named entities. For example,

Sentence: "Mr. Li was working in Educational Facilities Laboratories Inc. on Feb. 3rd , 1999, in Canada. "

First, we tag the features of words by using feature rules, which are shown in Table 3.3. 'Mr.' is tagged as *TitleSym*; 'Li', 'Educational', 'Facilities', and 'Laboratories' are tagged as *CapFirst*; 'Inc.' is tagged as *OrganizationSym*; 'Feb.' is tagged as *MonthSym*; '3rd' is tagged as *NumLetter*; '1999' is tagged as *NumRoman*. By using gazetteer, "Canada" is tagged as COUNTRY directly.

Second, we locate the named entities in this sentence through using the tagging rules, which are represented in Table 3.4. 'Mr. Li' is tagged as a PERSON, because it starts with a title followed by a word with Capital letter. In a similar way, 'Educational Facilities Laboratories Inc.' is tagged as an ORGANIGATION, and 'Feb. 3rd, 1999' is tagged as a DATE.

Obviously, these techniques are rather simple and may be error-proned. However, their advantage is that they are simple to implement. They do not require sophisticated analysis, yet may cover a variety of common forms of NE of different types. It is why we chose to use them in our system.

It is to be noted that our system is in prototypical development stage. Our aim is not to develop a NE tagging that can produce the best results. Rather, our purpose is to implement the basic NE tagging mechanism for the most frequent NE in the construction area. Later on, the rules and the gazetteers can be enhanced, without the mechanism having to be modified. It is also to be noted that there are many kinds of methods for named entity annotation. More sophisticated systems usually use learning techniques for identifying named entities. These latter may be incorporated in our future work.

3.3.2 Question tagging

The main difference between question tagging and document NE tagging is that question NE tagging has to determine the expected answer type, which is a named entity type that specifies the type of the answer the user expects to obtain. This NE type is crucial in determining whether a sentence can be a possible answer. In our system, we only process simple questions involving one NE (e.g., " Who is ...?", " What material ...?" and so on.). We do not consider the cases that include more than one NE (such as, " Who and when did ...?").

For tagging expected answer type, we need to analyse many kinds of sentence patterns, especially, WH-question. Some WH-words can determine the question types directly, such as, "when", "where", "who", "whom", "why" and so on (See Table 3.5). But for other WH-words, like "what", "which", and word "how", syntactic and semantic analysis for questions are needed to determine the expected answer types for questions. The expected answer types that we will identify in our system are displayed in Table 3.1.

WH-word	Question Types
When	TIME, DATE
Who, Whom	PERSON
Where	LOCATION
Why	REASON
How much	MONEY

Table 3.5: Relationships between WH-words and question types

- WH-word matching

  Some WH-words can determine the question types directly. For example, if WH-word "where" appears in the head of question, we can determine the expected answer type for this question as LOCATION.

- Syntactic and semantic analysis

  To do syntactic and semantic analysis, POS-tagging for the question is necessary. After POS-tagging, we do a partial syntactic analysis in order to recognize the structure of the question. The following question structures are recognized:

  1. *What/Which* Noun(s)/Noun Phrase(s) ...? The noun or noun phrase right after a DETERMINER word "what or which" can often be used to determine expected answer type. We define this noun as identifying word, which will be further used to determine the question type. For a noun phrase, we select the head noun of it as identifying word. For example,

     Question 1: What (Which) department in Canada is in charge of registering earthquakes and seismic activity?

     In this example, the identifying word is "department".

  2. *What is/are* Noun(s)/Noun Phrase(s) ...? We select the first noun (for noun phrase, the head noun of this noun phrase is selected) as the identifying word. For example,

     Question 2: What is the address of ...?

     In Question 5, the identifying word is "address".

  3. *How many* Noun(s)/Noun Phrase(s) ...? The noun (for noun phrase, the last noun of this noun phrase is selected) after a word "many" is defined as identifying word. For example,

     Question3: How many degrees is it usually in winter in Montreal?

     In Question 6, the identifying word is "degree".

  4. *How* Adj. Verb. ...? The adjective after a word "how" is defined as identifying word. For example,

Question 4: How hot is it in summer in Montreal?

In this example, the identifying word is "hot".

After the syntactic analysis, some semantic analysis is needed to determine the question type. First, we manually establish a semantic lexical base for identifying the expected answer type. This semantic base currently covers 9 NE types and 86 identifying words. However, it is easy to add new ones to it. This semantic lexical base allows us to map the identifying word (which may be a Noun or an Adjective) of query to the expected answer type. The mapping is shown in Table 3.6. Then we can determine the expected answer type through mapping the identifying word obtained from syntactic analysis to its corresponding NE type. Some examples are shown in Table 3.7.

NE Types	Identifying Word
ORGANIZATION	Administration, department, committee, ...
LOCATION	Place, city, province, ...
TEMPERATURE	Degree, temperature, hot, ...
DATE	Year, month, day, ...
TIME	Time, minute, second, ...

Table 3.6: Semantic lexical bases

Syntax	NE types
What/Which institute ...	ORGANIZATION
How old ...	AGE
How many degrees ...	TEMPERATURE
What is the address of ...	ADDRESS

Table 3.7: Examples for determining expected answer types

Up to now, we have described our mechanism for general NE tagging. Our approach is inspired by the existing approaches described in the related work. So it

is very similar to some of them. In the remainder of this chapter, we will describe the part that is different from the existing approaches. In particular, we will exploit domain-specific resources, namely a thesaurus, to extend the existing QA approaches to domain-specific concepts.

In the next section, let us first describe the thesaurus we used. This thesaurus is at the centre of our domain-specific QA processing.

3.4 Thesaurus

As our QA system is to be used in the construction domain, it is helpful to apply some domain knowledge in answering professional questions. Especially for the terms that have special meanings (not their common meanings in general domains) in construction, it is necessary to exploit domain knowledge to recognize them. For example, with the term "concrete", its common meaning is "naming a real thing or class of things". However in the construction domain, its meaning is "a hard strong building material". In order to reduce ambiguity, we need to add some semantic information to this kind of terms. For the term "concrete", we add semantic information (category) "building material" to reduce its ambiguity. To do so, a construction thesaurus is adopted in our system.

A thesaurus is a lexical knowledge base. It encodes not only the conceptual vocabulary but also semantic relationships between concepts. In [SM83], thesaurus is defined as follows:

> A thesaurus provides a grouping, or classification, of the terms used in a given topic area into categories known as thesaurus classes. As in the manual indexing case, thesauri can be used for language normalization purposes in order to replace an uncontrolled vocabulary by the controlled thesaurus category identifiers. A thesaurus may broaden the vocabulary terms by addition of thesaurus class identifiers to the normal term lists, thereby enhancing the recall performance in retrieval. Alternatively the

> thesaurus class identifiers can replace the original term entries in the hope of improving recall and providing vocabulary normalization. When hierarchical relationships are supplied for the entries in a thesaurus in the form of 'broader' or 'narrower' terms, the indexing vocabulary can be 'expanded' in various directions by adding these broader or narrower terms, or certain related terms, as the case may be.

In our system, the thesaurus that we have utilized is the Canadian Thesaurus of Construction Science and Technology [DJGC95]. This thesaurus is a vast network of approximately 15,354 concepts with approximately 26,000 links between them. It describes domain-specific terms and their relationships. Terms are organized into 11 levels, from 0 to 10. An excerpt of the thesaurus used in our system is shown in Figure 3.2, where circles denote terms of the thesaurus and arrows denote relationship symbols. The detailed meanings of relationship symbols are explained in Table 3.8.

Symbol	Description	Level
UF	Used for	
BT	Broader term relationship	$n- > n-1$
NT	Narrower term relationship	$n- > n+1$
WT	Whole term relationship	$n- > n-1$
PT	Part term relationship	$n- > n+1$
RT	Related term relationship	$n- > n$
GT	General related term relationship	$n- > 0$

Table 3.8: Relationships between terms in domain thesaurus

The thesaurus is composed of two parts; Part one represents the concepts or terms, Part two represents the relationships between terms. Some of terms are single words; the others are compound terms.The thesaurus is saved as a tree structure. Each term is a node of this tree. In Part one, each node has some attributes, such as, ID, English term, French term, Level. The higher the level is (the highest level is 0), the broader

the scope of the term is. In Part two, seven relationships are defined (See Table 3.8). We will use these relationships for acquiring and exploiting semantic information for defining domain categories.

The original thesaurus is in text format, which is difficult to use directly. In order to easily interact with the thesaurus, we transformed the thesaurus that is originally in text format into a MySQL database so that we can use SQL language to access it.

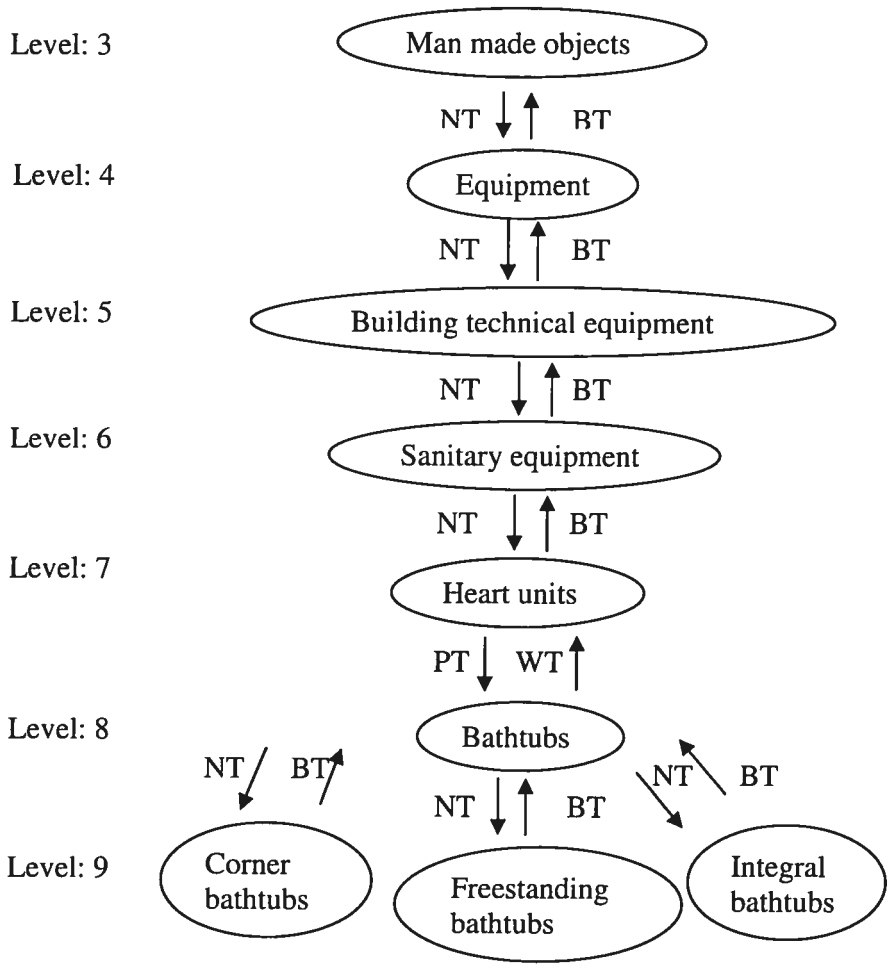


Figure 3.2: An excerpt of the thesaurus

3.5 Category(Domain-specific NE)

In the last section, we have described why we need a thesaurus in a domain-specific QA system and what kind of thesaurus we have in our system. In this section, we will describe how to make use of this thesaurus. Here, the method that we have employed in our system aims to assign dynamically a category for each term contained in thesaurus. Then, we tagged categories in questions and documents. This means that we add much semantic information into them. Therefore, the searching is not only based on keyword search but also based on concept search to some extent because the concept categories are also used as indexes, the user's query is reformulated in terms of categories, and we also give a category-based search strategy to match the user's category query. The category search is more precise than simple keyword search.

The strategy for assigning category to each term in the thesaurus is of great importance. If the choice of category is not reasonable, it will not improve the performance of system. Instead, it may worsen the performance of system.

3.5.1 Fixed categories

In our study, at first, we adopted a strategy of fixed categories. We chose about eighty terms, which have been recommended by domain experts as the most important categories of concepts. In their recommendation, domain-specific categories at level 4 are recommended, while non-domain-specific categories are set at level 3. Table 3.9 shows some examples of the recommended categories and their levels.

Level	Examples
Level 3	physics, commerce, chemistry, social life, economics, individual, living organism, physical geography, fluid mechanics, ...
Level 4	building process, manufacturing process, constructin, material, building economics, civil engineering work, equipment, ...

Table 3.9: Examples for determining expected answer types

We observed several problems with fixed cateories: First, their coverage is not large enough: just these 80 categories can't cover all the terms in the thesaurus. Some terms are not included in any of these categories. Second, these categories are from levels 3 or 4. The scope of these categories is usually too broad. Very different concepts may be tagged with the same category. For example, "corner bathtubs" is "equipment", so is "automobile". These concepts have very different meanings in construction. In order to recognize the finer semantic category of concepts in a more defined way, we need to refine the semantic categories.

3.5.2 Dynamic categories

In order to avoid the above problems, we defined a dynamic category assigning strategy according to the relationships and levels of concepts in the thesaurus. In this method, we assign the direct parent of a term in documents as its category. For example, the category "building material" is assigned to the term "concrete". In this case, it is clear that "building material" is a more suitable category for the term "concrete" than "material", which was obtained by using a strategy of fixed categories. The way of determining category for the term appeared in documents collection and user's question is different. We will represent them in the following parts respectively.

3.5.3 Tagging categories in documents

In document processing, tagging document category mainly depends on the semantic information of the thesaurus. The method that we used is as follows: for a term appearing in the thesaurus, we assign the direct parent of this term as its category. For the root node, its category is itself.

For example, we want to assign categories for terms "bathtubs", "corner bathtubs", "Integral bathtubs", and "freestanding bathtubs". Figure 3.2 shows that "heart units" is the direct parent of "bathtubs", thus, its category is "heart units". In a similar way, we define "bathtubs" is the category of the terms "corner bath-

tubs", "Integral bathtubs", and "freestanding bathtubs".

The reason for tagging categories in documents in this way is that we want a passage containing "corner bathtubs" can be considered as a possible answer to the question of "What bathtubs do you want to put in your bathroom ?" Therefore, when "bathtub" in this question is used as the category to look for, we can locate the appropriate passages containing concepts of the lower level.

Of course, we can use this principle further, for example, by allowing this reasonning to several levels of concepts. However, this will also increase the risk of finding remotely related concepts as in the case of fixed categories. So we only use the reasonning to one level in our current implementation. This can be changed later.

3.5.4 Tagging categories in questions

For user's question, part-of-speech of each word of question is first tagged. Usually, we define the head Noun as identifying word. If the identifying word is not included in the thesaurus, the category of the identifying word is Null. We don't process this case because what we have done on category are based on thesaurus. Thus, no further QA verification is possible, and we only return the IR results to the user. In our experiments, this case occurs 18 times out of 100 questions. If it is in thesaurus, we will give a method for finding a category based on thesaurus for the identifying word.

The method for tagging question category is as follows: for the identifying word appearing in the thesaurus, we define themselves as their categories except the terms that don't contain any sub-term (leaf node). We define the category of a leaf node as Null. For example,

Question 1: What bathtubs do you want to put in your bathroom?

First, we will determine the identifying word in Question 1, i.e., "bathtubs". Then, we find that it is not a leaf node from the structure described in Figure 3.2. Thus, the category of Question 1 is "bathtubs".

In last section, we define "bathtubs" is the category of the terms in documents

such as "corner bathtubs", "Integral bathtubs", and "freestanding bathtubs". In this case, if we submit Question 1 to Okapi, the passages that contain these terms will be regarded as containing the same category with which the Question 1 requires during the category-based search. In so doing, we will be able to identify the passages that contain one such implying concept, thus broaden the coverage of the retrieval.

3.6 Compound terms

Compound terms are composed of two or more single words. Usually, the meaning of a compound term cannot be fully expressed by the separate single words composing it. For example, "winter concrete" is a domain-specific compound term in construction. The single words "winter" and "concrete" cannot represent completely the meaning of "winter concrete". Therefore, we need to identify "winter concrete" as a single concept. It is better to keep compound terms without breaking them into words. The consideration of compound terms could reduce the ambiguity of specialized terms and enhance the precision of the system.

For domain-specific compound terms, we extract them based on the thesaurus, which contains a set of compound terms. Some common compound terms can be found from the gazetteers. For example, "United States" and "Hong Kong" are common compound terms that are stored in one of the gazetteers. Below, we will give more details on how to extract compound terms by using the thesaurus.

The following steps are carried out for finding compound terms.

1. For a word sequence w_1, w_2, ..., w_n.
2. Send a SQL request to the thesaurus to find all the compound terms starting with the first word w_1.
3. If a compound term corresponds to the part of the word sequence w_1, w_2, ..., w_i then w_1, w_2, ..., w_i is marked as a compound term.
4. Check the following word w_2 (repeat step 2, 3) until the word w_n.

For example, suppose a sentence: "window panes is subject to ... "

First, all the compound terms that starts with the word "window" are found from the thesaurus:

Window glass ,	Window eyebrows,	Window lights,
Window mullions,	Window shades,	Window transoms,
Window walls,	Window heads,	Window piers,
Window opening types,	Window panes,	...

Then, we select the compound terms that match the sequence of words in the input sentence. In this example "window panes" is recognized as a compound term – "window panes". Then the process continues on the next word "pane". Notice that two compound terms may overlap.

3.7 Search strategy

In this section, first, we will describe the question types and their identification in our system. Then, we will give the corresponding search strategy for each question type. The search strategies for general-domain QA system cannot be used for domain-specific QA system completely. Therefore, we developed our own search strategies for different question types.

3.7.1 Question type

We identified four question types: Definition, Named Entity, Category and Keyword question types. For a question, if its answer is a statement of the meaning of a word or word group, we define this question as Definition question type; if its answer should include a NE type, we define this question as Named Entity question type; if its answer should include a category, we define this question as Category question type; if this question does not belong to the first three question types, we define it as Keyword question type.

During question processing, the questions themselves were POS-tagged, morphologically normalized, and partial parsed. In addition, for identifying definition question type, pattern matching is applied.

The steps for finding question type are as follows:

First, pattern matching for identifying definition type (see Table 3.10) is attempted for question. If the question corresponds to the definition template, then the question type is Definition.

If the first step fails, we check the NE type of the expected answer type. If it is not Null, the question type is the NE.

If the NE is Null in the last step, we check the category type of the expected answer type (see section 3.3.2). If the category is not Null, the question type is that category.

In some cases question processing may fail to identify question type. This question will belong to Keyword type. In this case, no special post-processing for QA is possible and the IR results will be directly shown to the user. The percentage of this case will be reported in section 4.1.

The information of each question obtained through question processing will be used in the post-processing.

Non-terminal(Left)	Terminal and Non-terminal (Right)
Definition	what Verb Askingpoint|what Verb Adj Askingpoint| what Verb Dert Adj Askingpoint|...
Verb	is|are|was|were|mean|means|meant|define|defines|defined|...
Askingpoint	noun|noun phrase
Adj	any word that its POS is adjective
Dert	a|an|the

Table 3.10: Definition template (Question)

We use an IR system (Okapi) as the first filter to select the inputs to our QA processes. The principal advantage of using IR system first is that post-processing can

concentrate on the information extraction task to find the answers from a relatively limited quantity of text. At first, when a question is submitted to our system, a ranked list of passages possibly containing the best answers will be retrieved. After getting question type and 50 candidate ranked passages, the post-processing of identifying right answer starts. In the next section, we will describe how the post-processing is carried out.

3.7.2 Answer selection

In the post-processing, all the information tagged in the passage and in the question is used. Through question type, we can determine an appropriate search strategy for it. Different types of question require different formulas, patterns that may be matched in identifying possible answer location and weight assignment methods. This fact has been observed through our analysis of experimental results: we found that the factors and weights affecting search performance are different for different question types. We can't use the same formula for processing all kinds of questions. We propose an approach in which different types of questions are processed using different formulas. Figure 3.3 shows how many different types of question are evaluated. As we mentioned, in our system, questions are divided into four types: Definition, Category, NE, and Keyword. Each type uses a different search strategy.

We work out an evaluation formula for each search strategy. The parameters of each formula are discovered by a variety of heuristics. First, we select a set of empirical feature factors. These feature factors can be used for determining whether a given sentence or passage contains a precise answer to a question. Then, we made some experiments for testing which factors should be retained as parameters for each search strategy. This set of feature factors is different for each search strategy. The coefficients of each formula are also determined by experimental data. Finally, an evaluation formula combining different factors is used for calculating a final score of each sentence or each passage. We will explain in more detail each search strategy below.

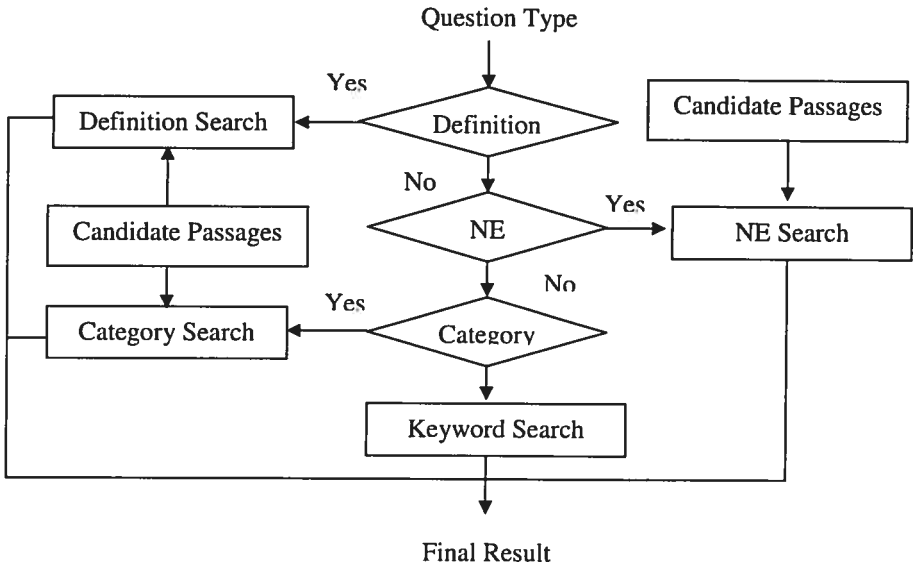


Figure 3.3: Answer selection

3.7.2.1 Definition search strategy

In Definition question type, we use a template for further locating candidate answer. Table 3.11 shows some of the templates. For each passage, we calculate a score of the passage using heuristics such as the length of each sentence(num_i), the number of sentences in one passage (N), keyword's position ($numBefore_i$, $numBetween_i$, $numAfter_i$) and each passage's original score returned by Okapi search engine ($Weight$). These heuristics are used due to our following observation:

1 A definition sentence usually includes a verb characterizing a definition.

2 A definition sentence usually starts with the concept, which is named Askingpoint, to be defined.

For example, a Definition question may be: "what is corrosion?". In this question, its Askingpoint is "corrosion", the characterizing verb for the definition type is "is".

The preferred structure of an answer to a definition question is that it contains a characterizing verb for definition, the concept to be defined appears at the beginning of the sentence, and there is a sufficiently long string of words after the characterizing

Non-terminal(Left)	Terminal and Non-terminal (Right)
Definition	NonVerbword Askingpoint NonVerbword Verb @Word
Verb	is|are|was|were|mean|means|meant|define|defines|defined|...
Askingpoint	noun|noun phrase
NonVerbword	any word except Verb
Lowercase	a|b|c|d|e|f|g|h|i|j|k|l|m|n|o|p|q|r|s|t|u|v|w|x|y|z
Uppercase	A|B|C|D|E|F|G|H|I|J|K|L|M|N|O|P|Q|R|S|T|U|V|W|X|Y|Z
Letter	@Uppercase|@Lowercase
Word	@Letter+

Table 3.11: Definition template (Answer)

verb. For example, "is", "means", etc. For example, the answer to the above question is as follows:

> Corrosion of metals is an electrochemical process in which the deteriorating area of the metal is the anode, the positively charged electrode of the galvanic cell.

We see that the Askingpoint appears early in the sentence, and there is a long string of words after the characterizing verb "is".

Below, we will give more details about the formula we will use for this type. Let us define the parameters as follows:

- $\boldsymbol{num_i}$ **:** the number of words in the i^{th} sentence.
- ***N:*** the number of sentences in one passage. For definition question, just one sentence usually cannot give a clear and complete explanation and it needs several sentences for explaining one concept. In addition, the right answer displayed to user is in passage format instead of one sentence in our system. So, this variable is necessary.

- ***Weight:*** This is a weight for passage derived from the original score returned by Okapi. This variable is used for those passages that don't contain definition answer pattern or candidate answer sentence. In this case, we still use the original score for this kind of passage's ranking. However, we didn't use original value directly, instead, we normalize this value first, i.e., divided by the maximum score. Thus, the value of this Weight is within the range of $0 < Weight \leq 1$.

- $\boldsymbol{numBefore_i}$ **:** the number of words before Askingpoint in the i^{th} sentence. As we mentioned, Askingpoint is determined by question processing. Usually, it is a Noun or Noun Phrase. For this variable, we also place a restriction on it: "$DefiWeight = \alpha * Weight, \alpha = 0.2$, if $numBefore_i > 3$". This restriction means that if there are too many modifications for Askingpoint, then the modifications have possibly changed the meaning of the Askingpoint. So, it is less likely that it is a suitable definition of the Askingpoint concept. In this case, we calculate the score of this sentence according to the variant of original score returned by Okapi search engine, namely, "$DefiWeight = \alpha * Weight, \alpha = 0.2,$". The coefficient ($\alpha$) is tuned by experiments.

- $\boldsymbol{numBetween_i}$ **:** the number of words between Askingpoint and Verb in the i^{th} sentence. If there is no word between Askingpoint and Verb, or the part-of-speech of these words is Adverb, $numberBetween_i = 0$. If the value of this variable in one sentence is large, then, the possibility that this sentence belongs to a suitable answer is small. If there is no word before Askingpoint and no word between Askingpoint and Verb, we assign $numBefore_i + numBetween_i = 1$. At this moment, the first item in the definition search formula gets the maximum value. This is the ideal case for definition answer pattern.

- $\boldsymbol{numAfter_i}$ **:** the number of words after Verb in the i^{th} sentence. In order to avoid selecting a too short sentence, we also place a restriction on it. It is "$DefiWeight = \alpha * Weight, \alpha = 0.2$, if $numAfter_i < 6$". If the value of

dividing $numAfter_i$ by num_i in $sentence_i$ is large, then the possibility that $sentence_i$ belongs to a right answer is also large.

During our implementation, a number of formulas have been tested based on the factors just mentioned. The following formula produces the best results among these tested:

$$DefiWeight = \begin{cases} \sum_{i=1}^{N}(\frac{1}{numBefore_i + numBetween_i} + \frac{numAfter_i}{num_i}) & \text{if } numBefore_i < 4, \text{ and } numAfter_i > 4. \\ \alpha * Weight, & \text{otherwise} \end{cases}$$

The formula combines different heuristic factors. Further experimental results show the final score calculated by this formula can successfully re-rank the candidate passages into a better list. We chose $\alpha = 0.2$ in our system according to the experimental results.

Here is an example, which shows that our post-processing for definition type can improve the results of the system. Suppose the question "What is corrosion?". Its AskingPoint is corrosion. The first passage returned by using Keyword search with Okapi is the following one:

> A variety of metals are used in building in many different ways. It is for this reason that the problems of **corrosion** in buildings cover a very wide range. In this brief article only an outline or classification of the main problems can be given, along with the basic principles, to guide the designer in his efforts to reduce the huge economic loss caused by **corrosion**. For specific information on the practical problems of **corrosion** the reader is directed to the extensive work of the various **corrosion** committees of the ASTM and of the British Iron and Steel Research Association. The National Association of **Corrosion** Engineers has published the results of much research in the field of **corrosion**.

This paragraph does not contain sentences that likely give a definition of "corrosion". After one post-processing, the following passage is re-ranked at the first place.

> **Corrosion** of metals is an electrochemical process in which the deteriorating area of the metal is the anode, the positively charged electrode of the galvanic cell. Positive potential of the metal indicates **corrosion** activity, i.e., the metal in this region is converting from the metallic to the ionic state. The value of the potential depends on the tendency of the metal to go into solution and, based on the concentration of ions around the electrode, is a good measure of the **corrosion** that has taken place.

The first sentence in this passage corresponds to a good structure of a definition: the key concept or Askingpoint occurs at the beginning of the sentence, followed by a verb characterizing a definition and a long sequence of words. Therefore, the global score of this passage is increased, and the passage is ranked higher. This is the correct passage that the user looks for.

3.7.2.2 NE search strategy

In NE search strategy, we take into account heuristics such as the number of matching words (*numMatchWord*), the number of named entity matching (*numMatchNE*), each passage's original score returned by Okapi search engine (*Weight*), n-gram in sentence *(numN-gram)*, as well as the length of candidate sentence (*numWord*). The following, we will give more explanations about these parameters for NE search strategy.

- ***Weight*** : This is a weight derived from the original score of Okapi. It is determined in the same way as the *Weight* in Definition type strategy.
- ***numMatchNE*** : the number of NE occurred in both question and answer candidate sentence at the same time. It is the key for NE search. If *numMatchNE* in both sentence and question is equal to zero, i.e., the question's expected answer type doesn't appear in this sentence, this sentence can't become a precise answer of this question. In this case, we don't need to do more analysis for this sentence and just assign weight for it. Otherwise, we will assign a weight to *numMatchNE*. This assignment is subject to the following parameter – *numMatchWord*.

- ***numMatchWord*** **:** the number of keywords occurring in both question and answer candidate at the same time. We use this parameter for scaling *numMatchNE's* weight assignment. In one sentence, if *numMatchNE* is greater than 0 and *numMatchWord* is also greater than 2 [2], we assign a higher weight for *numMatchNE.* [3] Otherwise, the weight of *numMatchNE* is zero. For example, if the question is "what is the address of ...?", then, if a sentence is tagged with the named entity "ADDRESS" and *numMatchWord* in this sentence is also greater than 2, this sentence will be assigned a higher score.

- ***numN − gram*** **:** the number of bi-gram, and tri-gram occurred in both question and candidate answer sentence. It can contribute more confidences for finding precise answer for user's question.

The calculation formula for NE search is as follows:

$$NeWeight = \begin{cases} numMatchWord+ & \text{15.0*numMatchNE+numN-gram} \\ & \text{if } numMatchNE > 0, \text{ and } numMatchWord > 2. \\ \alpha * Weight, & \text{otherwise} \end{cases}$$

It is a linear combination of these different heuristic factors. We chose $\alpha = 1.0$ in our system.

3.7.2.3 Category search strategy

In category search strategy, we take into account heuristics such as the number of matching words (*numMatchWord*), the number of matching Category (*numMatchThesaurus*), each passage's original score returned by Okapi search engine (*Weight*), n-

[2] if *numMatchWord* is smaller than 2, we cannot ensure that the sentence and question are relevant. Thus, we set this restriction.

[3] In our system, we set 15 as a coefficient for parameter *numMatchNE.* It came from experimental results.

gram in one sentence (*numN-gram*), as well as the length of sentence (*numWord*). Below we will explain why we choose these parameters for Category search strategy.

- ***Weight*** : This is the same weight derived from Okapi as before.
- ***numMatchThesaurus*** : the number of categories occurring in both question and answer candidate sentence at the same time. Its role is similar to *numMatchNE*.
- ***numWord*** : the number of words in the candidate sentence. In order to balance the probability for long or short sentences, we add some restrictions on this parameter. If *numWord* is smaller than 15, we set *numWord* to 15. If *numWord* is greater than 15 and smaller than 20, we keep its real value; if *numWord* is greater than 20, we set $numWord = 20 + (numWord - 20)/15.0$. This setting is to reduce the impact of length differences on the final weight (see the formula given below).
- ***numN − gram*** : the number of bi-gram, and tri-gram occurring in both question and candidate answer. It can contribute more confidences for finding precise answer for user's question.
- $\boldsymbol{\alpha, \beta}$ the final score of each candidate passage is a combination of Keyword search's score and Category search's score. α is the weight of keyword search's score– *Weight*, β is the weight of Category search's score. We set $\alpha + \beta = 1$. Here, we need consider the question: how to assign different weight to Keyword search and Category search? We have varied the weight of Keyword search and Category search in a series of experiments. Finally, we determine that α=0.3 and β=0.7 is a good combination.

The calculation formula for Category search is as follows:

$$CateWeight = \begin{cases} \alpha * Weight + \beta * \frac{2*numMatchThesaurus+numN-gram}{numWord} \\ \alpha + \beta = 1, \end{cases}$$

It is a linear combination of these different heuristic factors. We chose $\alpha = 0.3$, $\beta = 0.7$ in our system according to the experimental results.

3.8 Integration

Up to now, we have described the methods for tagging common NEs, domain-specific categories and compound terms and some strategies for retrieval. In this section, we will describe how to integrate these methods and strategies into our system. These techniques are used in document processing and question processing.

3.8.1 Document processing

In document processing, it is necessary to carry out the following processes: (a) the cleaning of documents (removing HTML markers), (b) the operations of annotation of the document collection, (e.g., extracting named entities, Categories, compound terms and tagging part-of-speech) The workflow of all the operations is shown in Figure 3.4.

After passing document processing step, some additional markers (e.g., $< ADDPHRASE >$, $< ADDNE >$ and so on.) are added into the documents collection to tag the semantic information explicitly. Figure 3.5 shows some examples of document processing. $< ADDPHRASE >$ contains compound terms that are recognized during this process. $< ADDNE >$ contains the NEs recognized. The numbers after each NE correspond to its beginning and ending positions in the sentence. $< ADDCATEGORY >$ contains the categories recognized. The number after each category also corresponds to its position.

3.8.2 Question processing

In question processing, we implement the following two functions:

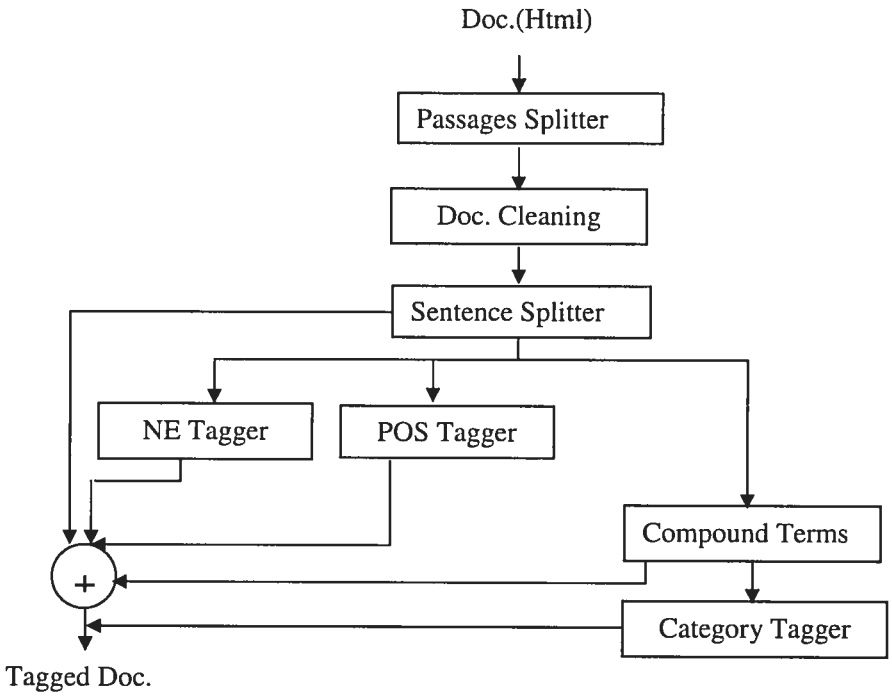


Figure 3.4: Document processing

1. to generate query that is submitted to Okapi for the passage retrieval to identify the best candidate passages.

2. to identify the type of question (i.e., Definition, Category, Named Entity, and Keyword in our system) so that the post-processing can determine the corresponding search strategy for finding the best answer from the passages.

For doing these, the question themselves were part-of-speech tagged, morphologically normalized, and partially parsed. For definition question type, pattern matching is applied. The workflow for question processing is shown in Figure 3.6.

Below are some examples.

Question 1: What is corrosion?

Keyword: corrosion

Compound terms: Null

<RD:7844>-
-
<ADDRESS> http://www.nrc.ca/irc/cbd/cbd209e.html </ADDRESS>
<SENTENCE> A simple energy analysis computer program is used to predict the approximate potential for fuel and cost savings.
<ADD PHRASE> computer_program
<ADD CATEGORY> data_processing_systems <4> costs <16> computer_programs <18>
<SENTENCE> The calculation is based on readily obtainable information about the school, its heating and ventilating plant, operation and fuel consumption.
<ADD PHRASE> fuel_consumption.
<SENTENCE> The service is offered by Educational Facilities Laboratories Inc., (3000 Sand Hill Road, Building 1, Suite 120, Menlo Park, California 94025) and costs between $60 and $90 per school building.
<ADD PHRASE> Educational_Facilities school_building.
<ADD NE> {{ORGANIZATION}} <5, 8> {{NUMBER}} <15> {{NUMBER}} <17> {{PROVINCE}} <20> {{ADDRESS}} <10, 21> {{MONEY}} <25> {{MONEY}} <27>
<ADD CATEGORY> facilities <6> laboratories <7> sand <11> landforms <12> buildings <14> costs <23> schools <29> educational_facilities <31>
-

Figure 3.5: Examples for documents processing

Matching definition pattern: Yes

Thus, the query of this question sent to Okapi is: "corrosion"

The question type of Question 1 is "Definition", and its corresponding search strategy is "Definition search" since Question 1 matches the definition pattern.

Question 2: What organization in Canada is in charge of registering earthquakes and seismic activity?

Keywords: organization, Canada, charge, register, earthquakes, seismic, activity

Compound terms: Null

Matching definition pattern: No

NE type: ORGANIZATION

Thus, the query of this question sent to Okapi is:

"organization + Canada + charge + register+ earthquakes+ seismic +activity+ ORGANIZATION"

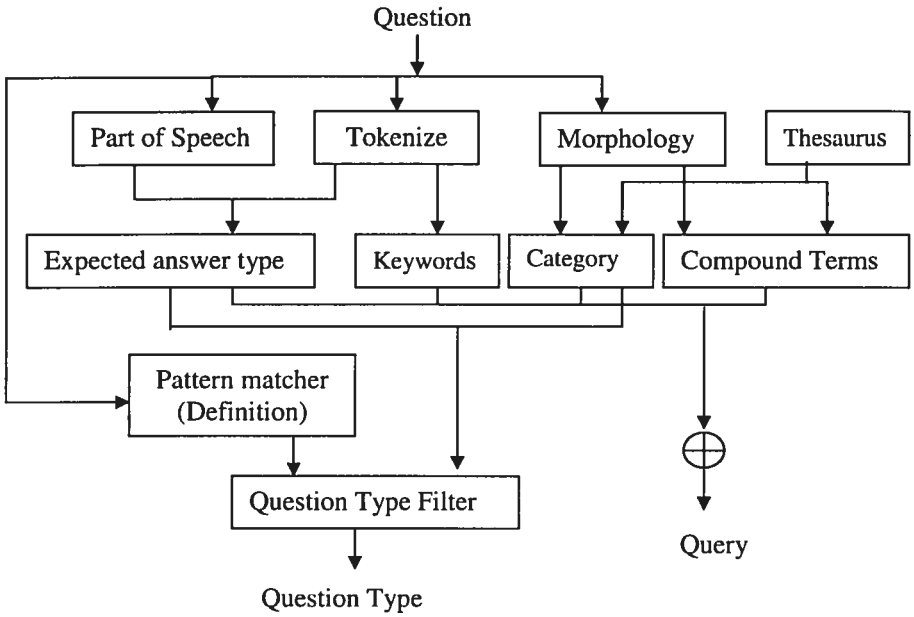


Figure 3.6: Question processing

The question type of Question 2 is "Named Entity", and its corresponding search strategy is "NE search" since Question 2 doesn't match the definition pattern and NE type is ORGANIZATION.

Question 3: What are the common thermoset foams used in frame construction?

Keywords: thermoset, foams, frame, construction

Compound terms: frame_construction

Matching definition pattern: No

NE type: Null

Category type: product_forms

Thus, the query of this question sent to Okapi is:

"thermoset + foams + frame + construction+ frame_construction+ product_forms"

Because Question 3 doesn't match the definition pattern, and its NE type is Null, and we can find a Category type for it. Thus, the question type of Question 3 is

"Category", and its corresponding search strategy is "Category search".

Question 4: What are the methods for determining pressure rating?

Keywords: methods, determine, pressure, rate

Compound terms: Null

Matching definition pattern: No

NE type: Null

Category type: Null

Thus, the query of this question sent to Okapi is:

"methods + determine + pressure +rate"

The question type of Question 4 is "Keyword", and its corresponding search strategy is "Keyword search" since Question 4 doesn't match the definition pattern and its NE type is Null and its Category type is Null.

3.9 Implementation

The system is constructed in different modules. Each module fulfils a task separately. In this section, we will give more details about our implementation.

3.9.1 Architecture

The system is implemented in Linux operating system, and programming languages are Java and C++. For extracting semantic information from thesaurus, MySQL database is used. For connecting the system into Internet, web-developing tools – Tomcat and Servlet are concerned. In order to ensure that the system can concentrate on the information extraction task for finding answers from a relatively limited quantity of text, we use an IR system (Okapi) as the first filter to select a set of passages as input to our system.

Okapi (Online Keyword Access to Public Information) is developed by the Polytechnic of Central London, now Westminster University, in 1982 and continued at the City University from 1989. More information about Okapi is given in [Okapi].

At present, most search engine returned a ranked list of documents with no indication of relevant passages within the document. Okapi search engine is not document-oriented but passage-oriented, where a passage is a paragraph. This corresponds well to our requirement. In addition, Okapi has shown very good performance in TREC experiments. This is why we chose Okapi as search engine in our system.

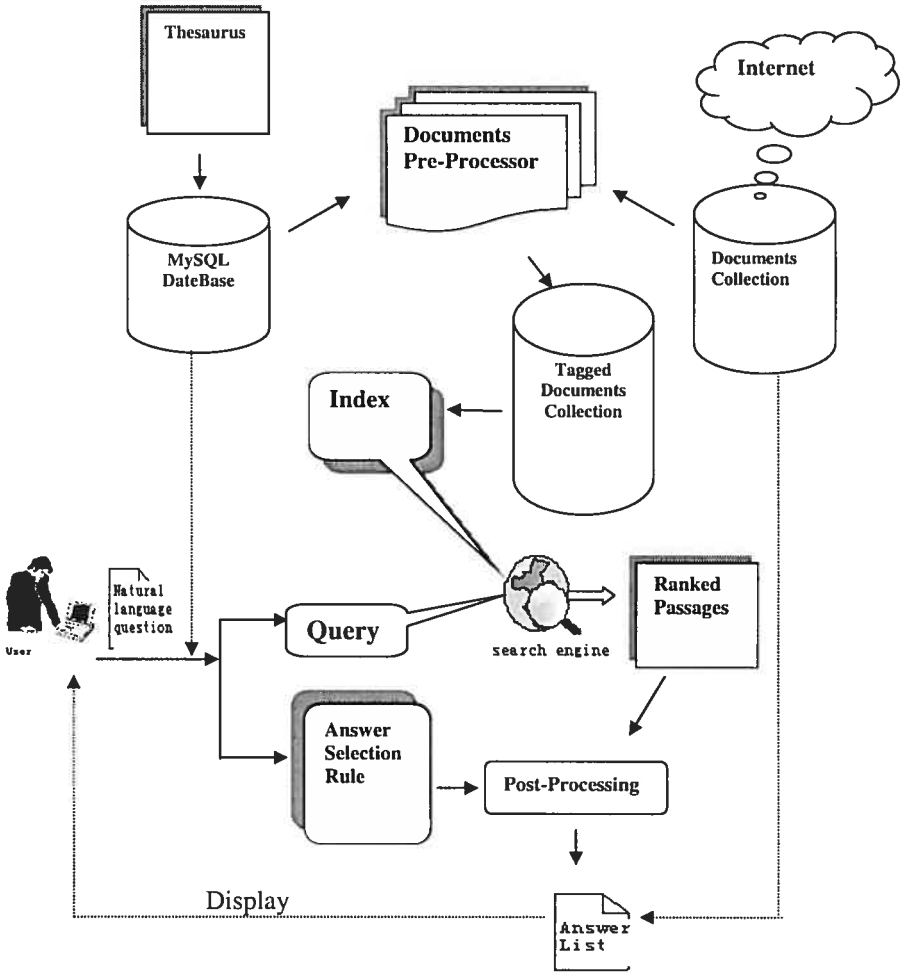


Figure 3.7: Architecture

Notice that before using Okapi for indexing, all the documents (and questions) have been analyzed so that annotations have been added. These annotations will also

be used as indexes. Figure 3.7 presents the architecture of the system.

3.9.2 Package source

We need to perform syntactic analysis of texts and questions. Thus a POS-tagger is necessary for us. We adopted an existing tagger from the RALI laboratory in University of Montreal. This tagger is implemented in C++ programing language. Because the other parts of the system were implemented in Java program language, we had to use JNI method in the system to call C++ program in a Java program.

The Java Native Interface (JNI) is the native programming interface for Java that is part of the JDK. The JNI allows Java code that runs within a Java Virtual Machine (VM) to operate with applications and libraries written in other languages, such as C, C++, and assembly.

The JNI serves as the glue between Java and native applications. Figure 3.8 shows how the JNI ties the C++ side of an application to the Java side.

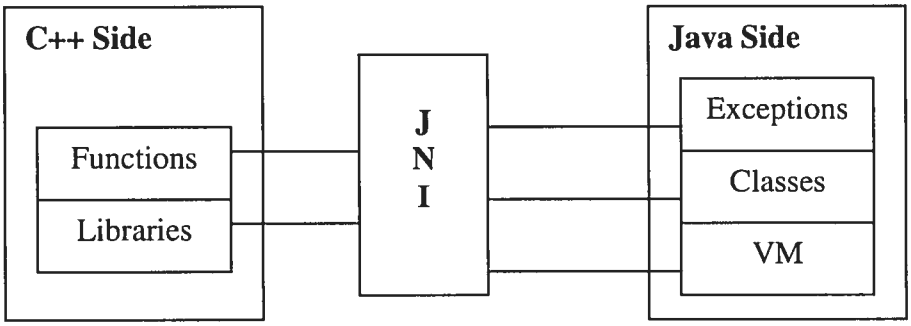


Figure 3.8: JNI application

3.9.3 Database

Our system contains a specialized thesaurus. The original thesaurus (see section 3.4) is in text format, which is difficult to use directly. In order to easily interact with the thesaurus, we transformed the thesaurus into a database MySQL. Below, we will show how the database MySQL is created and how it is used in our system.

MySQL is an open source database, recognized for its speed and reliability. It is the most widely used SQL database on the Internet. In short, MySQL is very fast, secure, reliable, and easy to use (for more details see [Mys]).

Based on the structure of thesaurus, we establish two tables for it. They are named "Table Thesaurus" and "Table Liens" respectively. In Table Thesaurus, there are five items. In Table Liens, there are four items. Some examples are shown in Table 3.12 and Table 3.13. The structures of Table Thesaurus and Table Liens are shown in Table 3.14 and Table 3.15. They display information about the Fields of the Tables.

id	frenchword(fword)	englishword(eword)	level
...	...	...	...
15354	Thesaurus	Thesaurus	0
151	Activité	Action	1
9722	Environnement physique	Physical environment	2
...	...	...	...

Table 3.12: Table thesaurus

id1	id2	relationship
...	...	...
9	563	RT
11	1870	WT
12	13769	WT
...	...	...

Table 3.13: Table liens

Once the database is created, we utilize JDBC to connect MySQL database and Java program. Then, we use Structure Query Language (SQL) to access the database. Our access is used to obtain the terms related to a given term by a given relationship. For example, we want to find all terms that have "NT" relationship with term

Field	Type	Null	Key	Default
id	Integer	Yes	Yes	Null
englishword	String	Yes		Null
frenchword	String	Yes		Null
level	Integer	Yes		Null

Table 3.14: Structure of table thesaurus

Field	Type	Null	Key	Default
id1	Integer	Yes		Null
id2	Integer	Yes		Null
relationship	String	Yes		Null

Table 3.15: Structure of table liens

"equipment". The SQL format is as follows:

SELECT "eword"

FROM "Thesaurus", "Liens"

WHERE id = "ID(equipment)" AND relationship =" NT".

The outputs are:

audiovisual equipment,
major domestic appliances,
equipment(tools),
maintenance equipment,
office equipment,
site equipment,
furniture,
handing equipment,
recreation equipment,
building technical equipment,
observing instruments,
factory equipment,
measuring instruments,
quarrying equipment,
testing equipment,
engines,
mining equipment,
transportation modes

3.9.4 Interface

We use the Tomcat server to set up a development environment, then, build web applications using Servlet and JSP pages.

Tomcat is the official reference implementation of the Java Servlet 2.2 and JavaServer Pages 1.1 technologies. Developed under the Apache license in an open and participatory environment, it is intended to be a collaboration of the best-of-breed developers from around the world. For more information about Servlet, one can visit [Tom].

Chapter 4

Experiments

In Chapter III, we have described our approach to domain-specific QA as well as its implementation. We mentioned that some settings (such as, for deploying scheme, assigning coefficient, choosing parameter, determining the evaluation formula and so on) were determined by experimental results. It means, once a basic framework for domain-specific question answering system was built, we have done a number of experiments based on it for determining the best configuration of the system. In this Chapter, first we will describe and analyze the main experiments that we have made for establishing Category search strategy. Then we will present the global evaluation of the system.

4.1 Document collection and question set

The documents collection contains 240 articles. The size of this collection is about 8M bytes.These articles are Canadian Building Digests published between 1960 and 1990 by NRC's Institute for Research in Construction and its predecessor, the Division of Building Research. The topics reflect the diversity of the industry and cover virtually every aspect of design and construction in Canada. This collection shows how the construction industry has evolved and also represents a real history of building practice thinking in Canada [IRC]. Thus, it is still useful for answering common

constructional questions.

Domain experts provide 100 test questions (see Appendix) based on the 240 articles for experimental evaluations. Each question is guaranteed to have one passage in the collection that answered the question. They also give the location of the correct answer for each question. The composition of these questions is as follows: 42% can be counted as Named Entity questions (42 over 100, e.g., "What is the address of the Educational Facilities Laboratories Inc."), 40% belongs to Category questions (40 over 100, e.g., "What product is used to remove the stains caused by? "), and the others 18% do not belong to these two kinds of type, and they are Keyword questions.

4.2 Evaluation method

In order to examine the system performance, which means the quality of the answers found by the system in this thesis, it is necessary to work out the measurable evaluation and analysis strategy.

There are many methods for evaluating the performance of QA system. One of them is mean reciprocal answer rank (MRR)(or reciprocal answer rank (RR)). The main idea about this method is that each question receives a score equal to the reciprocal of the rank at which the first right answer is returned (if none of the all answers is the right answer, the received score is zero.) and the score for a test set is the mean of each question's reciprocal rank (or the sum of each question's reciprocal rank). The calculation formulas for MRR and RR are as follows:

$$MRR = \frac{1}{N}\sum_{i=1}^{N}\frac{1}{rank_i}$$

$$RR = \sum_{i=1}^{N}\frac{1}{rank_i}$$

Where N represents the number of questions in test set; $rank_i$ represents the rank of i^{th} question's right answer, if none of the ranked passages list contains the right answer for the i^{th} question, $\frac{1}{rank_i}$ is equal to zero. By "system performance", we will

mean MRR.

The question answering track in TREC-8 adopted this evaluation method. However, they just took the first five responses into account. If none of the five responses contained a correct answer, the received score was zero [VH99]. In our system, we also adopt this method but broaden this limit to the first fifty responses.

For analysing the system performance, we further divide the test questions into three cases in terms of the test results; they are UP, DOWN and NO CHANGE. UP (improving) means the rank of the right answer moves up through the post-processing for one question. DOWN (worsening) means the rank of the right answer moves down through the post-processing . NO CHANGE means the rank of the right answer doesn't change through the post-processing for one question. The performances with post-processing are all compared with the results of the Keyword search by Okapi. Then we calculate the UP rate, the DOWN rate and the NO CHANGE rate. The calculating formulas are as following,

$$UP\ rate = \frac{the\ number\ of\ UP\ questions}{the\ number\ of\ questions}$$

$$DOWN\ rate = \frac{the\ number\ of\ DOWN\ questions}{the\ number\ of\ questions}$$

$$NO\ CHANGE\ rate = \frac{the\ number\ of\ NO\ CHANGE\ questions}{the\ number\ of\ questions}$$

Finally, we analyse the causes that changed the system performance.

In our description of experiments, we will use absolute improvements (instead of relative improvements as in the literature). For example, if the MRR is changed from 20% to 25% compared with baseline method, which is based on keyword search, we will talk about an improvement of 5%.

4.3 Experiments on Category search strategy

Category search strategy is used for solving domain-specific questions. The main idea is to identify the semantic categories of specialized concepts, so that one may ask questions on these categories. Several problems are concerned in this method:

- the determination of the categories.
- the determination of the weight.
- the retrieval strategy in combination with the keyword-based search.

4.3.1 Choosing categories

For choosing categories, first, we adopted a method of fixed categories. It means that some fixed thesaurus categories are chosen by experts as categories. We determined about 80 categories. Almost 70% of the terms in the thesaurus can be covered by these categories. We used these categories to tag documents and questions. Then, we test the performance of system. Unfortunately, this method gives a decrease of 6.1% in the system performance in comparison with keyword-based search. Through analysis, we find that the failure is due to the following reasons:

1. The coverage of the categories is not large enough. These 80 categories can't cover all the terms in the thesaurus. Some terms can't be tagged with a category. Therefore, some useful semantic information will be lost.

2. The scope of the categories is usually too large. All the terms in the thesaurus are divided into eleven levels. The higher the level is, the broader the scope of the term is. These categories are from level 3 or 4. In this case, specific terms are often over-generalized to their level 3 or 4 categories. As a result, a lot of noise is produced by the system.

3. Some terms and relationships are ambiguous, especially for the long links among terms. For example, suppose a category chain $A- > B- > C- > D$. It means

that A is a sub-category of B, B is a sub-category of C and C is a sub-category of D. If D is selected as a fixed category for tagging, then the concept A will be tagged as category D. However, as D and A are separated by several levels, their relationship may become weak. Therefore, tagging A as category D may become unreasonable.

For the three reasons listed above, we have to abandon this idea. In order to solve the problems, we design a dynamic method for choosing categories. This method contributes 7.11% improvement for the system performance. The detailed description about this method is given in Section 3.5. The main idea of this dynamic method is that we use the directly upper level's category (Broader Term relationship) and lower level's (Narrower Term relationship) terms. For example, for determining the category of the sixth level's term, we should check the fifth level's terms that have Broader Term relationship with this term and the seventh level's terms that have Narrower Term relationship with this term.

Now we will explain why these three problems happened in fixed terms' method can be solved. First, we know that each term in the thesaurus is accessible and there is no isolated node in the thesaurus. That seems we can find categories for all the term in the thesaurus. Obviously, the coverage is large enough. The first problem disappears.

The second problem is about how to determine the level of category. In this dynamic method, the level of category is subject to the level of the term. There are two cases; they are identical (category Level: n, Term Level: n), or the former (category Level: n-1) is one larger than the latter (Term Level: n). The case that the level of category (category Level: n-2, n-3, n-4 and so on) is much less than the level of Term (Term Level: n) doesn't exist any more. Thus, the lower-level terms are not converged overly.

Third, we don't use multi-level reasonning (such as, $A- > B- > C- > D$) in the dynamic method. In this way, we can limit the problems due to the thesaurus.

4.3.2 The weighting problem

For the weighting problem, our consideration focuses on finding the relation among the weights of the common NE, categories and Keyword. First, we suppose categories can provide the same semantic information as the common NE for sentence. Thus, we assigned a high weight to categories. However, our results show serious problem with this weight assignment. Its contribution to the system performance is negative. This indicates that categories are less important than the common NE. Then, we assigned the same weight for categories and keyword. However, for some professional questions, the role of categories is not stressed enough. Finally, we choose a combined method. The weight for categories is greater than the weight for keywords and less than the weight for common NE.

4.3.3 The combination of Keyword search and Category search

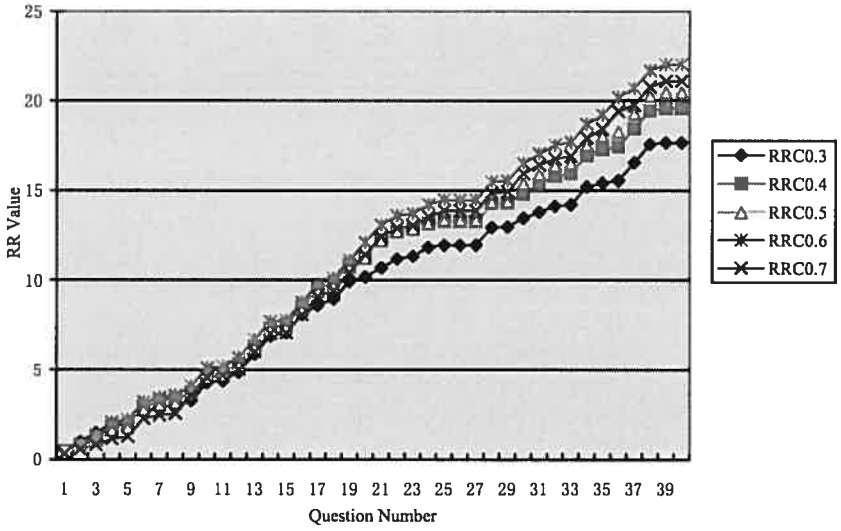
For the combination of Keyword search and Category search, we multiply a weighting coefficient for each search result respectively and limit the sum of this two weighting coefficient is equal to one. It is as follows:

$$Weight = \alpha * K_{weight} + \beta * C_{weight}, \quad \alpha + \beta = 1.0.$$

K_{weight} represents the weight of Keyword search. C_{weight} represents the weight of Category search. α and β are weighting coefficient for Keyword search and Category search.

Figure 4.1 shows the comparison of RR for different weight assignment methods. RRC0.3 (0.4, 0.5, 0.6, 0.7) represents the RR value that the weighting coefficient of Category search is equal to 0.3 (0.4, 0.5, 0.6, 0.7). Clearly, the system will obtain the best performance when the proportion of Category search and Keyword search is 3 : 2.

For the other search strategies, we conducted similar experiences to determine the coefficients used. We don't describe them in detail.



RR: Reciprocal answer Rank.
RRC: Reciprocal answer Rank for Category search.

Figure 4.1: Comparison of Category searches

4.4 Evaluation of the system

In this section, first, we will illustrate the experimental results of Category search and NE search. Then, we will show the performance of the global search strategy.

4.4.1 Category search performance

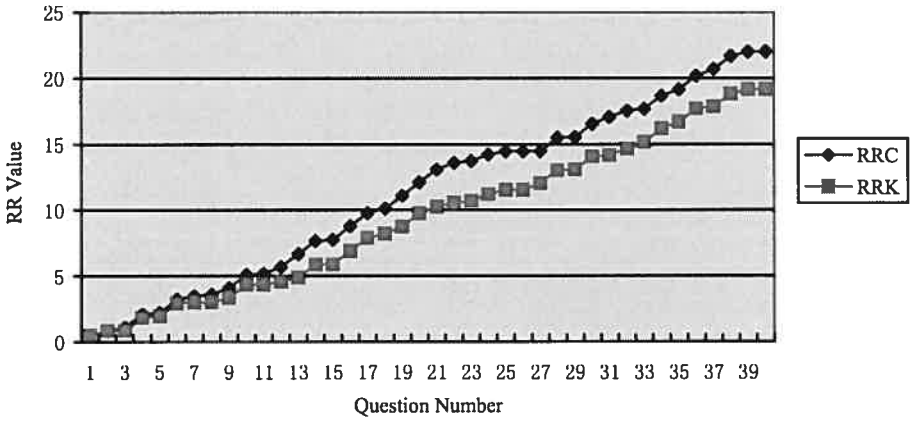
In this section, we mainly analyze the performance of Category search strategy. 40 questions out of 100 questions require Category search strategy. Figure 4.2 shows the comparison of RR between Category search strategy and Keyword search strategy. The MRR value of Keyword search is 0.4789. The MRR value of Category search is 0.55. The improvement of the performance is 7.11%.

Table 4.1 shows the detailed test results. We can see that 35% of questions are better answered with Category search. 55% of questions are unchanged. It seems the percentage of unchanged case is very high. However, through analysis, we find that the correct answer of 59.1% of NO CHANGE questions has been at the first position in Keyword search. On the other hand, our results show that only 10% of questions have decreased the system performance. Globally this result is encouraging.

Through our analysis, we found that the causes that contributed to the improvement in the system performance are as follows:

1. Before carrying out searching, we tagged categories in questions and documents. This means that we add much semantic information into them. Therefore, the searching is not only based on keyword search but also based on concept search to some extent because the concept categories are also used as indexes. The category-based search is more precise than simple keyword search.

2. In preprocessing, we extracted compound terms with the help of thesaurus. This contributed to reducing terms' ambiguities during searching.



RRC: Reciprocal answer Rank for Category search
RRK: Reciprocal answer Rank for Keyword search

Figure 4.2: RR performance comparison between Keyword and Category search

	Number	Rate
UP	14	35%
DOWN	4	10%
NO CHANGE	22	55%

Table 4.1: The test results of Category search strategy.

On the other hand, the decrease of performance for some other questions is due to the following factors:

1. The quality of the categories. There are still some problems on determining categories for domain terms. In the dynamic category method, we only use the direct upper level as the category of a term, i.e., we only use direct hierarchical links such $A- > B$. In some cases, longer links should be used (e.g., $A- > B- > C$) in order to extend the coverage of category search. In the future, it may be a good idea to associate a weight to each link, and to allow the use of longer links.

2. The quality of the thesaurus. The coverage of the thesaurus is limited. There exist some limits in thesaurus because there aren't precise classifying standards for some terms. Hence, sometimes, we cannot find a category (Null) or can find a wrong category for a term. Experimental results show a wrong category is even worse than Null category.

3. The assignment of weighting coefficient for the combination of Keyword search and Category search. In our system, we set the coefficients of Category search and Keyword search to 3:2. This setting works well for some questions but not for all. For some questions, a different setting such as 7:3 or 1:1 may be better.

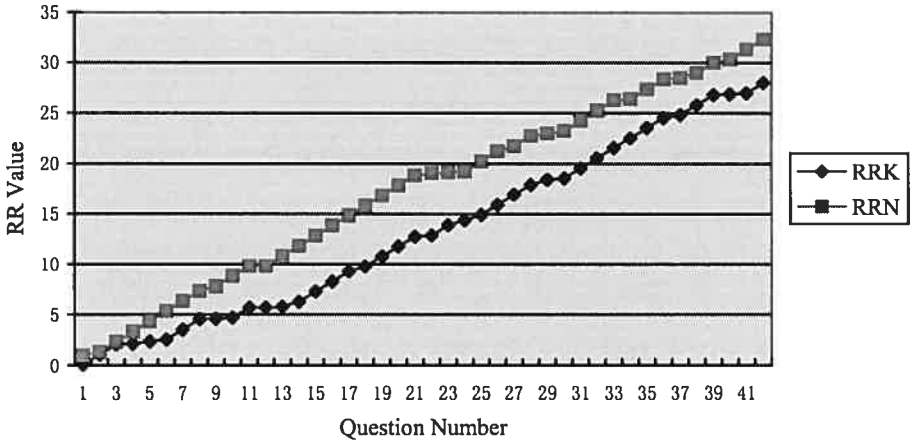
4. The correct answer isn't contained in the ranked passages list returned by Okapi search engine. If the correct answer is not in this list, there is no way for the post-processing to improve the result.

Our analysis results show that the first three factors are responsible for most of the DOWN questions and the fourth factor is responsible for the NO CHANGE questions.

4.4.2 NE search performance

In this section, we analyze the performance of NE search (including Definition search). 42 questions out of 100 questions are in this case. Figure 4.3 shows the performance comparison of NE search and Keyword search. The MRR value of Keyword search is 0.6663. The MRR value of NE search is 0.7698. The improvement of the performance is 10.35%.

Table 4.2 shows the detailed test results. 33.33% of questions have improved results. 47.62% of the questions are unchanged. Among them 95% questions have the correct answer at the first position in Keyword search. On the other hand, our results show that 19.05% of questions have decreased the system performance. This figure is higher than for Category search.



RRK: Reciprocal answer Rank for Keyword search
RRN: Reciprocal answer Rank for NE search

Figure 4.3: Comparison of RR performance between NE and Keyword search

	Number	Rate
UP	14	33.33%
DOWN	8	19.05%
NO CHANGE	20	47.62%

Table 4.2: The test results of NE search strategy.

Our analysis of the experimental results show that the following factors have affected the system performance:

1. The problem of question processing. There are several elements for this problem. First, the classification of question type is too coarse, especially for LOCATION and NUMBER type. We should divide them into finer question types. For

example, LOCATION type includes City, Country, Province and other entities. Sometimes, some questions just focus on City or Country but we still include them into the LOCATION type. This will worsen the system performance. Second, there are some errors in syntactic analysis so that the correct *identifying word*[1] cannot be determined. Consequently, we don't get the expected question type. Third, the thesaurus contain some ambiguities and has a limited coverage. This will affect our detection of question type.

2. The problem of NE recognizer. First, we adopt a heuristics-based method for tagging NE. Obviously, the techniques are rather simple and error proning. Second, the tagged NE types are not abundant. For some entities, such as Density, Pressure, and so on, we don't have enough features about them. Therefore, we didn't process them in questions and documents. In the future, we should do more on it because the majority of errors made by the name entity annotation can seriously affect the system performance.

3. The problem of weight assignment. We assign a high weight for the common NE. This is a tradeoff scheme. It is not suitable for all the NE types.

4. The problem of passage retrieval. The correct answer for some questions isn't contained in the ranked passages list returned by Okapi search engine. The post-processing cannot make any improvement for these questions.

The first two factors are responsible for 87.5% of the DOWN questions. The third factor is responsible for 12.5% of the DOWN questions. And the fourth factor is the main reason for 5% of NO CHANGE questions.

4.4.3 Global Performance

In the last two sections, we have given the detailed performance analysis about Category search and NE search. In this section, we will analyze the integrated system, which is the combination of several search strategies and other components.

[1] see section 3.2.2.

18 questions out of 100 do not contain either categories or Named Entities. They are evaluated only by Keyword search. For the others, they belong to NE, Category or Definition search. Globally, for all the 100 questions, the MRR value of Keyword search is 0.5826. The MRR value of the search with post-processing is 0.6545. The absolute improvement in the performance is 7.19%. If we ignore the 18 questions on which the Post-Processing search has no effect, the improvement of the performance is 8.77%. This result is very encouraging. It shows that our post-processing, although still simple, is quite effective.

All in all, through the above analysis, we can conclude the performance of the integrated system is acceptable and encouraging.

Chapter 5

Conclusions

In this chapter, we will draw some conclusions from our work. We will also point out the remaining problems in our system, and some possible future work.

5.1 Approach and advantages

In this thesis, a domain-specific question answering system is built based on IR and NE techniques. The goal of this project is to provide a precise answer for user's questions in the construction sector.

This work involves two main parts. The first part is of a general-purpose QA system that can be applied to many other QA contents — question and document analysis. For this, we extracted the common named entities from both documents and questions and process questions in a way similar to most of the QA systems. The second part is domain-dependent. For this, domain-specific concepts are extracted by using a domain thesaurus. Then, we take these concepts as the extended Named Entities. Here the key point of our system is to extend the open-domain QA approach (based on IR and NE techniques) to a domain-specific QA system by using domain thesaurus. The second part is the core of our study and it has not been dealt with in the literature.

In order to implement the second part of this project, we had to solve three

problems:

1. how to extract the extended NEs based on the thesaurus, and use them in question answering.

2. how to determine compound terms with the thesaurus,

2. testing what search strategies have to be used for incorporating the extended NE and domain-specific compound term that have been extracted from the thesaurus.

To answer question one, we designed a dynamic method for choosing categories. This method brought an absolute improvement of 7.11% for the questions of this type.

For question three, we designed three search strategies for Category search, NE search and Definition search. The system performance by using Category search strategy is increased by 7.11%, and by using NE and Definition search strategy, it is increased by 10.35%. Using these search strategies, the system performance is much better than using Keyword search strategy alone.

To sum up, through extending the common NE concept into domain-specific NE concept or categories, the method based on IR and NE techniques in open-domain QA can be extended to domain-specific QA. The performance of the integrated system is acceptable and encouraging.

5.2 Remaining problems

Although the performance of the integrated system in terms of effectiveness and response time is acceptable and encouraging, there is still room for improvement. In this section, we will show the existing problems on which improvements can be made in the future.

- **The problem of question processing**

  First, the classification of question type is too coarse, especially for LOCATION and NUMBER type. We should divide them into more refined question types. For example, LOCATION type includes City, Country, Province and other entities. However, a question asking a City cannot be answered by any LOCATION.

  Second, there are some errors in syntactic analysis by the statistitical tagger so that the right identifying word cannot be obtained. Consequently, we don't get the expected question type.

  Third, the thesaurus does not have a good coverage of all the specialized terms in construction. Thesaurus enhancement will be a key element for future improvement.

- **The problem of NE recognition**

  First, we adopt heuristics-based method for tagging NE. Obviously, the techniques are rather simple and error-proning.

  Second, the number of tagged NE types is not large. For some entities, such as Density, Pressure, and so on, we didn't process them in questions and documents. In the future work, we should extend the NE types recognized in our system. This is important because the majority of errors made by the name entity annotation can produce serious effect on system performance. This problem occurs mainly when NE search strategy is used.

- **The problem of the categories**

  There are still some problems on determining categories for domain-specific terms. In the dynamic category method, we only use the direct upper level as the category of a term, i.e., we only use direct hierarchical links such $A->B$. In some cases, longer links should be used (e.g., $A->B->C$) in order to extend the coverage of category search. In the future, it may be a good idea to associate a weight to each link, and to allow the use of longer links.

- **The problem of weight assignment**

  We assign a fixed weights to common NE, extended NE and keywords. This is a setting determined empirically. It is not the most suitable formula for all the types. We should define more elaborated weighting formula in the future.

- **The problem of passage retrieval**

  Sometimes, the right answer isn't contained in the ranked passages list returned by Okapi search engine. If the right answer cannot be included in this list, the post-processing can do nothing. In the future, we should also try to improve the quality of passage retrieval so that the correct passage will appear in the top-ranked results.

5.3 Future work

In order to solve the existing problems, we will discuss what we should do in the future.

First, we have to do more work on question processing and NE tagging. For question processing, we should refine our processing of user's questions so that we can identify more question types. The sets of questions of TREC provide a good reference for doing this. About NE tagging, there are some advanced methods published recently, namely, unsupervised learning method may be a good choice for us.

Second, we will pay more attention to the domain resource. It is useful to integrate an automatic knowledge acquisition component into the system to extend the thesaurus. A statistical thesaurus based on occurrence analysis may be a good complement to a man-made thesaurus. On the use of the thesaurus, as we discussed, it may be beneficial to assign a weigh to each link between two terms in the thesaurus, and to use longer link chains in our reasoning during semantic annotation and retrieval.

Finally, to improve the quality of IR system, it is possible to use multiple IR system. Many researches indicate that combining the results of different systems

acting on the same queries can provide superior performance than individual system [FD92, BCCC93]. Therefore, if we combine Okapi with some other IR systems (e.g., Smart), it is possible to obtain improved results.

Globally, this study has shown that the existing techniques for QA can be easily adapted to a specialized domain. If we change the application area, we have to deal with the following aspects: 1). defining new domain categories (extended NEs) based on the new thesaurus; 2). defining some new patterns and rules related to the most frequent NEs in the new domain; 3). tuning the coefficients and parameters by making some experiments. However, the basic approach and the mechanisms we implemented can be the same. In this work, we have shown that it is possible to extend the idea of named entity to specialized categories, so that professionals can also ask questions on these categories. Our experiments have shown that our approach is feasible and effective.

Bibliography

[AB00] G. Attardi and C. Burrini. The PISAB question answering system. In *Proceedings of Text Retrieval Conference (Trec-9), NIST*, pages 446–451, Gaithersburg (MD), November 2000.

[ABH98] D. Aliod, J. Berri, and M. Hess. A real world implementation of answer extraction. In *Proceedings of the 9th International Workshop on Database and Expert Systems, Workshop: Natural Languageand Information Systems (NLIS-98)*, 1998.

[ACS00] S. Abney, M. Collins, and A. Singhal. Answer extraction. In *the Proceedings of ANLP 2000*, Seattle, 2000.

[ARB94] A. R. Aronson, T. C. Rindflesch, and A. C. Browne. Exploiting a large thesaurus for information retrieval. In *Proccedings of RIAO 94*, pages 197–216, New York, 1994.

[BCCC93] N. J. Belkin, C. Cool, W.B. Croft, and J.P. Callan. The effect of multiple query representations on information retrieval system performance. In *Proceedings of SIGIR'93*, Pittsburgh, PA, 1993.

[BMSW97] D. M. Bikel, S. Miller, R. Schwartz, and R. Weischedel. Nymble: a high-performance learning name-finder. In *Proceedings of the Fifth conference on Applied Natural Language Processing*, 1997.

[BON03] Oliver Bender, Franz Josef Och, and Hermann Ney. Maximum entropy models for named entity recognition. In *Proccedings of CoNLL 2003 Shared Task Contribution*, Edmonton, Canada, May 31 - Jun 1 2003.

[Bur] U.S. Census Bureau. Census. http://www.census.gov/genealogy/names/.

[BYRN99] Baeza-Yates and B. Ribeiro-Neto. *Modern Information Retrieval.* ACM Press Books and Addison Wesley Longman Limited, 1999.

[CC03] James R. Curran and Stephen Clark. Language independent ner using a maximum entropy tagger. In *Proccedings of CoNLL 2003 Shared Task Contribution*, Edmonton, Canada, May 31 - Jun 1 2003.

[CCKL00] C. L. A. Clarke, G. V. Cormack, D. I. E. Kisman, and T. R. Lynam. Question answering by passage selection (MultiText Experiments for TREC-9). In *NIST Special Publication 500-249: The Ninth Text REtrieval Conference (TREC 9)*, pages 673–683, 2000.

[CMP03a] Xavier Carreras, Lluis Marquez, and Lluis Padro. Learning a perceptron-based named entity chunker via online recognition feedback. In *Proccedings of CoNLL 2003 Shared Task Contribution*, Edmonton, Canada, May 31 - Jun 1 2003.

[CMP03b] Xavier Carreras, Lluis Marquez, and Lluis Padro. A simple named entity extractor using adaboost. In *Proccedings of CoNLL 2003 Shared Task Contribution*, Edmonton, Canada, May 31 - Jun 1 2003.

[CN03] Hai Leong Chieu and Hwee Tou Ng. Named entity recognition with a maximum entropy approach. In *Proccedings of CoNLL 2003 Shared Task Contribution*, Edmonton, Canada, May 31 - Jun 1 2003.

[CoN02] CoNLL. CoNLL2002. In *Sixth Conference on Natural Language Learning*, Taipei, Taiwan, August 31 - September 1 2002.

[CoN03] CoNLL. CoNLL2003. In *Seventh Conference on Natural Language Learning*, Edmonton, Canada, May 31 - June 1, 2003.

[CS99] M. Collins and Y. Singer. Unsupervised models for named entity classification. In *Proceedings of the Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora*, 1999.

[CTHD00] P. Clark, J. Thompson, H. Holmback, and L. Duncan. Exploiting a thesaurus-based semantic net for knowledge-based search. In *Proceedings of the 12th Conference on Innovative Applications of AI*, pages 988–995, 2000.

[DeJ82] G. DeJong. An overview of the frump system. In *Strategies for Natural Language Processing*, pages 149–176, W.G.Lehnert & M.H.Ringle (Eds), Lawrence Erlbaum Associates, 1982.

[DJGC95] Colin H. Davidson, Michel Jullien, Pierre Garneau, and Jean-Jacques Chailloux. The canadian thesaurus of construction science and technology. http://irc.nrc-cnrc.gc.ca/thesaurus/, 1995.

[Edu] Buffalo Education. U.S. Gazetteer. http://wings.buffalo.edu/geogw.

[EFO+02] G. Eriksson, K. Franzen, F. Olsson, L. Asker, and P. Liden. Using heuristics, syntax and a local dynamic dictionary for protein name tagging. In *Proceedings of Human Language Technology 2002*, San Diego, USA, March 2002.

[FD92] P. W. Foltz and S. T. Dumais. Personalized information delivery: An analysis of information filtering methods. In *Communications of the ACM*, pages 35, 12:51–60, 1992.

[FIJZ03] Radu Florian, Abe Ittycheriah, Hongyan Jing, and Tong Zhang. Named entity recognition through classifier combination. In *Proccedings of CoNLL 2003 Shared Task Contribution*, Edmonton, Canada, May 31 - Jun 1 2003.

[Gat] Gate. Named entity extractor. http://www.gate.ac.uk.

[Gaz] The world gazetteer. http://www.gazetteer.de/.

[GG91] A. Graesser and S. Gordon. *Question-Answering and the Organization of World Knowledge*, chapter Question-Answering and the Organization of World Knowledge. Lawrence Erlbaum, 1991.

[GH00] R. Gaizauskas and K. Humphreys. A combined IR/NLP approach to question answering against large text collections. In *Proceedings of RIAO 2000: Content-Based Multimedia Information Access*, Paris, France, April 2000.

[Gro96] Sheffield NLP Group. Information extraction. http://www.dcs.shef.ac.uk/ research/groups/nlp/extraction/, 1996.

[Gro01] Sheffield NLP Group. Named entity recognition from diverse text types. http://gate.ac.uk/talks/ranlp01/03.html, September 2001.

[Gro02] Sheffield NLP Group. Information extraction. http://nlp.shef.ac.uk /research/areas/ie.html, 2002.

[GW98] R. Gaizauskas and Y. Wilks. Information extraction: Beyond document retrieval. *Computational Linguistics and Chinese Language Processing*, 3:no. 2, 17–60, August 1998.

[Ham03] James Hammerton. Named entity recognition with long short-term memory. In *Proccedings of CoNLL 2003 Shared Task Contribution*, Edmonton, Canada, May 31 - Jun 1 2003.

[Hea02] Marti Hearst. Search and retrieval: Term weighting and document ranking. http://www.sims.berkeley.edu/academics/courses/is202/f97 /lecture21/sld001.htm, 2002.

[HGH+00] E. Hovy, L. Gerber, U. Hermjakob, M. Junk, and C-Y Lin. Question answering in webclopedia. In *NIST Special Publication 500-249: The Ninth Text REtrieval Conference (TREC 9)*, pages 655–664, 2000.

[HMM+00] S. Harabagiu, D. Moldovan, R. Mihalcea, M. Pasca, R. Bunescu, M. Surdeanu, R. G. Irju, V. Rus, and P. Morarescu. Falcon: Boosting knowledge for answer engines. In *NIST Special Publication 500-249: The Ninth Text REtrieval Conference (TREC 9)*, pages 479–488, 2000.

[HvdB03] Iris Hendrickx and Antal van den Bosch. Memory-based one-step named-entity recognition: Effects of seed list features, classifier stacking, and unannotated data. In *Proccedings of CoNLL 2003 Shared Task Contribution*, Edmonton, Canada, May 31 - Jun 1 2003.

[IE001] Information Extraction, Query-Relevant Summarization and Question Answering: an Overview. http://www-users.cs.york.ac.uk / mdeboni/research/general/overview.html, 2001.

[Inf] MIT Infolab. Natural language question answering system. http://www .ai.mit.edu/projects/infolab.

[IRC] IRC. Institute for research in construction. http://irc.nrc-cnrc.gc.ca /cbd/cbd-e.htm.

[JC94] Y. Jing and B. Croft. An association thesaurus for information retrieval. In *Proceedings of RIAO*, pages 146–160, 1994.

[KSNM03] Dan Klein, Joseph Smarr, Huy Nguyen, and Christopher D. Manning. Named entity recognition with character-level models. In *Proccedings of CoNLL 2003 Shared Task Contribution*, Edmonton, Canada, May 31 - Jun 1 2003.

[Leh75] W. Lehnert. Script based techniques for question answering. In *Proceedings of Theoretical Issues in Natural Language Processing*, Cambridge, Massachusetts, 1975.

[Leh78] W. Lehnert. The process of question answering: a computer simulation of cognition. Lawrence Erlbaum, 1978.

[Lib02] Alexandria Digital Library. Gazetteer development. http:// alexandria.sdc.ucsb.edu/gazetteer/, 2002.

[Lin01] Jimmy J. Lin. Indexing and retrieving natural language using ternary expressions. Technical report, Massachusetts Institute of Technology, 2001.

[MD03] Fien De Meulder and Walter Daelemans. Memory-based named entity recognition using unannotated data. In *Proccedings of CoNLL 2003 Shared Task Contribution*, Edmonton, Canada, May 31 - Jun 1 2003.

[ML03] Andrew McCallum and Wei Li. Early results for named entity recognition with conditional random fields, feature induction and web-enhanced lexicons. In *Proccedings of CoNLL 2003 Shared Task Contribution*, Edmonton, Canada, May 31 - Jun 1 2003.

[MLP03] Robert Munro, Daren Ler, and Jon Patrick. Meta-learning orthographic and contex-tual models for language independent named entity recognition. In *Proccedings of CoNLL 2003 Shared Task Contribution*, Edmonton, Canada, May 31 - Jun 1 2003.

[MMP03] James Mayfield, Paul McNamee, and Christine Piatko. Named entity recognition using hundreds of thousands of features. In *Proccedings of CoNLL 2003 Shared Task Contribution*, Edmonton, Canada, May 31 - Jun 1 2003.

[MUC95] MUC-6. Named entity task definition. http://65.54.246.250:80/cgi-bin/, 1995.

[MUC03] MUC-7. Message understanding conference proceedings. http://www.itl.nist.gov/iaui/894.02/related_projects/muc/proceedings/muc_7_toc.html, 2003.

[Mys] MySQL. http://www.mysql.com/.

[Nie03] J. Y. Nie. http://www.iro.umontreal.ca/ nie/IFT6255/Introduction.html, 2003.

[NIS03] NIST. Tipster text program. http://www.itl.nist.gov/iaui/894.02/related_projects/tipster/overv.htm, 2003.

[Oka] Okapi. http://www.public.iastate.edu/ CYBERSTACKS/Onion5.htm.

[QBW02] U. Quasthoff, C. Biemann, and C. Wolff. Named entity learning and verification: Expectation maximization in large corpora. In *Proccedings of CoNLL 2002 Shared Task Contribution*, Taipei, Taiwan, September 2002.

[RFQ+02] D. Radev, W. G. Fan, H. Qi, H. Wu, and A. Grewal. Probabilistic question answering on the web. *Series-Proceeding-Section-Article, ACM Press*, pages 408–419, 2002.

[RPS00] D. Radev, J. M. Prager, and V. Saran. Ranking suspected answers to natural language questions using predictive annotation. In *Proceedings of ANLP'OO*, Seattle, WA, 2000.

[Sag81] N. Sager. Natural language information processing. Reading, Massachusetts: Addison Wesley, 1981.

[San02] Erik F. Tjong Kim Sang. Introduction to the conll-2002 shared task: Language-independent named entity recognition. In *Proceedings of the Sixth Conference on Natural Language Learning (CONLL-2002)*, Taipei, Taiwan, August 2002.

[SL00] R. Srihari and W. Li. Question answering system supported by information extraction. In *Proceedings of the 1st Meeting of the North American Chapter of the Association for Computational Linguistics (NAACL-2000)*, May 2000.

[SM83] G. Salton and M. J. McGill. *Introduction to Modern Information Retrieval.* McGraw-Hill, 1983.

[SM03] Erik F. Tjong Kim Sang and Fien De Meulder. Introduction to the conll-2003 shared task: Language-independent named entity recognition. In *Proccedings of CoNLL 2003 Shared Task Contribution*, Edmonton, Canada, May 31 - Jun 1 2003.

[Tom] Tomcat. http://jakarta.apache.org/tomcat/.

[Uni] Michigan University. Question answering:computational linguistic and information retrieval. http://perun.si.umich.edu /clair/home/qa.htm.

[VH99] E. M. Voorhees and D. Harman. Overview of the Eighth Text REtrieval Conference (TREC-8). In *Proceedings of the Eighth Text REtrieval Conference*, pages 1–33 and A.17–A.18, 1999.

[WNC03] Dekai Wu, Grace Ngai, and Marine Carpuat. A stacked, voted, stacked model for named entity recognition. In *Proccedings of CoNLL 2003 Shared Task Contribution*, Edmonton, Canada, May 31 - Jun 1 2003.

[WP03] Casey Whitelaw and Jon Patrick. Named entity recognition using a character-based probabilistic approach. In *Proccedings of CoNLL 2003 Shared Task Contribution*, Edmonton, Canada, May 31 - Jun 1 2003.

[Zha03] Qi Zhang. A domain-specific search engine for the construction sector. Technical report, Msc. thesis, DIRO, University of Montreal, 2003.

[ZJ03] Tong Zhang and David Johnson. A robustrisk minimization based named entity recognition system. In *Proccedings of CoNLL 2003 Shared Task Contribution*, Edmonton, Canada, May 31 - Jun 1 2003.

Appendix

Questions:

1. What is the coefficient of expansion for aluminum?
2. What are the common thermoset foams used in construction ?
3. What organizations have published information about corrosion tests?
4. What organization in Canada is in charge of registering earthquakes and seismic activity?
5. What is the address of the Educational Facilities Laboratories Inc.?
6. What are the dimensions of a Norman brick?
7. What organization publishes the Tables of Computed altitude and azimuth?
8. What sections of the National Building code deal with the requirements for smoke-generation in construction materials ?
9. What is the price of the climatological atlas of Canada?
10. What is the address of the Meteorological Branch of the Department of Transport?
11. What organization has a glossary of paint terms?
12. What is the relative humidity in Vancouver?
13. What organization in Canada distributes the book Concrete Floor Finishes?

14. What is the period of time in which caulking compounds become rigid?

15. According to the code, what is the maximum horizontal distance admitted in between ties in a regular cavity wall?

16. According to the National Building Code, what is the space required in between the inner and outer walls in a cavity wall?

17. What organization publishes the thermal resistances of building materials?

18. What is the potential tensile strength of glass?

19. What is the coefficient of expansion of glass?

20. What organization develops observations of ground temperature measurements in Canada?

21. According to the Canadian Standards, what is the maximum density of people per sq mt in an elevator?

22. What is the maximum inaccuracy between the main floors level and the elevators floor level keeping in mind handicap regulations?

23. What American institution regulates the standards and codes for constructions in concrete?

24. What publications include Canadian design specifications for disabled people?

25. What is the address of the Canadian Rehabilitation Council for the Disabled?

26. What is the movement capability of silicon sealants?

27. What A.C.I Committee publications deal with the properties and maintenance of sealants ?

28. What is the recommended dose of muriatic acid and water for after-construction cleaning of bricks?

29. What product is used to remove the stains caused by copper elements on bricks ?

30. What is the required amount of outdoor air required for the ventilation of a gym room?

31. What is the address of the Specifications Writers Association of Canada?

32. What is the contact address of the American Tile Council?

33. What is the location of the Canadian meteorological stations that measure skylight?

34. What is the temperature of the water required to prepare warm mortars for masonry construction?

35. What is the acceptable deflection of steel structural elements in normal conditions ?

36. What temperature of the water optimizes the service life of hot water tanks?

37. What is the recommended size of the gravel used for terrace roofs?

38. What publication of the National Fire Protection Association includes information about fire loads ?

39. What are the advantages of using superplasticizers?

40. What are the seismically active regions of Canada?

41. What are the methods for determining pressure ratings?

42. Which of the Canadian technical specifications are applicable to caulking compounds?

43. What kind of glass is used for kitchenware?

44. What is the best penetration non-destructive test of concrete?

45. What is the classification of Portland cement used in the United States ?

46. What are the main causes of deformation of building elements?

47. What is the recommended vibration frequency for the floors of dancing-club facilities?

48. What method is used to clean fireplace stains from smoke ?

49. What are the most common vapor barriers used in home construction ?

50. What are the causes of deflection of horizontal elements in floors ?

51. According to the National Building Code, what is the snow load that has to be considered for roofs in Canada?

52. What is corrosion?

53. Where can I find the thermal resistance of building materials?

54. Design of exit signs

55. Temperature gradient / building envelope

56. Soil / permeability

57. Issues about the location of drains

58. Aspects related with the chemical resistance of pipes.

59. Is it possible to use glass-fibre reinforced cement in structural elements?

60. Which norms of the building code have to be considered in the renovation of an existing building?

61. Where can I find information about the influence of radon in human health?

62. I am looking for information regarding the use of computers in the industry

63. How to prevent wood from decaying under the influence of water?

64. Drainage / erosion / filters

65. Design considerations for roofs in cold regions

66. Research about shadow angles and solar shading in faades

67. Doors insulation

68. Which trees should I use to reduce water demand in the soil?

69. Reducing rain penetration in prefabricated walls

70. In soil testing, what does swelling mean?

71. Glazing design / rain penetration / construction details

72. What are the silts?

73. Considering sound transmission, what are the specifications recommended for a party wall in between two apartments?

74. The selection of the type of foundation

75. What is the stack effect in buildings?

76. How to establish the air supply rate in buildings?

77. Established dimensions for the access of wheelchairs

78. How to build a winter shelter for construction sites in Canada?

79. What is polymer concrete?

80. Does the National Building Code accept the construction of wood frame foundations?

81. How to reduce the corrosion of the reinforcing steel in garages?

82. Where can I find a map of Canada with the seismic risk regions?

83. The address of the Standards Council of Canada

84. The Building Research Library

85. What causes air pressure differences in windows?

86. Waterproofing the Basement

87. How to find information about solar radiation on walls for the particular case of Canada?

88. What is the loss of noise transmission recommended for adjacent rooms in apartments?

89. What is efflorescence?

90. The effect of color in the temperature of roofs

91. What is the recommended temperature for the water of an indoor pool?

92. Volume changes in concrete structures due to moisture changes

93. The Canadian Building Digests

94. Do the clear urethanes perform well to the influence of UV radiation?

95. What is the recommended mortar for laying reclaimed bricks?

96. Rock formations and pyrite

97. Central control and monitoring systems

98. Does it exist a relation between condensation and roof forms?

99. What is the maximum tolerable noise level accepted in apartments?

100. Degree of comfort of ground-level winds