An Intelligent FAQ Answering System Using a Combination of Statistic and Semantic IR Techniques

Liu Jianan
Department of Computer Science and Electronics at Mälardalen University
jlu05001@student.mdh.se

Supervisors: Peter Funk
Dept. of Computer Science and Electronics at Mälardalen University
Peter Sävström
Allab IT Consultant Company, Västerås, Sweden

Examiner: Peter Funk

Masters Thesis
Nov. 2006
Abstract

In this thesis, we introduced an intelligent question-answering system called intelligent FAQ answering system. Generally speaking, it’s a natural language question answering system that uses files of frequently-asked questions as knowledge base, in another word it retrieves existing answers found in FAQ files. To achieve an exact matching result, a combination of standard information retrieval techniques from statistic point of view and semantic analysis which employing natural language processing techniques are embodied in the searching procedure. Besides, a rule-based and human expert module services as an assistant to give solution in case that no eligible answer were retrieved by the previous processing to ensure a satisfying performance.

This system is motivated by the public needs and will be embedded in a company website as information service to answer user’s FAQs about their products. In this thesis, we did deep researches on question-answering systems and designed the intelligent FAQ answering system which is fitted to this project at hand, while we believe it would be suggestive to the public concerns in this field as well. Detailed principle of this system and the algorithm proposed to implement the function are depicted in this thesis.

The thesis includes five chapters which covers three blocks of research problems: the first and second chapter gives an introduction as well as background knowledge on question-answering system. Research motivation, object and system summarize are presented in chapter one. In the chapter two we give a survey of question-answering systems and techniques involved, in addition, related work in this thesis are summed up. The third chapter which is the second block of the thesis introduces the principle of algorithm of the system proposed by us. Since our system is constructed by five functional modules, the algorithm of the whole system is interpreted in terms of describing the implementations executed in each module respectively. The third block contains chapter four and five, we firstly make an evaluation to the system effect on the basis of theory analysis and results from previous project and other researchers’ experiments, then advantages and disadvantages of the algorithms involved in this system and the system designing are deduced accordingly, in the end, we indicate some suggestions for future work.

Index terms¹:

FAQs, QA system, IR, KR, natural language understanding, NLP, WordNet, hyponymy, antonymy, meronymy, Synonymy, function word, suffix, Euclidian distance, ATN, part of speech, propagation, Accuracy, Recall.

¹ All these index terms and abbreviations are explained in Appendix section.
Table of Contents

ABSTRACT.......................................................................................................................................................... 1

CHAPTER 1 INTRODUCTION .......................................................................................................................... 2
  1.1 MOTIVATION................................................................................................................................................... 2
      1.1.1 Usable Information versus Useful Information .................................................................................. 2
      1.1.2 Question-Answering Systems ........................................................................................................... 2
      1.1.3 Natural Language Processing .............................................................................................................. 3
  1.2 PROJECT INTRODUCTION AND RESEARCH OBJECT .............................................................................. 4
      1.2.1 Project Introduction .......................................................................................................................... 4
      1.2.2 Research Object .................................................................................................................................. 4
  1.3 SYSTEM DESCRIPTION ............................................................................................................................... 6
      1.3.1 Fundamental ......................................................................................................................................... 6
      1.3.2 Primitive Flow Chart ......................................................................................................................... 7
  1.4 AILAB ........................................................................................................................................................... 8

CHAPTER 2 BACKGROUND KNOWLEDGE AND RELATED WORK .................................................................. 9
  2.1 DEVELOPMENTS AND BRANCHES OF QA SYSTEM ............................................................................... 9
      2.1.1 Development History ........................................................................................................................ 9
      2.1.2 Important Branches ........................................................................................................................ 9
  2.2 MAIN ENDEAVORS IN QA SYSTEMS ....................................................................................................... 11
  2.3 RELATED WORK ........................................................................................................................................ 13
      2.3.1 Information Retrieval ....................................................................................................................... 13
      2.3.2 Knowledge Representation ............................................................................................................. 14
      2.3.3 Natural Language Understanding .................................................................................................. 19
  2.4 A BACKGROUND OF INTRODUCING SEARCHING ASSISTANT ................................................................ 23

CHAPTER 3 PRINCIPLE AND ALGORITHM .................................................................................................. 25
  3.1 TOKENIZER ................................................................................................................................................. 25
  3.2 FILE RETRIEVAL ...................................................................................................................................... 26
      3.2.1 Weighting Terms ................................................................................................................................ 26
      3.2.2 K-NN ............................................................................................................................................... 28
  3.3 STATISTIC MATCHING .............................................................................................................................. 30
      3.3.1 TF-IDF ............................................................................................................................................. 30
      3.3.2 Statistic Similarity Calculation ......................................................................................................... 30
  3.4 SEMANTIC MATCHING .............................................................................................................................. 32
      3.4.1 NLP ............................................................................................................................................... 33
      3.4.2 WordNet ......................................................................................................................................... 36
      3.4.3 Semantic Similarity Calculation ....................................................................................................... 42
  3.5 SEARCHING ASSISTANT ........................................................................................................................... 44
      3.5.1 Rule-based Query Reformulation Assistant ...................................................................................... 44
      3.5.2 Human-based Assistant .................................................................................................................... 45

CHAPTER 4 EVALUATION OF THE ALGORITHMS ....................................................................................... 48

CHAPTER 5 SUMMARY AND FUTURE WORK ............................................................................................. 52

ACKNOWLEDGEMENT .................................................................................................................................. 55

APPENDIX .......................................................................................................................................................... 56

REFERENCES...................................................................................................................................................... 58
Chapter 1 Introduction

1.1 Motivation

1.1.1 Usable Information versus Useful Information

The dramatic development of internet enables people achieve a large number of information which is comparable with a super cyclopedia, nevertheless the concern of public is not only the possibility of meeting broad information, but also the capability of retrieving their desirable information, however when the range of available information increased the act of abstracting specific information may become tough. In another word, it implicitly forms a contradiction between usable information and useful information. Therefore, as users struggle to rein the enormous online wealth, a series of searching engines which act as assistant to help people explore wealth have been renewing accordingly.

A survey showed 85% of internet users using search engines and search services to find specific, at the same time, it also indicated that users are not satisfied with the performance of the current generation of search services due to majority of searching engines developed well are only able to return ranked lists of related documents, but they don’t deliver exact answers, so users have to extract the demanded answer from the large volumes of information. Ideally, the searching engines could be designed as question-answering manners, which are able to incept people’s query and output a valid answer by consulting the information library automatically. In a further step, the engine is expected to allow common users to use a kind of easy communicating language to express their demands without too much skills or experiences. It requires an advanced searching system with more sophisticated technical issues rather than simple key words matching searching engine which at a lot of time can’t catch user’s original desire or fail to pick the target from the ocean of information. An information resource with an ideal searching system just like the cyclopedia has mind, it not only has usable information available but also creates a possibility to connect users’ concern to the implicit solution in the information corpus, in another word it could recognize which is useful for you. Therefore the problem of transferring vast various internet resources to user’s desire information stimulates international interests and activities.

1.1.2 Question-Answering Systems

Question-Answering system (QA system) addressed this problem. This subject has been focused for a few decades in different forms, and currently still is a challenging topic that is actively researched. Recent successes have been reported in a series of Question-Answering evaluations that started from 1999 as part of the Text Retrieval Conference (TREC) and also have been encourage by the Message Understanding Conference (MUCs). At present, the most outstanding systems are able to answer more than two thirds of factual questions in that evaluation. [1]

As to the information resource for QA system, the uses of WWW (World Wide Web) as knowledge base were suppose to be able to answer a broad range of questions, but practically, the size and unstructured nature of the WWW makes it a very difficult task. To conquer this problem, in one hand, question-answering searching engines to open
domains, for example, Google, have been making effort in improving Information Retrieval (IR) algorithm to make their searching more exact, completed and intelligent; on the other hand, ambitious organizations, who are another group of searching engines providers, prefer to provide their customers a more advanced IR system which could return an explicit answer to the query amid a range of specific domain which is related to their business. Actually, it’s reasonable since users who visit certain community’s website are supposed to concern problems in the range of that field, and the information they concerned normally are generated by the community as an inner resource which is not fit to open widely, therefore an offset of QA systems are researched and embedded in websites of communities. To the latter one, an on-line resource or database which stored certain professional knowledge to their members or customers is linked as knowledge base, and they don’t have to tie up network resource which may complicate the matching. This kind of knowledge resource normally enables the searching and answering more efficient and exact.

There are several examples of application belongs to the type of QA systems with database. Companies for profit may want to provide an ardent service which could stimulate profit potentially in long term. A QA system to answer questions about products extends further the intent of FAQ files. Some organizations like insurance agent want to gain trusts by a friendly communicating platform, QA system. Besides, staffs may prefer to make some of their work or the general interests of their customers into the open, so that interested users may acquire related information freely. For those frequently concerned information or questions, operators can discuss about the answers to get a consensus among the group and make it available as a direct reply, thereby they don’t need to answer same queries again and again. One of the most outstanding examples of this phenomenon can be found in the vast assortment of frequently asked question files, many associated with USENET newsgroups. [2]

To a lot of QA systems with database, taking frequently-asked questions (FAQs) and answers as the content of database is a smart choice. FAQs are collected and been distributed with corresponding answers, then they are structured into a database and available on a website. Users are allowed to type in inquiries, further more the system will match the input with stored questions to find the given semantically similar question and the attached answer as well. This is the primitive principle of a FAQ answering system, which has been wildly used in different manner and purpose and becoming an important component of QA system. In this project, the QA system we introduced is characterized as a FAQ answering system with information database involved.

1.1.3 Natural Language Processing

No matter what kind of knowledge and purpose a QA system database contains, the common issue a designer has to face is how users communicate with it efficiently. Since users normally tend to be human agents, the most optimize choice of getting access to databases is querying with natural language, which indicates the direction of system characteristic. So to answer a question, a system must capable of analyze the natural language question, perhaps in the context of some ongoing interaction; it has to find one or more answers by consulting existing database. A natural language front-end database usually implements transition from natural language to SQL. While a series of problem
produced in the implementation because of the special features of human language and it widely recur to Natural Language Processing (NLP) techniques.

Considering the broad practicability of FAQ answering systems and the restrictions of NLP skill, an intelligent FAQ searching system which could recognize natural language and perform semantic matching is urgently called, this is the kernel in our project. We believe such a system will be consistent with public interests, not only to limited domain QA system users, but also to information researchers to open domain due to the similarity in searching techniques and principle.

1.2 Project Introduction and Research Object

1.2.1 Project Introduction

In this project, we are required to design an IR system for a company which could accept user’s FAQs about their products in form of natural language and return the answer according to a previously generated list of frequently asked QA pairs. It can be interpreted as to propose an intelligent FAQ answering system for a company which is used to introduce their productions and answer customers’ questions about the products in use.

The problem can be simply addressed as: Design a natural language front-ends QA system to database with files of FAQs! Primarily speaking, we aim to design an algorithm which is able to analyze and recognize users’ inputs in natural language and then compared them with prepared questions in database, finally retrieve the semantically similar one and display it together with its attached answer to users. In such a course, it involves IR technique, NLP techniques, Knowledge Representation (KR), matching algorithm and Rule-based reasoning technique. Standard information representation and statistical similarity algorithms are used in information retrieval; in addition, shallow semantic analysis is also carried through the use of WordNet, a semantic network of English words. Our research object is taking advantage of statistical and linguistic skills combination to insure an ideal result.

1.2.2 Research Object

- Searching by Statistic Similarity

On one hand, our research object is to compare similarities between users’ questions and question sets in database from the statistic point of view. In this level, we introduced two modules, File Retrieval and Statistic Matching, to accomplish it. In both of the two modules, we employed IR techniques which by means of statistic similarities to in turns selected the most relevant FAQ file and relevant QA pairs. Several existing standard IR techniques were taken into consideration and K-Nearest Neighbor (K-NN) technique, which is a well known and effective skill based on weighted Vector Space Model, and the modified K-NN are finally competent to fit the object best. Questions are represented to vectors and each unit in those vectors are then been valued by TF-IDF formula. Meaningful key words and corresponding scores are calculated, so a statistical similarity comparison can be made through certain algorithms. In this system, I compare the statistic similarities by computing Euclidean distances and Cosine values of question vectors.
Searching by Semantic Similarity

On the other hand, we hope to carry out a further matching in semantic level as similarity of questions lies in meanings. The algorithm in the first step is totally based on the assumption that a similarity in words statistical attribution indicates pertinence in subject. Such an assumption is sensible because normally subject decides the range of dictions, but it doesn’t take linguistic complexity into account, so we can consider it as a comparison of words rather than concepts. Linguistic complexity refers to cases like Synonyms and Homographs, etc. Statistical similarity could only look the grapheme of words but faint in watch words meaning, so it takes “by sea” and “by ship” as irrespective phrases, which will results misjudge. Therefore, the following research object is to handle variations of lexicon in natural language so that a correct understanding to users’ inquiries and exact matching of answers could be achieved.

To calculate semantic similarity, KR which enables a further semantic processing being implied is required, so we have to decide what kind of representation method is suitable here. By balancing representation depth, coverage breadth as well as the project’s preference, we believe a shallow lexical semantics would be an ideal level for the system, and WordNet which is an on-line lexical reference system, inspired by current psycholinguistic theories of human lexical memory, is involved, because researches denote that such a frame could well recognize familiar basic linguistic relations including Synonyms, Hyponymy, Antonymy and Meronymy but doesn’t dig deeper, which steers clear of a far too much representation. The most ambitious feature of WordNet, however, is its attempt to organize lexical information in terms of word meanings, but not word form. In another word, it provides a system to connect word meanings through various relations. Knowledge representation includes representation of input queries and stored question-answer pairs. To increase the efficiency when running this system in real time, we can represent information in database in advance. After the representation, it is possible to employ some algorithm to weight the semantic distances between word meanings by means of the relations indicated in WordNet.

We find from the characteristic of WordNet that the shortage is words’ part of speeches affect the usage of WordNet and the calculation of semantic similarity, so one important research object in this stage is to find an approach which could confirm words’ part of speech in some certain contexts. We introduced two NLP techniques, including parsing with augment transition network (ATN)grammars, using an existing Part-of-Speech Tagger, besides, unrestricted Marker-Passing, to achieve it.

Searching Assistant

A few existing searching assistants including Shoval, IOTA, RUBRIC, etc are researched, and we are inspired that an rule-based expert system of rules could be employed to formulate the inquiry, analyzes retrieved FAQ files, reformulate inquiries if necessary and rank the search results. Here we have to pay attention on the tradeoff of query formulation degree and user’s intent. Another assumption is we can cope with it with human-based support when implicit logic happened. It means human being can be an assistant to solve user’s problem in using the searching system. Actually in some occasions, experts’ help may greatly encourage users’ trust, for example, bank or insurance, the trust from customers will directly influence corporation’s profit, so in these areas human support is necessary although it is resource cost.
In addition, to improve database, those originally uncovered questions which were answered by experts can be added to the database afterwards. It equals to a real time update to the database, so the performance of such a system will be better and better since the information resource been maintained.

1.3 System Description

1.3.1 Fundamental

Our system provides an on-line information service, which could be embedded in various websites to carry out the function of answering user’s queries by performing a searching in the available on-line database. The system includes five functional modules; they are Tokenizer, File Retrieval, Statistical Matching, Semantic Matching and Assistant. Besides, three knowledge resources which are a structured database with collected QA pairs, an on-line lexical reference system WordNet and a rule database to reformulate users’ questions, are involved. Each component of the system implements an efficacy in searching the closest QA pair and finally the selected question with a concise answer be will the output.

Here, we will firstly illustrate the flow of the whole procedure, and in next segment, the corresponding flow chart will be added. Our friendly interface suggests that users feel free to pose complete questions in natural language, and then the query is transferred to the first module, Tokenizer, where the sentence is represented to normalized term vector. After that, the tokenized user question together with QA pairs in the database is sent to the second module so that a comparison between user’s input and FAQs in different files (subjects) is performed. In this module, we employed a classical IR technique, K-NN, in consequence, K set of files ranked by relevant degree are achieved, and the user can select the top one as the further searching bound. The third module then match the term vector of user’s question and the questions contained in its confirmed relevant QA file through a statistical approach, another IR technique similar with K-NN is used here. The result is a few QA pairs which are considered to be matched with the desire in a statistical level are retrieved. In the fourth module, which is the kernel of our system, user’s term vector and those opted QA pairs in the former step are performed Natural Language Processing, Knowledge Representation, Marker-Passing algorithm and semantic similarity computation, if the outcome is satisfying (higher than a threshold), the searched question will be taken as the best matching one, and the answer attached will be presented to the user. Otherwise, if the semantic matching indicates the degree of similarities to those candidates QA pairs are all not high enough, the user’s question will be sent to the fifth module, Assistant, which relies on rule-based expert system and human intelligence to reformulate the user question or answer that question directly. An ideal reply will be given finally, and maintenance of the database is done at the same time.

We can find from the flow that the knowledge base, QA pair database and WordNet, played important roles. In the database, frequently asked question and answer pairs are collected previously and categorized to files depend on their subjects, which enable the rapid shrink of searching bound. This act embodied human intelligent greatly improved searching efficiency and exactness. In the hand, the on-line language resource, WordNet, provides the underlying semantic framework for the Semantic Matching module. It’s the foundation of the ability of recognizing variations among human words.
During the matching procedure, there are some work on the raw FAQ entries in database could be carried out off-line to improve running efficiency:

- The first one is the QA pairs can be tokenized to term vectors in advance, the method is same with what we will do to users’ questions in the first module, in addition, importance of each term in the vectors need to be weighted by TF-IDF technique, it’s a preparation for the processing in module two and three.

- The second is we have to tag the terms’ part of speech to terms of FAQs so that we can locate them in WordNet, it is required by the structure of WordNet.

- The third one is the terms should be indexed which is required by Marker-Passing algorithm in module 4.

- The forth is a primitive rule database have to be built to reformulate user’s questions if there isn’t a clear matching happened.

### 1.3.2 Primitive Flow Chart

Figure 1 illustrates the primitive flow of the processing in our system.
1.4 Allab

Our project provider, Allab, is an IT consultant company with focus on development of intelligent software that could understand human orders and then act it. The object of their products is to by means of intelligent techniques to help people to use technologies fully in nowadays. For a long time, Allab cooperate with Malardalen University in form of information exchange with different research projects, and our system in this thesis provides a scheme to one of their software products, which will be used in a company website to answer customers’ questions about their electronic products. [3] Actually, the application of our system is not limited in such one company, most commercial information networks such as America On-Line are supposed to need this kind of system.
Chapter 2 Background Knowledge and Related Work

2.1 Developments and Branches of QA System

2.1.1 Development History

QA systems have taken different forms in the past half century, a brief review of the general development of this kind of system is presented first.

- Sixties:
  Natural language query interfaces to relational databases.

- Seventies:
  Began to include the text understanding programs such as SAM (Schank and Colby, 1973), Ms.Malaprop (Charniak, 1977), PAM (Wilensky, 1978) and POLITICS (Carbonell, 1979). These programs all parsed natural language text and created a knowledge base augmented by a knowledge-engineered structure like a situational script, a frame, or a plan. Because of the amount of knowledge engineering involved, the domains of these systems were all limited to certain topics. [4]

- Recently:
  Examining three recent systems, FAQ Finder, START, and MULDER, it reveals different approaches and challenges to using online resources as a knowledge base for a QA system. The FAQ Finder System (Burke, et al., 1997), developed by University of Chicago, uses FAQ files and the online lexicon as its knowledge base. START which is developed by Boris Katz et al uses sentence level NPL to match inquiries with sentence representations stored in its knowledge base. The digested sentences can be derived from and refer to resources on the WWW (Katz, 1997). MULDER (Cody Kwok et al., 2001), proposed by University of Washington, attempts to find answers through searching engines that index the WWW.

2.1.2 Important Branches

Since ask and answer is one of the most common ways for people to cognize new information, it happens at large. Wide application made it had been concerned for long, especially after the introduction of the Question Answering track in the Text Retrieval Conferences in 1999. In early days, natural language processing researcher Simmons once published a paper named “Answering English Questions by Computer”, where there were introductions on achievements of studies to several implemented English language question answering system around that period. During the past half century, question answering system have experienced quite active and positive changes, we will describe some representative branches in QA system which provided an important background to FAQ answering systems.

- Natural language query interfaces to databases.

The first well known natural language front-ends to database program should be BASEBALL in 1961, which is designed to answer questions about American league baseball game in the past season. Users are allowed to input queries in natural language,
then the interface will analyze the syntaxes and meanings by using linguistic knowledge, finally the initial sentence may restructured or modified to some standard format so that a further comparison to stored questions could be carried. [5] The limitation of this system is it is only feasible in baseball domain; in addition, the database has to be structured, which means it will not work to database like text collections. However this system is sophisticated even by current standard.

In nowadays, the primary challenge in this work is still about the system knowledge. Structured knowledge base in limited domain had been researched widely for a long time, while the main concern now is the problems in open domain with unstructured texts. Besides, the representing way of searching result between a direct answer with information requested only and a response containing not only the answer but also relative tips which could help understanding as well as decreasing misconception is also debated. [6]

- Question answering in human-machine dialogue

Question answering in human-machine dialogue was a theoretical interest in early times. Originally well known dialogue systems like SHRDLU (Winograd, 1972) and GUS (Bobrow, 1977) were all built as research systems to help researchers understand the issues involved in interactive advisory system and modeling human dialogue, rather than real commercial application. Both these experiments indicated some challenging problems in implementation, especially in dealing with anaphora and ellipsis. [7] Currently international attention is on spoken language interface, one successful example is MIT’s Jupiter system which is by means of telephone interface and capable of answer questions on weather report, searching information from online web, then response automatically.

- Question answering by story comprehension

Asking questions to human beings when they finish reading a story is a common testing method of judging comprehension ability, similarly, asking questions to a natural language understanding system could be a way to evaluate its quality. One famous instance is that Schank once implemented her theory on question answering in a system called QUALM (Lehnert, 1977) Her notable insistence is that an enquiry into the nature of knowledge that is needed to understand the world and therefore to understand natural language, and structured knowledge dominates understanding. She disagree the notation that natural language question answering should be seen merely as a front-end to a completely separate data or information retrieval process. [8] The following works are mainly within the psychology community during 1980s. While recent trend in this area is focus on evaluation task by story comprehension, for example, Hirschman’s work in 1999.

- Question answering by logic techniques

Parallel with the development of automated question answering by computer, there is another survey of work refers to take logical techniques to the analysis of this problem. It began with the proposal of “erotetic logic” by M.L and A.N.Prior in 1955 and been carried on extend to parallels in the following decades. From the Introduction to Belnap and Steel, we find such description, “We hope thus to illuminate the question-answer
situation in English in much the same way as formal logic illuminates the inference situation in English, in order to thereby contribute to our understanding of erotetic “deep structure” of natural language. [9] The fact is logicians don’t pay much attention on the practice issues in question answering, and in recent year works, their result didn’t be employed to large extend, accordingly a further effort should be made in shortening the distance between theory and practice, so that there might be an original frame of reference elicited.

Frequently asked question answering system we presented in this thesis is an important subject in the researching field of QA systems with natural language question interface to database. Our concentration to it is motivated by its wide applications in communities and problems concealed currently. A presentive example is the largest, most widely-available electronic information service, USENET newsgroups. The sheer size and distribution of the information resources makes users have difficulty knowing in which FAQ file to look for answers to specific questions. [10] Therefore, an embarrassment is leaded that although answers to users’ concerns are available there, they are inaccessible to them.

This problem is quite common in a lot of commercial networks, for instance, American on-line, which especially tend to use FAQ answering system, so an overwhelming need of modifying conventional FAQ system is driven. We did modification on the basis of searching results from general QA systems, accordingly, in the following part we will begin with attempting to obtain more background knowledge from this field.

2.2 Main Endeavors in QA Systems

Although QA systems with different applications various in design and emphasis, still there are some main concerns can be summarized to familiar with the study of QA systems. We briefly concluded five main endeavors, including Question analysis, Knowledge base preprocessing, Relevant document selection, Candidate answer analysis and Answer extraction, which cover the essentials QA system research. Such a classification is conceived on the assumption that posing natural language questions to a system which has knowledge base with a large collection of natural language texts. This is a model used for studying, and it doesn’t mean all those research problems listed here have to be included in one QA system, our system is a case in point, it doesn’t involves Question analysis and Answer extraction.

- Question analysis

When users input queries in their own way, most natural language front ends systems designed constrains to restrict inputs so that they could be standardized to easier processing way in a degree. For example, users’ inputs might be required to fit for some kind of syntactic form or terms they used are limited. Otherwise, in some systems interactive dialogue is supported, so those questions considered to be implicit by computer will be suggested or modified to an appropriate way. In our system, we allow our users to input queries in the way they like, and an Assistant module is employed to reformulate users’ query through a primitive rule base to those questions with implicit logic.
After the system confirmed the input, a further preparation will be made. If the following procession contained IR technique, the input would be thereby represented to term vectors. Beside it, a lot of question answering systems have a procedure of identifying semantic type or question type. This measure aims to identify the object of the answer been sought, say, a person, a date or a place. The most straightforward way is to check the question word, like when, what or where, whereas some of them are not always refer to a static meaning, for instance, which and what could involve a quite wide topic. To conquer this problem, various systems have made effort on it. Srihari and Li (2000) sort to a shallow parser to identify the question type and put it into the corresponding category of their question type hierarchy which is built as an extension of the MUC entity. Similar works are done by Hovy, Gerber, etc in 2001. The group of Harabagiu connects the answer type hierarchy with parts of WordNet to extend the set of possible answer types available in their system. In this thesis, we only represent the input to term vectors, while the identification of question type is researched by another group of us.

- Knowledge base preprocessing

Question answering systems normally attached with a knowledge base which is originally a large number of text collections, while the systems at the same time are expected to answer questions in real time, then some preprocessing to the knowledge base become urgently needed. So far most TREC QA systems appear to rely on conventional document indexing engines to do this. This means preparing the raw document collection into an easily accessible representation of documents. This transformation from a document text into a representation of text is known as indexing the documents. Generally speaking, transforming a document into an indexed form involves the use of a library or set of regular expressions, parsers, a library of stop list, or other miscellaneous filters, for instance, if the system employs classic IR technique, a representation of text to term vector should be taken in advance; similarly, a logical form representation of text have to be derived at first if the system requires, say, ExtrAns system designed by Molla Aliod in 1998. Further more, if the system involves natural language understanding, a preprocessing on linguistic issues, like tagging part of speech, shallow semantic analysis and parsing could be done in this stage. SRI highlight information extraction system (Milward and Thomas, 2000) is a case in point. These pretreatments mentioned here are all involved in our system.

- Relevant document selection

We can easily found an obvious characteristic that in most TREC QA systems; they usually take advantage of conventional IR search engine to perform a primary selection so that only those relevant documents or files will be retrieved to carry a further process in detail. Such an action will greatly shrink the searching area and increase efficiency as well as accuracy. In this stage, designers have to consider about two problems. One is what kind of searching engine will be used, boolean or ranked answer, because there isn't a confirmed judge shows which one of them fit best in conjunction with a QA system. [11] No matter which one we prefer, we have to face the other issue: To what extend the engine retrieve candidate documents. Restrictions on how far to retrieve will influence the final results, because a loose restriction may not able to contribute to efficiency obviously and too strict may result in misjudge. We also did the filtering to FAQ files in the beginning to shrink searching bound.
2. Candidate answer analysis

After statistical relevant document sets are retrieved, a further matching follows. To analyze texts in semantic level, the simplest measure is named entity identifier. It could recognize the objects of sentences and classify it to some defined categories, for example, classes defined in the Message Understanding Conference Named Entity Task or 25 unique word beginners of WordNet nouns. Besides, some processes in further step like syntactic analysis, part of speech tagging, trunk parsing (identifying noun groups, verb groups, some prepositional phrases) and natural language understanding are also taken. A new form of representation will be constructed and corresponding algorithm will be carried on it. Ferret, Grau, etc, (2000), for example, designed a QA system which employs shallow syntactic analysis to re-index and re-rank the documents before matching against the question representation. [12] In the system FAQ Finder by Burke, etc, it represents questions stored by WordNet to create a separate semantic similarity metric for question-matching, which could be combined with the term-vector metric to reach a final similarity score. In our system, we only analysis the candidate questions, which have been confirmed through term-vector metric, by means of NLP and shallow semantic representation to reach a final similarity score, in another word, our system more emphasize semantic similarity contribution.

- Answer extraction

This stage only refers to the systems take a series of texts as knowledge base rather than question-answer pairs. To this kind of system, final answer will be extracted from selected paragraphs or sentences by applying information extraction technique. The answer should be accordant with the expected answer type of the question, optionally, some additional constrains will also be referenced. Semantic type consistence can be satisfied by constructing synonymy or hyponymy in online lexicon such as WordNet. A typical example of follows this approach is the system implemented by Moldovan, Harabagiu, etc in 2000. Relevant paragraphs contained the correct answer type are selected, and then an answer window around the candidate is established and various quantitative features such as word overlap between the question and the answer window are used in a weighted numerical heuristic to compute an overall score for the window. [13] In our FAQ answering system, we take QA pairs as knowledge base which steers clear of Answer extraction.

2.3 Related Work

To design the QA system for the project, many specific works are involved. We can briefly classify those detailed works to three blocks, information retrieval, natural language understanding and knowledge representation. In this section, I will introduce the related work by giving a list of specific work in the three fields respectively at first, and then follows some background knowledge to each work which is summarized when I review the researches on that field.

2.3.1 Information Retrieval

Since question answering is a more fine-grained form of information retrieval, it’s reasonable to consider IR as an important technique sponsor in QA system. In this thesis, we did researches on IR and implied representative IR techniques, including Vector
Space model, K-nearest neighbour and Cosine measure, in our QA system.

![Diagram of IR techniques](image)

Figure 2 Techniques implied in IR

Figure 2 simply demonstrates the employment of the three IR techniques in our system. User’s query and questions in FAQ files are firstly represented to term vector through Vector Space model, then a single relevant file is retrieved from FAQ files by carrying out K-NN method, finally Cosine measure is used to pick out the close QA pairs from the confirmed file.

Information retrieval technique is important and widely used in artificial intelligence (AI) field, but also an often loosely-defined term in definition. An appropriate straightforward definition is given by Lancaster in IR System: Characteristics, Testing and Evaluation:

*Information retrieval is the term conventionally, though somewhat inaccurately, applied to the type of activity discussed in this volume. An information retrieval system doesn’t inform the user on the subject of his inquiry, it merely informs on the existence or non-existence and whereabouts of documents relating to his request.*

The study on employing IR technique in QA system to retrieve relevant documents in response to user queries has been developed actively from the fifties. Although it doesn’t really feedback a simple direct answer and the user has to extract what he or she desire from those relevant documents returned, it is still been considered as a significant technique to QA system. We can understand it from the following two points. Actually, in recent years IR techniques have been improved greatly, and one aspect is the accuracy of the result, in another word, the size of the retrieved relevant documents. We can find the trend is it could return the relevant passage in the document as well, so in some cases, if we are able to reduce passage in a further step, the result might be the very answer required. In this point of view, IR is indeed closely related to question answering. In addition, during the past years IR has been developed a mature methodology of evaluation, a case in point is the Text Retrieval Conference (TRECs) held by the US National Institute of Standards and Technology, and they are relevant in the sense that the current evaluation on question answering is developed on the basis of TRECs standards. In our project, the three technique used in locating relevant FAQ file and question sets selection are representative and well-developed in IR.

### 2.3.2 Knowledge Representation
Since similarity of questions lies in meanings, word forms can’t always indicate word meaning, recognizing semantic similarity requires knowledge representation. In this system, we selected the semantic system, WordNet, which provides relations between concepts, to represent knowledge. Knowledge in our system is QA pairs in FAQ files and user’s question, each question is represented as a node in the WordNet semantic space, with links to the lexical semantics of its components, therefore a shallow semantics of a term represented in WordNet can be obtained by considering the relations it connected with.

In 1985, a group of psychologists and linguistics at Princeton University get down to develop a lexical database along directions existing investigations suggested. The original idea was to provide assistant for looking up dictionaries conceptually, rather than alphabetically only, which was going to be used in close conjunction with a conventional online dictionary. As the work went on, they found it demanded a more ambitious formulation of its own principles and goals, so WordNet was introduced. [14] WordNet is a dictionary based on psycholinguistic principles.

Words in WordNet are divided into four categories, which are nouns, verbs, adjectives and adverbs. The function words are omitted on the assumption that they are probably stored separately as part of the syntactic component of language. (Garrett, 1982) We benefit from the syntactic categorization on WordNet that fundamental differences in the semantic organization of these syntactic categories can be clearly showed and systematically exploited; although it may leads redundancy that conventional dictionaries could avoid, say word “name” exists in more than one category. Essentially, the most ambitious feature of WordNet, however, is its attempt to organize lexical information in terms of word meanings, but not word form.

- Lexical matrix

*The start of lexical semantics can be considered as the mapping between word forms and word meanings.*

Miller, 1986

The initial assumption is that different syntactic categories of words may have different kind of mappings. Generally speaking, there are two ways to describe WordNet by psycholinguists, namely lexicon matrix and box-and-arrow. Box-and-arrow represents words by two boxes which labeled “Word Meaning” and “Word Form”; arrows with two directions between the boxes indicates that we begin with concepts in mind and try to express them in right form, or the form is known as a start then we look for a proper meanings. Analysis denotes this kind of representation has the advantage of considering the difference between word meaning relations (in the Word Meaning box) and word form relations (in the Word Form box), but still it has weakness that many mapping in details between meanings and forms are implicit. So here we tend to introduce lexical matrix to describe WordNet.

By the following table, we can get a direct impression of a lexical matrix notion.
Table 1 Lexical metrix of WordNet

In the table, row headings are made up of word forms \( F_i \) while column headings are word meanings \( M_j \). In the matrix, each crossing presents one mapping, in another word it implies the form in that volume can be used (in an appropriate context) to express the meaning in that row. [15] If more than one crossing accrued in the same row, it means these words are with different forms but of the same meaning, namely synonyms, say, \( W_{11} \) and \( W_{12} \). Similarly, words in the same word form heading are polysemies. Actually synonymy and polysemy are problems that arise in the course of getting access to the right information in mental lexicon. We can imagine that when a writer or speaker was going to express certain meaning, he has to face the problem of selecting a word among synonymies; when the reader or listener received a sentence, he has to identify the meaning of polysemies in that context.

The reason we introduce lexical matrix above is to make the notion of WordNet concrete, so secondly, we will discuss how word semantics are represented in WordNet by simulating a lexical matrix. Two problems have to be solved in representing, word forms and word meanings. Inscriptions can provide a reasonable satisfactory solution for forms, while how to represent meanings is worthy of discussing.

Recall the representing method, definition, developed by lexicographers, to some extent it could be accepted as a simulation of word meanings played in human mind. While it is not always easy to illustrate concepts exactly with rich information, and there is some reason to believe that the definitions found in most standard dictionaries do not meet this requirement. [14] As to a lower require, it is feasible to represent a meaning with symbols which enable one getting a desired mapping of the known concepts. The one only wants to identify the lexicalized definition of that concept, so the synonyms could be a good indicator here. For example, one may know the light refers to a source of light or opposite of heavy, so the synonym sets \{light, lamp\} and \{light, ethereal\} provided will be quite helpful to meet the user’s desire. We can find it doesn’t explain the concept actually but displays the existing words with same concepts, however, the user who has known the concepts are expected able to identify them from the listed sets.

A lexical matrix, therefore, can be represented for theoretical purposes by a mapping between written words and synonym sets.

If there is no proper synonym for some word, we can use a concise note instead. For example, \{light, (Electromagnetic radiation of any wavelength)\}. Pay attention there is a few differences from synonym; it doesn’t link the unknown concept with known but make the user differentiate it from the other confusing concepts.

- Semantic relations
WordNet is organized by semantic relation; we can describe the notation by the figure 3.

As we discussed before, the key of semantic representation is word meanings, and meanings can be represented as synonym sets for the theory purpose, so WordNet can be represented by synonym sets relations. The characteristic of semantic relations are reciprocated: if the relation between meaning \{x, x', ..., \} and \{y, y', ..., \} is \textit{R}_1, there will also be relation \textit{R}_1' between \{y, y', ..., \} and \{x, x', ..., \}. In figure 3, relation \textit{R}_n is the general designation of meaning relations between synonym sets as well as form relations between the words in the sets. Relations \textit{R}_n which creates the WordNet can be understand from the main components of them, namely Synonymy, Antonymy, Hyponymy, and Meronymy. We explained those terms by reviewing some relevant documents.

1. Synonymy

   It is reasonable the most important relation for WordNet is the similarity of meanings. One definition of synonymy is two expressions are synonymous in a linguistic context \textit{C} if the substitution of one for the other in \textit{C} does not alter truth value. It is necessary to define it in the range of certain context, take an example, \textit{light} and \textit{ethereal} are synonyms when the context is about weight, but the substitution in the other context may change the truth value. Substitution was mentioned in the definition which determined synonyms should be in the same part of speech, in another word, words in a synonym set are from the same part of speech category. The use of synonym sets to represent word meanings is considered with psycholinguistic evidence that nouns, verbs and modifiers are organized independently in semantic memory. [16]

2. Antonymy

   For most of time, the symmetric part of word \textit{W} can be expressed as \textit{not-W}, but sometimes not, say, \textit{smart} and \textit{stupid} are antonym but not smart doesn’t means
stupid. Another thing is on the premise of “the symmetric part of word \( W \) can be expressed as \( \text{not-}W \) for most of time”, antonyms are lexical relations of word forms rather than word meanings or synonym sets, although both word form and word meaning are of lexical semantic relations.

(3). Hyponymy
A concept represented by the synonym set \( \{x, x', \ldots\} \) is said to be a hyponym of the concept represented by the synonym set \( \{y, y', \ldots\} \) if native speakers of English accept sentences constructed from such frames as \( x \) is a kind of \( y \). Hyponym can be described by a kind of hierarchical structure, and as to expression, the relation can be represented by including in the subordinate synonym set a pointer to its super ordinate, and including in super ordinate pointers to its hyponyms. Figure 4 shows the structure of hyponymy.

![Figure 4 Structure of Hyponymy](image)

(4). Meronymy
It is a part-whole relation between word meanings. A concept represented by the synonym set \( \{x, x', \ldots\} \) is a meronymy of the concept represented by the synonym set \( \{y, y', \ldots\} \) if native speakers of English accept sentences constructed from such frames as \( x \) is a part of \( y \).

In a word, WordNet is a hierarchical architecture providing a system of relations between words, synonym sets and synonym sets themselves; it provides the underlying semantic framework for the FAQ system. Figure 5 is an example which depicts how to represent the term vector of a question in WordNet.

![Figure 5 Represent a question in WordNet](image)
This is a simple introduction to WordNet from the overall structure. In next chapter, a detailed description on how it organizes words and how to connect terms in semantic level will be presented.

### 2.3.3 Natural Language Understanding

Natural language understanding promises to make it easier for people to find desirable texts and improve *precision* as well. It’s also a significant feature in our system.

- We propose two natural language understanding techniques to identify terms’ part of speech.

It is due to the organization of WordNet. On one hand, because WordNet is organized by words’ part of speech, the entire system is split to several subsystems where terms with same part of speech are collected; on the other hand, it’s common that one word has more than one part of speech when it appears in different context, for example, the word *book*, it could be either a noun or a verb, hence it forms a problem that when we represent such a term in a question to WordNet, we have to decide in which subsystem of WordNet to locate it, at this moment, a part of speech identifier is called to recognize the it in a certain context. In this thesis, we introduced two natural language understanding techniques, parsing with augmented transition network grammars to analyze the roll of the term and an existing Part of speech tagger Xerox. Both of them are analyzed and the further selection will be performed after implementation.

- We introduced Marker-passing algorithm over WordNet to capture semantic distances between questions.

As we mentioned, the distance of two terms in the semantic network suggests the semantic similarity, which is also the gist to do the further semantic matching of two sentences. In this thesis, we propose to use Marker-passing algorithm to draw the distances between terms in WordNet. It can be considered as we makes the system has the ability of feeling the semantic relevancy between terms in user’s question and stored questions, then it’s possible to give an exact semantic similarity score of two questions.

When I review the researches on natural language understanding, I found languages are actually studied in several different academic disciplines. In the field of computational linguist, the main task is to develop a computational theory of language, using the notions of algorithms and data structures from computer science. [17] Typical questions like “How is the structure of sentences identified”, “How can knowledge and reasoning are modeled” and “How can language be used to accomplish specific tasks” are the corn issues in research. Two important application classes of natural language understanding are text-based applications and dialogue-based applications. Text-based applications involve the processing of written text, such as books, newspapers, documents, manuals, e-mail messages, and so on. Reading-based tasks, for example, searching desired documents on some subjects, extracting abstracts of given articles or communication which includes spoken language and keyboards interaction. Question answering systems over internet or telephone, spoken language control of a machine like voice control of a VCR or computer are of this class.
Obviously, such a significant background in QA system indicates natural language understanding is an un-omitted component in FAQ answering system, so in this part essentialities, problems and basic principles of natural language understanding in QA system will be introduced in a further step.

- **Essentiality**

Since vast knowledge is available online, the manner of searching information via internet becomes quite ordinary. Besides professional experts, common users are getting used to this way, so an easy grasped means to communicate with computers is urgently called. Researches on natural language interface which takes the most familiar method to get information in daily life are thereby pursued, especially to systems which emphasize dynamic interaction between users and machines. In this thesis here, we also provide such kind of interface to provide a flat between our customers and company, and to problems introduced by natural language, we believe some more measures should be taken.

There are some question answering systems use simple matching techniques, say, keyword-based technique and statistic method to retrieve relevant messages. Syntactic parameters including word order or location and similarity are employed to rank or further filter the candidate questions. Such shallow processing which didn’t take semantic factors into consideration for sure has advantages, it’s easier to implement and response time should be faster, in addition, it indeed works well on some specific cases, for example, seeking factual tidbits of information such as names, dates, locations, and quantities. [18] However, to most natural language question answering systems, especially those to service customers in companies websites, they hope their users or customers could pose questions without too many restrictions, and because of the complexity and variety of language itself, we have to introduce more sophisticated NLP techniques to get query semantic concepts and transform various expressions to equivalent simpler questions.

A few significant problems involved semantic understanding in natural language QA system are summarized below. Firstly, questions with similar syntactic structures may emphasize totally different aspects, but system doesn't connect them to relate background will not tell the differences. Secondly, morphology, synonymy, homograph and so on are common linguistic phenomenon, while they are also common leads of misjudge in information retrieval if we don’t take any specific NLP technique. Thirdly, most languages don’t specify the corresponding relations between adjectives and nouns if more than one adjective exist. For example, pretty Jean’s friend, it can be interpreted as both Jean is pretty and Jean’s friend is pretty. All of them suggest NLP is absolutely necessary.

- **Difficulties**

Communicating with natural language greatly depend on our knowledge and expectations within the domain of discourse. Understanding language is not just a simple literal translation; it requires inferences about the speaker’s aim, emphases, emotion as well as the context of interaction. Implementing a natural language understanding program requires that we represent information and intention contained to some processable manner or model for further internal disposing by computer. We
must consider such issues as non-monotonicity, belief revision, metaphor, planning, learning, and the practical complexities of human interaction.

Three major attributes of natural language which lead difficulties in NLP are: firstly, a large amount of human knowledge is required. It is quite common in human language that even a simple short sentence might refer to complex relationship and backgrounds. We can only catch the meanings exactly by taking related knowledge into account; secondly, language is pattern based. Communicating is on the basis of some constrains on components of sentences, say regulations of combining phonemes, words and phrases. It is doesn’t assembled randomly. Thirdly, language acts are products of agents, either human or computers. Agents are embedded in complex environments with both individual and sociological dimension. Language acts are purposive.

- Principle

Although natural language understanding programs vary from different purposes and applications, they have similar principles and processing stages. All of them have to translate the original sentence into an internal representation of its meaning. Generally, depend on George F Luger’s research [19] natural language understanding follows steps below. We will give a concise explanation to each step, and extract a primitive idea for the Semantic Matching module in our system. Figure 6 shows factors in natural language understanding.

![Figure 6 Primitive flow chart of natural language understanding principle](image)

(1) Input and output

Input can be a question or a story and so on; it depends on what function of the system, e.g. a front end for a database, an automatic translation system, a story understanding program, etc. The corresponding outputs are sent to, say, question answerer, database query handler, translator, etc. In our project, inputs are tokenized questions term vectors proposed by production users and outputs are relevant retrieved question-answer pairs from database.

(2) Parsing

The input sentence is parsed by syntactic structure to form a frame for semantic analysis. The main task in this procedure is to identify the major linguistic relations like
subject-verb, verb-object and the part of speech of terminals. It is usually represented by a parse tree. In this project, we also have to make syntactic analysis because we need to identify parts of speech of key terms which is required by the way we represent knowledge: words in WordNet are organized by part of speeches. Figure 7 shows an example of parsing.

![Parse tree of “Tom beats Jerry.”](image)

(3) Semantic interpretation

By employing the linguistic structure provided from the former stage and the knowledge on the words meaning, we can construct a conceptual graph which could show the roles of words. This stage can also perform semantic checks, for example it can examine whether the verb could match with the object semantically. The straight forward word sense can be found in on-line lexicons. In our system, we recurred to a well known on-line lexicon, WordNet, to perform a shallow semantic interpretation. Figure 8 is an example of interpreting conceptual graph.

![Conceptual graph of the sentence](image)

(4) Contextual knowledge interpretation.

In this step related knowledge from the database are called to expand the literal meanings of sentences. It results representation of natural language meaning and can be used by the system for a further processing. In our system, WordNet we employed records not only meanings of words but also the various semantic relations between synonym sets. Such a characteristic helps to carry out a extending of knowledge.
interpretation. Figure 9 shows how to perform a contextual knowledge interpretation to the last example.

After the analysis to sentence, we are supposed to understand user’s query in semantic level. In next step, an algorithm could be designed to compare the semantic similarities between questions.

2.4 A Background of Introducing Searching Assistant

In early days, information seekers didn’t search information themselves but through an intermediary who were trained and normally knowledgeable about seeks’ interest area, so by communicating, the intermediary could get what he or she need and perform the actual search, of course, with some skills and experiences, in the end send the feedback to the seeker. During this phase, problems on searching skills are implicit because of human intelligence, while as artificial intelligent searching system greatly developed in recent years, users are much more freely in searching information they desired, thereby conflicts between strict searching principle and casual seekers’ unfamiliar to engines as well as inadequacy in searching skill become more and more obvious. It is an embarrassing complexion that the high accuracy designer pursued hardly all the time is affected by terminals’ personal affairs, so some kind of searching assistant is called to help those inexperienced seekers to accomplish their objects.

Studies of user behavior have indicates that merely providing end-users the function to search is not enough, assisting them to grasp the tactic of searching seems necessary. The study found that whereas system mechanics are rarely a problem for any but very inexperienced and infrequent users, even experienced searchers have significant problems with search strategy and output performance. (Borgman, 1986) Another research found experienced searchers lost sight of the search logic, missed obvious synonyms, and searched too simply. (Fenrichel, 1981) In one study, a quarter of the objects were unable to pass a benchmark test of minimum searching skills. (Borgman, 1986) In an experiment contrasting the searching of novices versus experienced searchers, the novices found some relevant documents easily, but they failed to achieve high recall and were unable to reformulate queries well. (Oldroyd, 1984)
experienced searchers in the study were more persistent and willing to experiment than the novices, which enabled them to achieve better search results.

All these problems could lead a poor performance of searching although the designer might have employed quite complicated algorithm to improve the accuracy. We have looked over a few existing searching assistants including Shoval (1985), IOTA (Chiaramella & Defude, 1987), RUBRIC (McCune et al, 1985 & Tong et al, 1987) etc. Generally speaking, they all recur to an expert system of rules to help to formulate the inquiry, analyzes the retrieved test, reformulate and rank the search results. For the sake of avoid the negative results from users’ personal abilities we were suggested by those developed ideas and thinking of introduce a searching assistance to help the users to formulate queries properly. We can carry out some modification to users’ queries from our designer’s point of view so that an easier and exact information retrieval could be achieved, and a human expert will answer questions which are failed to answered by system to increase customer trust.
Chapter 3 Principle and Algorithm

In the past two chapters, we have given a general introduction on the overall structure of our FAQ answering system, and presented the background knowledge involved, which aims to illustrate the idea, essentiality and applications of the system. In this chapter, we will describe the system principle and algorithm in detail.

The system designed in this project is made up of five functional modules to carry out an immediately natural language access to large volumes products information. The friendly interface accepts user’s query in natural language, and then the text is represented to term vector in the first module, Tokenizer. After that, the File Retrieval module, Statistic Matching module and Semantic Matching module performs a sophisticated matching between user’s question against structured FAQs in the knowledge database which were generated together with corresponding answers in advance by products engineer, finally, the closest QA pair to user’s desire is retrieved and presented, at the same time, the system stop running as has accomplishing the answering task. If the system failed to retrieve a question-answer pair, the Assistant module will be triggered, and a rule-based question reformulating assistant will restructure user’s question for a second searching cycle, hopefully, this modified query is a better representation of the real user need. In case that either the matching failed again or the user is not satisfied with the answer in the second attempt, a human expert will assist to give a final solution. For a clear description, the system principle will be depicted by illustrating the principle and algorithm of these five functional modules respectively.

3.1 Tokenizer

In IR fields, using term vector model to represent information in text is normally taken as a basic first step. User’s query in natural language is inputted into the system, and then the Tokenizer module takes every word as a feature in the vector. Because of the huge dimensionality of this vector, a reduction of the vector dimension by removing function words and stripping suffixes is needed to improve retrieval effectiveness.

In this module, we only perform a temporary operation to standardize input text and reduce vector size, and a further processing will be taken in the second and third modules. Two tactics are implied here. Firstly, remove function words by filtering the sentence with a function word blacklist. Such a noisy filtration has been proved works well in reducing vector dimensions without an obvious damnification in sentence meaning. Then, term stemming, which is a technique to provide ways of finding morphological variants of search term, is carried out. To implement this function, we employ the Porter stemming algorithm, which is proposed by M.F Porter in 1980 and is a process for removing the commoner morphological and inflexional endings from words in English. It is frequently used as a part of term normalization process that usually done when setting up IR system. [20] In this system, we mainly utilize its operation of stripping suffixes of terms. It is worth to mention that the application of stripping suffix is constrained by some rules, because it doesn’t always work to simply remove a letter set looks like a suffix. For example, the word *Wand* and *Wander* refer to totally different meaning. Since this well known algorithm has been developed well for long, we don’t repeat it here anymore. The output of this module is a term vector of...
user’s query only has meaningful terms included. The processing in this module is showed by figure 10.

3.2 File Retrieval

The input of this module is tokenized user question, in form of term vector

\[ [t_1, t_2, t_3, \ldots \ t_{i-1}, t_i] \]

where \( t_i \) is the \( i^{th} \) term of the query.

The main function of this module is to compare user’s question with questions in FAQ files and Pick the relevant file with same subject. The problem in this module can be interpreted as designing a classifier to categorize the input to a potential file. After considering many standard classifier, we choose K-nearest neighbor classifier, which deems a neighbor as statistically nearest if it has the smallest Euclidian distance in feature space, it means term vectors of questions in the same direction indicates close relevancies.

3.2.1 Weighting Terms

Before implement the algorithm, an action on weighting terms importance in the term vector have to be taken. Many weighting schemes have been proposed, for example, TF-IDF method, Relating term precision to term frequency method, Term discrimination method, etc. In our system, we make use of TF-IDF method which has a long history in term vector metric field to pick out the terms with high significance to perform the further comparison and classification.

TF-IDF method measures terms in the vector and assign weights, which denote importance, to terms.

\[
\text{Term weight } w_i = tf_i \times \log \left( \frac{D}{df_i} \right)
\]  

(1)

Where

- \( tf_i \) = term frequency a term \( i \) occurs in a question
- \( df_i \) = number of questions containing term \( i \) in a range
- \( D \) = number of questions in the range
We can find from the equation that it assigns weights to terms by considering two factors: local information from individual questions and global information from collection of questions. In one hand, \( w_i \) increases if a term existed in the question for several times, which means to a certain sentence the occurrence of a term effects its weight. Because questions with many words would have a high term frequency, the frequencies must be normalised in order to compare with other questions. To normalize the words, each word frequency has to be divided through the number of features in the text. In another hand, \( w_i \) increases as \( df_i \) decreases. Formally we call \( \log(D / df_i) \) in equation (1) as inverse document frequency (IDF), which is a measure of the sheer volume of information (and entropy) associated to a term within a set of documents. [21]

It denotes that common terms exist in quite a lot questions will receive a low weight, and terms appears in few questions will get a high weight. If a term appears only in one or two question it will be given a maximum weight because it means this term is so representative to the question that we can distinguish the question from the other by this single term. But we can’t make some irregularities exclusive, for instance, the term only occurred in one question might be noise too, say, misspelling, as we know it is to some extend impossible that a meaningful term only appeared in one question. We can try to decrease noises by means of minus those terms with extreme weight, for example.

After weighting the terms in questions from local and global aspects, we can then extract the text to a vector of important terms by defining some criteria to select the terms with high weight.

To the term vector of user’s question \([t_1, t_2, t_3, \ldots, t_{i-1}, t_i] \), the weight of the \( i \)th term \( t_i \) is expressed as \( w_i \), the value is calculated as the formula below:

\[
\begin{align*}
w_i &= tf_i \times \log(D / df_i) \\
&= \frac{\text{occurrence of } t_i \text{ in user's question}}{\text{number of terms the user's question contains}} \times \log \frac{\text{the total number of all the questions in FAQ database}}{\text{number of questions in FAQ database containing term } i}
\end{align*}
\]

(2)

The questions in FAQ files have to be represented in this way accordingly. To improve the running efficiency of our system, the transformation of questions in database to term vector model can be carried off-line.

We assume there are \( m \) FAQ files contained in our FAQ database and the weight of the \( i \)th term \( T_i \) in a question in the \( n \)th FAQ file is expressed as \( w_{T_i(n)} \), the value is calculated as formula (3)

\[
\begin{align*}
w_{T_i(n)} &= tf_i \times \log(d / df_i) \\
&= \frac{\text{occurrence of } T_i \text{ in a question in the } n \text{th file}}{\text{number of terms the question contains}}
\end{align*}
\]
\[
* \log \frac{\text{the total number of all the questions in the } n^{th} \text{ file}}{\text{number of questions in the } n^{th} \text{ file containing } T_i}
\]

(3)

Note that all the term \textit{question} we mentioned in the formulas above refer to tokenized term vector rather than original typed question. Figure 11 shows the entire procedure to represent a sentence to a feature vector.

**3.2.2 K-NN**

The K-nearest neighbour method is a supervised learning algorithm where the result of a new classification to query is based on older texts.

To classify a new question, in another word, to classify user’s question to one of the categories (files) among several FAQ files, the Euclidian distance between the user’s question and each already classified questions, is calculated. The distance is defined as an index of similarity so that the minimum distance is identical to the maximum similarity. Then the questions are ordered in order of the Euclidian distance. The K means that all the K nearest questions will be listed. The category with the maximum occurrence out of the K texts is calculated and used to classify the new question.
The Euclidian distance between to p-dimensional vectors \( x = [x_1, x_2, .., x_p] \) and \( y = [y_1, y_2, .., y_p] \) is defined as:

\[
D(x,y) = \sqrt{(x_1 - y_1)^2 + (x_2 - y_2)^2 + ... + (x_p - y_p)^2} \tag{4}
\]

Example (k=5):
User’s query: \{battery, type, rechargeable, mp3,…..\}
We firstly represent user’s question and questions in FAQ files in the vector space, dots in Figure 12 shows all the possible palaces of those questions, and the yellow dot is the demonstrates user’s query. If we set k=5, five questions with the top five shortest Euclidian distance (marked with red colour) were retrieved and listed in table 2. Because the retrieved questions don’t always in the same category, the category that occurs most, is used.

**Table 2 Euclidian distances to an example question**

<table>
<thead>
<tr>
<th>Question</th>
<th>Distance</th>
<th>File category</th>
</tr>
</thead>
<tbody>
<tr>
<td>Is that rechargeable battery or …</td>
<td>20</td>
<td>Mp3 player battery</td>
</tr>
<tr>
<td>How long to charge battery…</td>
<td>15</td>
<td>Mp3 player battery</td>
</tr>
<tr>
<td>How many battery …</td>
<td>11</td>
<td>Mp3 player battery</td>
</tr>
<tr>
<td>How long to charge the battery for the first time…</td>
<td>6</td>
<td>CD player battery</td>
</tr>
<tr>
<td>Is that lithium battery or zinc …</td>
<td>3</td>
<td>Cassette recorder battery</td>
</tr>
</tbody>
</table>

We can see from the table, three out of five competitive questions belong to the file, namely Mp3 battery, and then this file is selected to execute the further matching in the following modules. We can find that searching area greatly shrinks.
3.3 Statistic Matching

In this module, we are going to perform a further matching by computing the similarity scores between users require and each question in the confirmed file. This is a score describing the similarity between two questions from the statistic point of view. The idea behind is based upon the assumption that we could consider two sentences as the same, or at least relevant meaning if they showed great consistent in term statistical attributes. Finally, a few QA sets with similarity scores higher than a threshold in that file will be transferred to next module.

3.3.1 TF-IDF

Similar with in the last module, some preparations have to be made. Questions in database have been processed to term vector model off-line, while tokenized user question has to be weighted again since some parameters of the weighting formula have changed in this step. TF-IDF method is employed again, and we present the formula below:

\[ w_i = \text{tf}_i \times \log(D / \text{df}_i) \]

\[ = \frac{\text{occurrence of } t_i \text{ in user's question}}{\text{number of terms the user's question contains}} \times \]

\[ \log \frac{\text{number of questions in the target file}}{\text{number of questions in the target file containing term } i} \]

(5)

Target file in the formula refers to the FAQ file we selected in File Retrieval module.

3.3.2 Statistic Similarity Calculation

In this module, we still employ a vector based matching to retrieve desired information. Cosine measure is taken here. Cosine of the angle between the vectors representing the document and query, and questions in the same direction are closely related.

Simply speaking, for two vectors \( d \) and \( d' \), the cosine similarity between \( d \) and \( d' \) is given by

\[ \frac{d \times d'}{|d||d'|} \]

(6)

where \( d \times d' \) is the vector product of \( d \) and \( d' \) calculated by multiplying corresponding weights together. The cosine measure calculates the angle between the vectors in a high-dimensional virtual space, here, for an easier illustration, we give the formula on the assumption that the vector we will discuss are consist of two terms, in another words, two dimensions. (figure 13)
For an easier illustration, we give the formula on the assumption that the vector we will discuss are consist of two terms, in another words, two dimensions. We assume user’s question vector is \( \mathbf{d_1} = [x_1, y_1] \) and one of the question \( \mathbf{m} \) in QA sets is \( \mathbf{d_2} = [x_2, y_2] \)

- Compute DOT products
  \[
  \mathbf{d_1} \times \mathbf{d_2} = x_1 * x_2 + y_1 * y_2 \tag{7}
  \]

- Compute vector magnitudes (Euclidean distance)
  \[
  |\text{Question m}| = x_1^2 + y_1^2 \tag{8}
  \]
  Similarly, |user’s query| is computed as well.

- Compute cosine of the angle between vector representing the user’s question and the vector representing the stored question \( \mathbf{m} \).

  \[
  \text{Cosine angle} = \frac{\mathbf{m} \cdot \text{user’s query}}{|\mathbf{m}| \times |\text{user’s query}|} \tag{9}
  \]

All the questions in the opted files are computed in this way, the one with a high cosine indicates a good match to the input.

Then we give an example of 5 dimensions to calculate the similarity score depend on the formula above. Let \( \mathbf{d_1} = [2,1,1,1,0] \) and \( \mathbf{d_2} = [0,0,0,1,0] \)

\[
\begin{align*}
\mathbf{d_1} \times \mathbf{d_2} &= 2 \times 0 + 1 \times 1 + 1 \times 0 + 1 \times 0 + 1 \times 0 = 1 \\
|\mathbf{d_1}| &= \sqrt{2^2 + 1^2 + 1^2 + 1^2 + 0^2} = 2.646 \\
|\mathbf{d_2}| &= \sqrt{0^2 + 0^2 + 0^2 + 1^2 + 0^2} = 1 \\
\text{Similarity} &= 1/(1 \times 2.646) = 0.378
\end{align*}
\]

We can find from this whole process that we compare two sentences totally by using a statistical idea which doesn’t refer to the understanding of words meaning at all. Actually, TF-IDF has a long history in information retrieval and has been proved works well especially to the comparison of long texts. It is reasonable that the accuracy must be improved when the number of statistical objects is large. This attribute remind us to consider about the applicability of statistic means in our project as normally inquires are short sentences. The advantage of Cosine measure is it’s a straightforward ranking with simple query formulation so that pretty effective.
After all the similarities are calculated, the problem followed is a decision has to be
made on how many candidate questions could be passed to the next module. Our
measure is to give a passing threshold, questions with similarity scores higher than the
threshold will be selected. While the threshold could be empirically determined when
implementing it, a too strict threshold may increase efficiency but also may miss useful
information; a too loose threshold may reluctant to show the effect of this module. We
believe a medium or a little bit loose criterion will be ideal since we have a more
advanced module followed.

3.4 Semantic Matching

Our system is to retrieve the relevant QA pairs automatically. Statistic similarity we
discussed above is one parameter in matching; it relies on the global statistical
properties of large documents and large queries to ensure the pertinence, which
indicates an inability in word meaning analysis, so when it comes to consider lexical
content varieties, this parameter only can not achieve the performance we desire.

Take the following two inquires as example:
What is the optimal way from Finland to Sweden, by air or by sea?
- Whether fly to Sweden from Finland is better than by ship?
- In which seasons people in Finland and Sweden need to use air cooler?

The previous modules are supposed to retrieve the second and the third question as
relevant questions to user’s query, and they probably be given a same similarity score in
statistical point of view, but it’s easy for us to find that only the first one is what we
want. Such a judgment is easily achieved by our intelligence, we can easily identify the
synonyms, homographs and structure variety between the two sentences, but computers
might lead a false judgment if they only have the statistical ability to compare words
rather than concepts. (Synonyms means two distinct words represent the same concept;
Homographs are words that can refer to two or more distinct concept*) It is quite
similar with “look” and “watch”. The allowance of inputting natural language inquiry
makes it inevitable to express the same meaning through different ways although we
tend to select the same words as whole, so a method to identify the meaning similarity
lies in text is definitely necessary.

Knowledge representation is a classic AI study concerning about how knowledge about
the world can be represented and what kinds of reasoning can be done with that
knowledge. Important issues include the tradeoffs between representational adequacy,
fidelity, and computational cost, how to make plans and construct explanations in
dynamic environments, and how best to represent default and probabilistic information.
[22] Semantic network usually be used to represent knowledge. Each node represents a
concept and arcs are used to define relations between the concepts. The detail about
employing semantic network will be talked over later. In this system, we also have to
fix the contradiction that the tradeoff of representation depth and coverage breadth. In
view of the effect we desired, a quick assistant response to various inquiries is important,
therefore a shallow lexical semantics is considered as an ideal level of knowledge
representation for the system. This kind of representation has three important
advantages here: [23]

- It provides critical semantic relations between words
It doesn’t require expensive computation to compute relations;
- It is readily available.

A shallow semantics can provide us a system contains the relationship of words and their synonyms, the connection among synonyms as well. But as the name indicates, it couldn’t perform a deeper inference, like to process a causal relation or other logical deducing.

An example of semantic network is WordNet, an online lexical reference system whose design is inspired by current psycholinguistic theories of human lexical memory. English nouns, verbs, adjectives and adverbs are organized into synonym sets, each representing one underlying lexical concept. Different relations link the synonym sets. [24] In our system, WordNet provides the underlying framework for semantic matcher. Such a network has two purposes: to produce a combination of dictionary and thesaurus that is more intuitively usable, and to support shallow automatic text analysis. It doesn’t involve a deep semantic association but coverage basic lexical relations for people browsing. We can say that WordNet is the basis and core of the semantic analysis in our project.

In this module, we represent knowledge in a semantic system, WordNet, which provides relationships of words and their synonyms firstly and then execute marker-passing algorithm over WordNet to draw the semantic distances of terms, finally calculate question semantic similarity score by a matrix depending on the semantic distances. We will give a detailed explanation of fundamental that how to organize words in WordNet and how to catch the semantic distance between words by means of carrying out Marker-Passing algorithm, besides, NLP techniques involved in the procedure in also introduced.

### 3.4.1 NLP

Noted the characteristic that WordNet uses different, separate networks for each different part of speech (noun, verb, adjective, etc.) and homographs (mainly refer to its variety in word classification here) are inevitable, we are required to find a way to confirm the part of speech of words before implementing Marker-Passing algorithm. To solve this problem, two methods, which recur to NLP techniques, to identify part of speech are introduced.

- Parsing with ATN grammars

The tagging and parsing methods worked equally well when Burke measured their effects on precision in his FAQ finder system, so here we firstly take parsing method as the way to identify part of speech.

Syntax parsing is performed to questions. Parsing using context-free grammars is the easiest means. A legal sentence is any string of terminals that can be derived using rules which define the grammar. We give an example here. The rules below define a grammar for simple transitive sentences

```
1). Sentence – non_phrase verb_phrase
2). Noun_phrase – noun
```
3). Noun_phrase – article noun
6). Article – a
7). Article – the

The derivation of the sentence “The mp3 uses the battery” is given by

<table>
<thead>
<tr>
<th>String</th>
<th>Apply Rule Number</th>
</tr>
</thead>
<tbody>
<tr>
<td>sentence</td>
<td>(1)</td>
</tr>
<tr>
<td>non_phrase verb_phrase</td>
<td>(3)</td>
</tr>
<tr>
<td>article noun verb_phrase</td>
<td>(7)</td>
</tr>
</tbody>
</table>

The mp3 uses the battery.

We can also represent the derivation to a parse tree by making the set of rules of grammar as each node of the tree. So both constructing a derivation for the input sentence from a formal definition of grammar and a parse tree are all equally belong to parsing problem. One troublesome issue is how to determine which of many potentially applicable rules should be used at any step of derivation. This is handled by either allowing the parser to set backtrack pointer and return to the former asking situation if an wrong selection was made or using look-ahead to help determine a proper rule.

While it’s easy to find that context-free grammar define rules with only a single terminal on their left side, so any occurrence of that word would applied the rule with the word without considering context. It might allow some grammar mistakes like number disagreements or person disagreements. The context-sensitive languages form a superset of the context-free languages. Context-sensitive grammars allow more than one symbol on the left side of a rule which make it possible to define a context the rule could be applied in, for example:

- plural noun – men plural
- singular verb – bits

In the grammar above, the “singular” and “plural” provide constrains on selecting which rule to apply and ensure number agreement. What’s more we can also use context-sensitive grammar to perform checks for semantic agreement.

Although this kind of parsing method can achieve the aim of preventing some error which couldn’t achieved by context-free parsing, it has many disadvantages. The fiercely increased rule number is the most obvious one.

After researching the two means above, we are inspired to find a parsing way which keeps the simple structure of context-free grammar rules and performs some necessary contextual test at the same time. Finally, we found Augmented transition network (ATN)
parsing takes the point. It uses argumentation of context-free grammars but it describes
the notions like number, tense as features to attach to terminals and non terminals of
grammar. [25]

An example:

An example:

<table>
<thead>
<tr>
<th>sentence</th>
<th>Noun phrase</th>
<th>bite</th>
</tr>
</thead>
<tbody>
<tr>
<td>Noun phrase:</td>
<td>Determiner:</td>
<td>Part of speech: verb</td>
</tr>
<tr>
<td>Verb phrase</td>
<td>Noun:</td>
<td>Root: bite</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Number: plural</td>
</tr>
</tbody>
</table>

Figure 14 Frame like structures of some non-terminals and terminal

The derivation of ATN is quite similar with transition network. A transition net work
parser represents a grammar as a set of finite-state transition networks formed by
labeled states and directed arrows connecting them. (figure 15)

The parser is aim to find a path through the “sentence” network. In order to cross each
arrow, it has to check the label. An arrow could only be crossed when a matched branch
in a matched network been found. If it fails with one network, the parser has to
backtrack and try another path. The difference to free-context passing is it checks the
grammar agreement when building the parse tree. The input sentence is parsed by
syntactic structure in this way, the result of it is that the major linguistic relations like
subject-verb, verb-object and the part of speech of terminals are identified.

- **Use an Existing Part-of-Speech Tagger.**
Another method is employing Part of speech tagger. One existing Part of speech tagger is Xerox Tagger (Cutting, 1992), which tags words based on their statistically most frequent word sense. Here they make use of stochastic tools in language analysis. Stochastic language techniques view natural language as a random process. It allows us to redefine many of the basic problems within natural language understanding in a rigorous, mathematical manner. [26]

First we define the problem formally. \( S_w = \{ w_1, w_2 \ldots w_n \} \) is a set of words, and a set of parts of speech or tags are \( S_t = \{ t_1, t_2 \ldots t_m \} \)

\[
P(t_1, t_2 \ldots t_n | w_1, w_2 \ldots w_n)
\]

which stands for the probability of such a corresponding distribution.

After steps of inference, a conclusion is drew that

\[
P(t_1, \ldots t_n | w_1, \ldots w_n) = \prod_{i=1}^{n} P(t_i | t_1, \ldots t_{i-1}, w_1, \ldots w_{i-1}) P(w_i | t_1 \ldots t_{i-1}, w_1, \ldots w_{i-1})
\]

(11)

With Markov Model approach, we made some useful approximations of equation (11):

\[
P(t_i | t_1, \ldots t_{i-1}, w_1, \ldots w_{i-1}) \text{ approaches } P(t_i | t_{i-1})
\]

\[
P(w_i | t_1 \ldots t_{i-1}, w_1, \ldots w_{i-1}) \text{ approaches } P(w_i | t_i)
\]

The Markov assumption assume the present thing under consideration is in dependent of things in the far past.

\[
\prod_{i=1}^{n} P(t_i | t_1, \ldots t_{i-1}, w_1, \ldots w_{i-1}) P(w_i | t_1 \ldots t_{i-1}, w_1, \ldots w_{i-1}) = \prod_{i=1}^{n} P(t_i | t_{i-1}) P(w_i | t_i)
\]

(12)

The probability of equation (12) is easily be maximized by the Viterbi algorithm and a tagger using this method is about 97% accurate.

Additionally, we will introduce another method as option, namely Unrestricted Marker-Passing, to solve the part of speech problem. It is different with the two approaches above due to it doesn’t tag part of speech to any term at all, it takes all senses were scored if a word had more than one sense. However, this approach led to too may false matches, because this method doesn’t take any semantic or syntactic tips into account. They just treat all the possibilities in the same manner, and therefore the result can only be seen as a lucky one rather than a reasonable one, so this method is not recommended.

3.4.2 WordNet

Since word meanings or synonym sets in WordNet are organized by part of speech, here we take nouns as example to illustrate how WordNet represents knowledge and its knowledge architecture. We can then understand how the system connects two questions with totally different word forms and same meanings together to perform an intelligent matching.

Review the expression of words in traditional dictionary, a store of information including pronunciation, spelling, part of speech, definitions and illustrative uses of
alternative sense, synonyms and antonyms, special usage notes are common aspects packed into lexical entries. For nouns, we can not extract the underlying logic that primary super ordinate and its distinguishing feature from those aspects above. Generally speaking, the store of words together with its information is discrete from each other, which is not accord with the psycholinguistic principle in human memory.

For instance, the word *tree* in conventional dictionaries is defined as: *a plant that is large, woody, perennial plant with a distinct trunk*. In terms of Miller’s opinion, there are something omitted in this kind of definition. Firstly, important information about the super-ordinate term *plant* is missing from the definition, although users can find it out in the dictionary finally, it doesn’t have any direct link available to this significant term. Secondly, to distinguish the feature, curious readers may like to know its coordinates, which has no help given to be found. Readers have to look up every word from A to Z and identify the words explained under *plant*. Another similar challenge leave to users are to know something about different kinds of trees. They might be interested to know which trees are deciduous or which are hardwoods, which are indeed contained in the dictionary, but only the most determined reader would try to dig it out. So for most of them, they could only get an unsatisfied or inadequate result.

We can find that the missing information is due to structural reason rather than factual reason, in another word, lexicographer has done a good job in covering important information to each word, but the alphabetized organization leads a discrete concept for users which brings incomplete in understanding.

- Inheritance system in WordNet

Whereas the shortcomings caused by structural factor in conventional dictionary we have mentioned above could be well solved by more competent information architecture. Look into noun entries organized by lexicographer, information common to many items in the lexicon needn’t be listed in every entry; users can find that kind of information by looking up the generic terms involved in the explanation of the noun at hand, so that the super-ordinate of that term together with the newly exploited information are displayed. Below the same super-ordinate, those terms in the same level indicate coordinating relations between each other, and by comparing coordinates we can easily find the differences among them. [27] This is actually a kind of implicit hierarchical structure. What’s more, another important advantage of such hierarchical structure is that it could effectively save space.

Inspired by this idea, nouns in WordNet are organized by hierarchy which provides conceptual skeleton for them. Semantic relations between words in the system are primarily divided to two types: super-ordinate relations (hyponymy) and specialization relations. The former relation is represented by ‘@→’ in WordNet which goes from specific to generic. It can be considered as ‘is a kind of’, so the example a noun *W_h* @→ a noun *W_s* indicates *W_s* is the super-ordinate of *W_h*, at the same time, there is always an inverse relation *W_s* ~→ *W_h* means *W_h* is the subordinate of *W_s*, which represents the later relation. Computer scientists call such hierarchies inheritance system.

Although both common dictionary and WordNet have resort to hierarchy structure in organizing nouns, the distinctions in effect are obvious. Due to there are too many subordinates or specific terms corresponding to one noun, the series of terms are usually
omitted in the definition in conventional dictionary, therefore readers are not able to get a full understanding. In common dictionary, despite super-ordinates are available, because of the hierarchy structure, generic features of the current noun are assumed inherited from its super-ordinate and are not listed anymore, but users couldn’t get any assist from its super-ordinate term if they don't understand that term either, then users have to look up again. While in WordNet, lexicographers code the generalization relation ‘@’ and specialization relation ‘~’ with labeled pointer and inverse pointer between lexical sense, so the database is a hierarchy that can be searched upward or downward with equal speed and retrieve those additional information directly. [28] We can easily find that the advantage and effect of inheritance system is much more explicit in WordNet.

In WordNet entries for nouns are described in the form of synonym sets like *tree* is `{tree, plant, @ conifer, ~ alder, ~…….}`, *plant* `{plant, flora, organism, @ tree, ~…….}`. The computer is programmed to use the pointers to extend to whatever information a user request, in our project, the user is us and we aim to connect the nouns in input queries to corresponding nouns in the FAQ file. When the terms are matched, or our request is met, the symbol @ and ~ are suppressed.

As we mentioned before, WordNet is inspired by human lexical memory and designed on the basis of psycholinguistic fundamental, so the heritance organizing system of nouns should be consistent with human mind. Actually there are some evidences to support it.

(1) The study on anomic aphasia by Caramazza and Berbdt in 1978 showed that the semantic memory for nouns of those patients seems disconnected from the other lexicons. This observation denotes that the isolation of nouns to a separate subsystem is reasonable.

(2) The hierarchical organization of nouns is accordant with human mental. This point of view is illustrated by Bever and Rosenbaum through two examples. Firstly, super-ordinate nouns can serve as anaphors referring back to their hyponyms, say, *she wants to buy the novel, but the book has been sold out*. It is natural for us to take the *book* as referring the former *novel*. Secondly, super-ordinates and their hyponyms cannot be compared in the same level, like *novel* is more interesting than *book* is easily to be considered as semantic error.

(3) The tests about people’s reaction time to words located in different hierarchy by Quillian in 1969 proved that in human mental generic information is not stored redundantly, but is retrieved when needed, which is the idea of heritance system used in organizing nouns in WordNet. While there are also some belief insisted by many psycholinguists that English common nouns are organized hierarchically in semantic memory, but whether generic information is inherited or is stored redundantly is still moot. (Smith, 1978)

Anyway, we can make a conclusion that hierarchy organization and inheritance system of nouns in WordNet is simulation of human memory to a large extend, and therefore is competent to be taken as toll to process human natural language. [29]

- Nouns partition: Semantic component
Depend on the attributes of inheritance system and hierarchy organization, the word in the top level is suppose to contain the most generic feature which is inherited by the other nouns. If it is given that all the nouns in WordNet are organized in a single inheritance system, the top word has to be provided with the common feature shared by all the nouns in the domain, which is almost impossible. So normally the whole nouns domain is divided into several semantically separated subsystems and in each of them nouns in the same field or with some kind of relations is collected to express a same concept. It is relative easier to find an appropriate word as the abstract generic term.

The following problem is to decide what these primitive semantic components should be. A straightforward criterion is they should provide a space for every English word. WordNet has adopted a set of twenty-five unique beginners. Those subsystems are various in size and number of hierarchy. Originally the depth of each subfield is not limited, but in our project it seldom do more than 10 levels deep because the further terminal indicates a more specific term, while too specific term may ran out of daily spoken vocabulary. The partition of nouns is represented in the following list.

```
{act, action, activity}   {natural object}
{animal, fauna}          {natural phenomenon}
{artifact}               {person, human being}
{attribute, property}    {plant, flora}
{body, corpus}           {possession}
{cognition, knowledge}   {process}
{communication}          {quantity, amount}
{event, happening}       {relation}
{feeling, emotion}       {shape}
{food}                   {state, condition}
{group, collection}      {substance}
{location, place}        {time}
{motive}
```

Unique word beginners of WordNet nouns (Miller)

- Relations of nouns in WordNet

As we discussed before, the structure of nouns in WordNet are called hierarchical inheritance system, so the most common relation referred is hyponymy. While focus on a hierarchy, terms are not only share the features inherited from their super ordinates but also separated from each other by its distinguishing features. For example, novel is not suppose to be defined just with the attributes of book, distinguishes like fictional, length, plot and unfolded are more significant.

Ideally, distinguishing features should consists of at least three aspects, namely attributes, parts and functions. In the novel example, attributes includes fictional and parts includes length, plot and unfolded, etc. As distinguishing feature being introduced, tiny distinctions are described, therefore synonyms might be not sufficient to depict all the details, then it was decided by researchers to include distinguishing features in the same way that conventional dictionaries do, by including short explanatory glosses as a part of synonym sets which are contained polysemous words. For example,
Glosses marked by parenthesis serve to make distinct identified, but leaded redundancy to some extent, especially to the super-ordinate term indicated by “@”. As more distinguishing features come to be indicated by pointers, these glosses should become even more redundant. One feasible means proposed by Miller is to generate glosses from the information involved by pointers through programming.

(1) Attributes

Attributes are normally adjectives used as modification to illustrate the specific natures, say, *fictional* is an attribute of *novel*. It is reluctant to include the relationship between the adjective and the noun it modified into any typical relations we have mentioned before, but hyponymy seems to be the closest sort although actually not. When we referred to *novel* we would definitely considered *fictional* as one of immediate characteristics, so *novel* seems as the super ordinate of *fictional*, but when we focused on *fictional* only, we don’t have to collect words like *novel* to describe this concept, thereby the term *fictional* is not a subordinate of *novel*. The implementation of attribute can be taken as the similar way with hyponymy, where there are labeled pointers between terms, while the distinct is it doesn’t have a reverse pointer from the adjective to the noun. Figure 16 is a fragment of WordNet to show attributes of nouns.

![Figure 16 Attributes of nouns in WordNet](image)

Psycholinguistics argued that the primitive way people learn nominal concept hierarchy is by observing what can and can’t be predicted at each level. For instance, kind and evil are used to modify person rather than common animal. This idea is also helpful to carve up nouns to the 25 categories in WordNet.
(2) Parts
The part-whole relation between nouns is generally considered to be a semantic relation, called meronymy, following Cruse. (1986) When WordNet introduced distinguishing features in a single hierarchy, we have to decide through which point of view to emerge nouns’ features effectively. Attributes we discussed above is an important parameter, additionally, it has been approved that meronymy is particularly important for defining features for basic term, [31] especially to words of physical objects categories. It was found that meronymy is frequently happened in the {artifact}, {body, corpus} and {quantity, amount} semantic components.

Normally meronymy is acceptable as the frame “W₁ is a part of W₂”, for example, bulb is a part of lamp, so the term bulb is a meronymy of the term lamp. This relation is generally been taken as asymmetric and transitive, and relate terms hierarchically, (Miller and Johnson-Laird, 1976), but Lyons (1977) observed that transitivity is limited in some cases. For instance, bulb is a meronym of lamp and lamp is a meronym of furniture, yet we rarely say “furniture has a bulb”. By Lyons’ example, they suggest that “part of” is sometimes used where “attached to” would be more appropriate. Actually, Winston (1987) had concluded six types of meronyms which are component-object (branch/tree), member-collection (tree/forest), portion-mass (slice/cake), stuff-object (aluminum/airplane), feature-activity (paying/shopping), and place-area (Vasteras /Sweden).

Meronymy is often been compared with hyponymy which simplified as “a kind of” relations. Meronyms are distinguishing features that hyponyms can inherit. Like, bulb is a meronym of light, and lamp is a hyponymy of light, so light inherits bulb to its hyponymy lamp and lamp should has a bulb.

(3) Functions
No matter from the linguistic or psychological point of view, function is quite necessary. For many nouns, function is the best way to describe it and make it understandable. For example, although you had collected a series of adjective like color and nouns like material to illustrate the attributes and parts of a decoration, still it might be implicit for a reader to get the point until he or she was told it is used to decorate. Further more, functions of nouns sometimes play an important role in defining other kind of words. A typical example cited by Miller is the definition of good, the uncertainness of this word is due to the variety of subjects and the corresponding functions, like when we judge a pen is good the word good denotes write well, or a bike is good means ride easily, whereas it is impossible to list all the possibilities in the entry, but this problem could be solved by attach the function of nouns like write, ride to their concept definition, and made the sense of good fluctuate depend on its subject and the function attached.

The implementation can be taken in the light of the way attributes have taken. We have marked pointers there from nouns to adjectives, and similarly functions of nouns are normally verbs, therefore labeled pointers can be added from nouns to verbs.

(4). Antonymy
Psychologists suggest that antonymy is the most direct reaction to some kind of words. People used to describe a noun through its opposite concept, say, happiness
is the opposite side of *pain*. In WordNet semantic opposition is not a fundamental organizing relation between nouns, but it does exist and so merits its own representation. (Miller) For example, the synonym sets of happiness are separate \{[happiness, pain,!], feeling,\@...\} (a kind of beatific feeling) and \{[pain, happiness,!], feeling,\@...\} (a kind of bitter feeling). The symmetric antonymy in both sets are represented by the “!” pointer.

### 3.4.3 Semantic Similarity Calculation

Before semantic similarity calculation, we have to introduce an important algorithm, marker-passing algorithm, by which we evaluate intersecting paths of two terms (markers). Marker-passing is a parallel search technique based on activation theory of human semantic memory organization and retrieval proposed by Quillian in 1968. [32] This technique assumes that the target semantic network which is representation of a knowledge base contains many nodes and each node in the network is an individual processing element. Each node processes a bag of information referred to as a marker. The node receives a marker from adjacent nodes and passes it onto other nodes when finished processing. The assignment of simple processing elements to each node in the network that may operate in parallel greatly eases the computational burden of the system.

By using marker-passing algorithm, we are able to make a “text inference” throughout our semantic network, WordNet. The basic idea is they start from a node in the network (i.e., a word) and radiate outward from there to find "similar" words. Here we illustrate the algorithm by figure 17. The start node is initiated by user’s query, take the word *ship* in user’s question as example, and then it performs propagation throughout WordNet exhaustively to find the similar word in candidate questions, in the end, it get in touch with the relevant word, for instance, *ship* in stored questions. The other terms in the term vector of user’s question perform such a text inferences in the same way in parallel, finally, the algorithm would stop running when all the connections between term vector of user’s question and term vector of candidates’ questions were found.

The marker-passing algorithm allows markers to propagate along the semantic links it connected with, and this propagation will not stop if it doesn’t reach any term appeared in candidate question, at the same time, the path the propagation passed by during seeking relevant term is reported in a mechanism. The mechanism records the distance by accumulating the markers one completed connection contained, the final value is used in calculating semantic similarity score.

Back to the example we have cited before, we can also relate the phrase “by air”, “by sea” with “fly” and “by ship” through the WordNet and marker-passing algorithm, so ideally, in the end a various form of inquiries with the similar meaning will convergence to the original question we have stored in the database. However, because of the parallel inference in passing, sometimes too many potential inferences might be returned, so restricts on the propagation of marks in the network should be set to reduce the number. [33]
Back to the example we have cited before, we can also relate the phrase "by air", "by sea" with "fly" and "by ship" through the WordNet and marker-passing algorithm, so ideally, in the end a various form of inquiries with the similar meaning will convergence to the original question we have stored in the database. However, because of the parallel inference in passing, sometimes too many potential inferences might be returned, so restricts on the propagation of marks in the network should be set to reduce the number. [33]

Regarding to the variety of vocabulary and structure of text and the parallel character of marker-passing, it often occurred that more than one question are identified to be relevant to the input. At this moment, we have to check how different from those retrieved questions to users’ query and select the closest one. For this purpose, we try to educe a semantic similarity score by utilizing the principle of Marker-Passing technique to weight relativities. We consulted the algorithm in FAQ Finder system, and conclude the general idea of calculating score, that is creating a score matrix of all possible combinations of word-to-word comparisons. Each comparison results in a score based on the scaling of the path length between the two words into a range. The word-to-word comparison matrix is then reduced to a single number that measures the semantic similarity of two questions.

The similarity score of two words $u_i$ (from user’s inquiry) and $f_i$ (from question library) is given by the following equation:

$$S(u_i, f_i) = H - (p \frac{H - L}{D})$$

(13)

Where $p$ is the length of path between two words $u_i$ and $f_i$; $D$ is the maximum path length permitted by the system; $H$ and $L$ are constants that define the range of $s$. 
An Intelligent FAQ Answering System

$$S_{u,f} = \left[ s(u_1, f_1) \ldots s(u_1, f_m) \right]$$  \hspace{1cm} (14)

The single value $w(u, f)$ for semantic relatedness between user’s question $u$ and stored question $f$:

$$w(u, f) = \frac{\sum_{i=1}^{n} \max(s(u_i, f_i), \ldots, s(u_i, f_m))}{n}$$  \hspace{1cm} (15)

This value denotes the semantic relativity between two questions, similarly such a comparison would extend to all the QA pairs been selected in the last step, Statistic Matching module, and finally the stored question in QA pair with the highest semantic similarity values would be considered closely related to user’s inquiry and be retrieved.

Overlook the computing of semantic similarity, ideally, synonym problem, which is the biggest weakness in statistical similarity calculation measure is solved in the equation (13); equation (14) which contains all possible combinations of word-to-word comparisons insures a tolerance of syntactic variety between questions. Of course, WordNet provide the frame as the precondition. However, we can find that the homograph problem which may decrease the precision of answer retrieval because false matches might be made is still unsettled.

3.5 Searching Assistant

We have explained the reason we introduce this module in last chapter, in this part, we will give a further description of the principle and implementing scheme in detail. When a user query comes, it is firstly processed by the FAQ searching system. If the answer was retrieved by the computer-based support perfectly, the program would stop running; otherwise the information retrieval system will appeal to Assistant module. This situation includes two possibilities: one is the highest final similarity score is too low (less than the threshold) or even no matching answer retrieved; another is the user doesn’t satisfy with the result and require an exact answer. The unresolved user inquire is then submit to this module, and a reformulation to user’s question is performed to achieve a clear logic or proper terms, after that the reformulated user question will be processed by the system for the second time. If failed again, a human expert will answer it directly. We will give a figure in the end to depict the flow of this module after the illustrations of principles. (figure 18)

3.5.1 Rule-based Query Reformulation Assistant

To reformulate questions, the idea is like this: when a strange term or implicit logic accrued in the user query, which doesn’t have a similar term in the WordNet corresponding to, or all the queries stored have very low similarity scores, we would employ the rules in rule database of searching assistance to deduce the concept and display an existing relevant term and structure, then the user could judge the modified query and a reformulation could achieved.
In this procedure, we have to tradeoff a conflict, namely how to reformulate the query while don’t breach user’s desire. We are supposed to modify user’s input on the basis of the following belief:

- Each search term in the initial query represents one concept on which the user does, or explicitly doesn’t, want information.
- The user’s initial search terms are the best indication of the user’s areas of interest.
- Some terms from the lexicon or WordNet may be helpful, but others will not.
- The expert system should never discard concepts in which the user has indicated an interest.

There are two well developed query reformulation tactics which are expanding concept and changing query structure. [34]

- Expanding concept

Many researchers including Bates (1979) and Salton (1986), etc, had proved that replacing a term with its parent term to broaden a query is a common practice. So we consider that some rules could be defined to recur to the corresponding parent term when rare term happened in user’s question so a extending of query scope could be achieved. However, as mentioned before, user’s initial search terms are the best indication of the user’s areas of interest, and then we believe adding the super-ordinate term rather than replacing the original one is supposed to be better on string along with users.

- Changing query structure

The query can be changed in two ways: one is the AND structure in the query can be switched to OR structure, and change OR structure to AND structure; the other is the parts leaded by OR are removed. [34] For example, the question Is the green button or the red button the reset button? Can be restructured to Is the green button the reset button?

### 3.5.2 Human-based Assistant

When the user is not satisfied with the reformulated question or the matching in the second time is still not triggered, a human-based support will be utilized. Unsolved question is submit to the corresponding expert, and the expert will feed back the solution. This function is inspired by the Answer Garden system (Ackerman) which is a question-answering system involving human experts and LivePerson’s (LivePerson.com) comprehensive software platform, namely Timpani, which delivers tools that support and manage all online customer interaction: chat, email management and knowledgebase. Both of them have been successfully employed as commercial systems for human assistance in web sites.

All these procedures are of resolving queries, while the other issue is to update the FAQ items. It is reasonable that the initial generated FAQ items couldn’t cover all the common questions in the domain, in addition users’ concerned question are changing with new products’ developing, so update is suppose to be necessary. After the expert answering user’s question, two problems he has to face are deciding whether the
question at hand should be added to the FAQ file, and judging again which category it belongs to. Some criteria should be set by the company engineering depend on the content of FAQs.

Actually we found immediately that the introduction of human-based assistance leads a conflict between pros and cons. User trust might be increased by an expert’s assistance, especially in the fields of bank or insurance company. Users with queries related to decision and important personal affairs seemed more prefer to get a corresponding human voice rather than an automatically respond by computer, so for this kind of commercial website user trust is crucial. At the same time, in one hand, human assistant requires more resource cost, and to non-profit website it is an important parameter; in the other hand, to most information searchers, they just want to check things out, and they had no real need to chat with a human assistant even when they failed to retrieve answers they want. In one word, the effect of this function relies on its property of application. In our project, the company aims to answer user’s question about their products, so we believe it could be necessary to reply interested users in a more complaisant manner to get more favor and profit.

A study about collection and exploitation of expert knowledge in web assistant systems from department of Computer and Information Science in Linkoping University showed some practical problems of applying effect:

- In some cases it was difficult to decide the appropriate topic category because there were more than one reasonable alternative. In such cases we made arbitrary but consistent choices.

- They found that different experts had interpreted the criteria for when to create a new FAQ item differently. A normal standard is a general enough question should be added to the FAQ item base which is not quite explicit. So exact definition or parameter to weight the universality should be introduced.

- Questionnaire about the reasons for not having a conversation with an assistant showed that the most case is the user registered just for checking things out but not really need to chat with a human assistant (38%); the second most frequent case is there is no assistant logged in when the user used the system (29%). If the first ratio is too high we have to reconsider whether the object web is fit for have a human assistant. The second problem could be solved by recruiting assistants in different time zone to make 14 hours assistance available.
Figure 18 A flow chart of Assistant Module
Chapter 4 Evaluation of the Algorithms

In this chapter, we will give evaluations to the some algorithms employed in our system. Since this system has not been implemented yet, the evaluations below are stated on the basis of experiments in my previous project and other researchers’ conclusion, so that could give readers a sense of the effect of system. An illustration of the considerations when we design such an algorithm will be presented, besides, the fitness and limitations of those algorithms in implementing each function will also be included.

- Porter stemming algorithm

The first module of our system aims to perform a pretreatment to user’s question in natural language. After the processing in this module, sentences inputted would be transformed to term vectors, which provided a feasible format for the following measures. Since we believe that common stem normally have similar meanings, we employed Porter stemming algorithm in this module to strip suffix so that an improvement of IR performance could be achieved. The removal of the various suffixes like -ED, -ING, -ION to leave a valid stem will lead variables conflated to single term which will directly reduce the total number of terms in the IR system, and therefore reduce the size and complexity of the data in the system accordingly.

This algorithm is implemented by employing a list of suffixes attached with a rule constrains the application of the striping. Such an idea makes it a simple program to be embodied. This is also why we prefer it rather than the other stemming algorithm. The most obvious advantage of this program is that it’s small and fast which is accord with our desire here. It is less than 400 lines of BCPL and may process a vocabulary of 10000 different words in about 8.1 seconds on the IBM 370/165 at Cambridge University. [35] Another characteristic of this algorithm which attracts us comes from a conclusion after a test on the well-known Cranfield 200 collection, it suggested this algorithm shows superiority in processing relative small vocabulary text lists of an information retrieval system compared with the other even more elaborate program which has been in use in IR research in Cambridge since 1971. Questions which are short sentences are the main object in this system, hence it’s extremely reasonable and suitable to apply Porter stemming algorithm here. [36]

- Information retrieval with statistical matching algorithm

Both the second module and the third module in our system perform a information retrieval from a statistic point of view, since a standard technique, the vector space model is used in representing documents, corresponding ways to measure how similar a FAQ is to a user’s query are researched. Euclidian distance and cosine measure are very common similarity measures, we benefited from these two measures on their straightforward ranking, intuitively appealing and effective

In the second module, we hope to classify user’s question to one of the categories (files) among several FAQ files by subject. Several learning-based approaches have been applied in text classification beside K-NN, like Bayesian probabilistic approach, inductive rule learning, neural networks and decision tree, however, K-NN is still a popular text technique, and known as the top-performing method in many fields, what’s
more, some modern classifiers like kernel-based Bayesian classifiers also employs similar computational regimes to K-NN. In the processing of classifying user’s query, we employed Euclidian distance technique to locate the k nearest files and rank them with similarities. The third module has a similar principle with in the second, the difference is a cosine measure is used in retrieving close QA pairs.

A comparison of classification ability between k-NN classification methods and Bayesian algorithm which is an ambitious candidate in text classification are made by testing the classifying accuracy. I did this work in another project, Filtering spam with different approaches, with Laszlo, Marco, Santiago and Sarmad. Table 3 and table 4 shows the training and test set for a 3-FAQ files test, and the results of testing by using all the 3 categorize training and test set are showed in table 5.

<table>
<thead>
<tr>
<th>Training Set</th>
<th>User’s query</th>
<th>File 1</th>
<th>File 2</th>
<th>File 3</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>No. of questions</td>
<td>72</td>
<td>30</td>
<td>33</td>
<td>38</td>
<td>173</td>
</tr>
</tbody>
</table>

Table 3 Training sets for 3-files test

<table>
<thead>
<tr>
<th>Testing Set</th>
<th>User’s query</th>
<th>File 1</th>
<th>File 2</th>
<th>File 3</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>No. of questions</td>
<td>69</td>
<td>30</td>
<td>33</td>
<td>41</td>
<td>173</td>
</tr>
</tbody>
</table>

Table 3 Testing sets for 3-files test

<table>
<thead>
<tr>
<th>Classifier</th>
<th>Not put to file 1</th>
<th>Not put to file 2</th>
<th>Not put to file 3</th>
<th>Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>K-NN</td>
<td>9</td>
<td>3</td>
<td>6</td>
<td>86%</td>
</tr>
<tr>
<td>Bayesian</td>
<td>9</td>
<td>2</td>
<td>9</td>
<td>81%</td>
</tr>
</tbody>
</table>

Table 4 Results of the 3-files test

The results above shows that in a small categorize test, K-NN performs better than Bayesian algorithm, which means the location of relevant FAQ files and stored QA-pairs in our system could be satisfying in this case, while it’s noticeable that the more files there are, the accuracy get worse.

Pernkop Franz also made such a comparison in his research in 2005, but he generated more categories. The result was that the average classifying accuracy of Bayes was 9.2 percent higher than that of the k-NN classification method. In addition, his experiment shows that classification time of K-NN was only long when number of dimensions of text was large. To our system, we considered about the query classification is only a primitive processing as a preparation for the further information retrieval, and the kernel of our system is semantic matching, therefore, it’s not worth to implement a more competitive program than K-NN in this step to improve the temporary accuracy by 9.2 percent, in addition, since the dimension of texts (questions) in our system are small, the distinct in processing speed is not obvious between theses two methods. Therefore, it seems it’s feasible to employ K-NN here to classify user’s question, it gains victory in its easily implementing and acceptable accuracy.
Information retrieval with semantic algorithm

In the fourth module, relevant QA-pairs are retrieved in a semantic point of view. Due to the knowledge representation is performed through WordNet which is organized by word’s part of speech, ATN grammars parsing method and an existing POS tagger are introduced. In R. Burke’s similar project FAQ finder system, which also involves WordNet as knowledge framework, evaluation showed that parser and POS tagger works equally well in identifying part of speech. After that marker-passing algorithm, by which we evaluate semantic relativities of two terms (markers) is employed, and the semantic similarity score is computed to determine the final matching QA-pair. One feature of marker-passing algorithm is it works quite fast, actually, if we employed restricted marker-passing algorithm to constrain the parallel propagation, the running speed would be even faster.

It’s noticeable that the NLP involved in marking part of speech is computation consuming, which is decided by the nature of its principle and algorithm, and Burke had indicated in their FAQ finder system the parsing doesn’t contributed much to system performance, so we may also consider about using word’s default common part of speech instead. Besides, their experiments on recall for ablation study shows that semantic method only works more or less the same with statistic method only in their system on the recall evaluation; they are separately 55% and 58%. Here we quote the experiment result as some indication to our future implementation. (figure 19)

![Figure 19 Recall for ablation study][23]

System overall performance

We can find from the result above, the recall of right answer exists highest in the full system. We can deduce that the recall of our system is suppose to be higher than 66%, since our last Assistant module will run when no matching answer achieved. Both the reformulating to user’s query and expert’s involvement guarantees a high recall of right answers.

We should aware that the traditional evaluating metric to IR system, recall and percentage, doesn’t fit well to our system.
Recall=$\frac{\text{relevant documents retrieved}}{\text{relevant documents}}$ (16)

Precision=$\frac{\text{relevant documents retrieved}}{\text{retrieved documents}}$ (17)

The reason is traditional IR systems aim to retrieve all the relevant documents in a collection of documents, whereas there is only one desired answer stored in FAQ files to match with user’s query in our system, in another word, the relevancy here is 0% or 100%. The judgment to the quality of this kind of system should be the percentage of the retrieval of the right answer when it existed. The recall in the graph above also defined like this.
Chapter 5 Summary and Future Work

We have completely depicted the principle of our FAQ answering system in depth, and now we can give a summary to it in a few words. The system interacts with users through natural language, and then matches the user’s inquiry against information in database. The matching system contains five modules, which generally speaking can be departed to three parts, Information Retrieval from statistic point of view, semantically matching and searching assistant. The database which provides knowledge resource to answer question is organized by FAQ pairs in different subject categories, rather than unstructured free texts. The procedure of the whole searching is represented by the figure below. (Figure 20)

We can expect an exciting searching result by taking advantage of the powerful theory foundation. Actually there are some features designed in our system ensures not only precision and recall, but also user’s satisfaction. Here we summarize these features and give a concise analysis from an academic level.

- We combined both statistic technique and natural language understanding technique in searching target information. The K-NN information retrieval technique used in file retrieval and Cosine measure in question selection are standard IR techniques which have been developed for long and approved works quite well in document retrieval, we benefit on its easily implementation and running speed, therefore the matching in the statistical point of view is supposed to be effectively. Although these techniques are not that ambitious in principle, but we prefer its easily implementation and fast attributes, in addition, the searching effectiveness and accuracy is acceptable in this step. The kernel of our searching system, semantic matching performs retrieval in an advanced level. It fetches up the disability of statistic information retrieval by employing natural language processing and knowledge representation, linguistic phenomenon like synonym which lead mismatching before are successfully solved.

- In this system, we use categorized FAQ questions generated in advance as knowledge resource. The size and structure provides a great help in increasing efficiency and precision.

- It is easily to find that, our system quite emphasizes interaction with users during the searching procedure. The retrieved file in the second module is displayed to the user for confirm before further processing, and the reformulated user question is also sent to user before the second iteration. All the decision from users in the processing ensures a correct direction and an exact result.

- The last module, Assistant, not only increases the recall rate by reformulating user’s question, but also gains the trust of customers by means of expert’s reply. A maintain to FAQ database is performed at the same time so that the database becomes more and more consummate and updated which potentially improves the searching results.

The four features of our system guarantees an excellent performance and we are confident to expect an effective intelligent searching system applied in various
websites with public-accessible knowledge resource and aim to give a straight forward answer to users.

Due to some constrains, in this thesis we only give the principle and algorithm to the FAQ answering system in theory, the implementation to it is urgently expected to evaluate its performance and make further improvements. Some implicit problems hidden in assumptions could only be revealed in implementing, and then corresponding steps could be explored. However, there are points which we have recognized during the designing of this system are worthy of discussing in the future work, we can indicate them at present.

- To provide a free comfortable interface, we allow users to type in questions without any constrains on structures. While many similar systems have shown that completely unconstrained input may lead numerous problems in the following processing, [37] it’s necessary to think about give some restrictions on user’s input or extending the function of our reformulating assistant to help to reformulate user’s input before matching, so that a consistence on question structure among questions involved in matching is steered.

- We have mentioned that the identification to question type is quite suggestive in question matching. On account of during the searching we have performed NPL which revealed term roles in a sentence, it’s easy to expand the function that including question type identification. Actually another group in the department has researched on it, and we believe the embodying of their tactics to our system in the future is helpful to improve system performance.

- As the knowledge domain providing information to answer questions is focus on some specific topic about products, a number of technical terms are contained, whereas the lexicon, WordNet, we employed mainly covers daily used English words, it causes error in concept connections. So it’s ideally to find a lexicon more suitable to technique domain questions.

Figure 20 An overall flow chart of the system
Input

Tokenizer

File Retrieval

FAQ Files

Retrieve relevant file by K-NN

Term Vector

FAQ File

QA sets in forms of weighted term vector classified by subjects

Output

Natural Language Interface

Reformulated user question is sent to be matched again

QA System

QA set

QA sets in forms of weighted term vector classified by subjects

Selected QA sets with high score

Connect terms conceptly

Calculate semantic similarities between user’s question and QA sets

Works when reformulation failed

Maintain FAQ database

NLP

Marker Passing

Computing

WordNet

POS tagging

Rule Database

Question Reformulation

Human Expert Answering

New QA set

Works when reformulation failed

If failed to retrieve any QA set

Compare statistic similarity by Cosine Measure

Calculate statistic similarity by Cosine Measure

Retrieve relevant file by K-NN

If failed to retrieve any QA set

Selected QA sets with high score

Calculate semantic similarities between user’s question and QA sets

Reformulated user question is sent to be matched again

QA System
Acknowledgement

I would like to acknowledge my supervisor Peter Funk and project director Peter Sävström for their professional directions and supports to my thesis work in the past months, and sincerely appreciate Funk and Ning’s kindly encouragement when I was in trouble during the work. I can’t accomplish this thesis without your help. Thank you very much!

Liu Jianan

2006-11
Appendix

FAQs: frequently-asked answers.

QA system: question-answering system.

IR: information retrieval, the science of searching for information in documents, searching for documents themselves, searching for metadata which describe documents, or searching within databases.

KR: knowledge representation, a research and application domain in artificial intelligence, cognitive science, as well as in the knowledge management & and knowledge engineering.

Natural language understanding: it is referred to as an AI-complete problem, because natural language recognition seems to require extensive knowledge about the outside world and the ability to manipulate it. The definition of "understanding" is one of the major problems in natural language processing.

NLP: natural language processing, a subfield of artificial intelligence and linguistics. It studies the problems of automated generation and understanding of natural human languages.

WordNet: it is a semantic lexicon for the English language.

Hyponym: in linguistics, a hyponym is a word or phrase whose semantic range is included within that of another word.

Antonym: from the Greek anti ("opposite") and onoma ("name") are word pairs that are opposite in meaning.

Meronymy: it is a semantic relation concept used in linguistics. A meronym denotes a constituent part of, or a member of something.

Synonymy: in scientific nomenclature, synonymy refers to the existence of more than one name for one taxon.

Function word: or grammatical words, are words that have little lexical meaning or have ambiguous meaning, but instead serve to express grammatical relationships with other words within a sentence, or specify the attitude or mood of the speaker.

Suffix: in grammar, a suffix — a form of affix — follows the morpheme to which it attaches. It may have a morphemic or a grammatical (inflectional) function.

Euclidean distance: in mathematics, the Euclidean distance or Euclidean metric is the "ordinary" distance between the two points that one would measure with a ruler, which can be proven by repeated application of the Pythagorean theorem.
**ATN**: augmented transition network, a type of graph theoretic structure used in the operational definition of formal languages, used especially in parsing relatively complex natural languages, and having wide application in artificial intelligence.

**Part of speech**: in traditional English grammar, which is patterned after Latin grammar, still taught in schools and used in dictionaries, there are eight parts of speech: noun, verb, adjective, adverb, pronoun, preposition, conjunction, and interjection.
References


[12] 2QALC the Question-Answering system of LIMSI-CNRS


[22] http://cirl.uoregon.edu/research/kr.html
[37] K. Hammond, R. Burke, C. martin, and S. Lytienen, *FAQ Finder: A Case-Based Approach to Knowledge Navigation*, the University of Chicago