2010

An ontology-based concept search model for data repository with limited information

Ding Chen
University of Windsor

AN ONTOLOGY-BASED CONCEPT SEARCH MODEL FOR DATA REPOSITORY WITH LIMITED INFORMATION

by
Ding Chen

A Thesis
Submitted to the Faculty of Graduate Studies
through School of Computer Science
in Partial Fulfillment of the Requirements for
the Degree of Master of Science at the
University of Windsor

Windsor, Ontario, Canada
2010

Your file Votre référence
ISBN: 978-0-494-62732-7
Our file Notre référence
ISBN: 978-0-494-62732-7

Author's Declaration of Originality

I hereby certify that I am the sole author of this thesis and that no part of this thesis has been published or submitted for publication.

I certify that, to the best of my knowledge, my thesis does not infringe upon anyone's copyright nor violate any proprietary rights and that any ideas, techniques, quotations, or any other material from the work of other people included in my thesis, published or otherwise, are fully acknowledged in accordance with the standard referencing practices. Furthermore, to the extent that I have included copyrighted material that surpasses the bounds of fair dealing within the meaning of the Canada Copyright Act, I certify that I have obtained a written permission from the copyright owner(s) to include such material(s) in my thesis and have included copies of such copyright clearances to my appendix.

I declare that this is a true copy of my thesis, including any final revisions, as approved by my thesis committee and the Graduate Studies office, and that this thesis has not been submitted for a higher degree to any other University or Institution.

Abstract

Many of the problems in natural language processing (NLP) or information retrieval (IR) stem from the rich expressive power in natural language. The use of concept search to overcome the limitations of keyword search has been put forward as one of the motivations of the Semantic Web since its emergence in the late 90's. A lot of efforts have been made to adapt concept search principles for improving the performance of information retrieval systems. However most approaches are designed for data repositories which contain a large number of items, each with rich information. We propose a knowledge based concept search method that is particularly designed for the data repository with limited information items by narrowing down a query into one concept. Also, a framework adapting this method is proposed to solve the practical problem about how to extract the information from a specification document into an existing unstructured database. As an important part of the framework, a concept selection method using genetic algorithms and semantic distance is proposed to filter the meaningless or less important information in query generation and matching process. An application development process in mould engineering domain is introduced as a case study to show how to use this framework. The experiment results show our proposed concept search method performs better than classical keyword based search especially for the documents with ambiguous words and the concept selection method has a potential to further improve this concept search method.

To my kind parents for the encouragement and support of a lifetime.

Acknowledgements

I would like to express my gratitude to all those who gave me support during my master's study. I want to especially thank my supervisor, Dr. Ziad Kobti, for the energy and enthusiasm he invested in this research. His guidance has been essential to my graduate study. Moreover, his passion and humor during work makes the journey of my graduate study more enjoyable. I am grateful as well to my thesis committee members, Dr. Yuan and Dr. Gokul, who have been all generous and patient. Their suggestions make this thesis a better work.

Also, I would like to thank Mr. Alan Baljeu, the president of Cornerstone Intelligent Software Cop. His professional experiences in mould design engineering is the priceless help for this work.

This work was supported by a grant from Ontario Centers of Excellence and NSERC Discovery, with matching funds from Cornerstone Intelligent Software Corp.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Many of the problems in information retrieval (IR) stem from the rich expressive power in natural language. Natural language ambiguities including lexical, syntactic and semantic ambiguity [43] have been recognized as an enormous block to information processing in general. Usually, a human has little difficulty to find the intended sense of an ambiguous expression. Naturally, it remains a challenge for machines to perform anywhere near a human's comprehension of natural languages and common concepts. Furthermore, a human may use a number of variations, in words and structure, to describe the same concept, which makes natural languages even more complex for computers to process. According to Bates in [2], the probability of two persons using the term in describing the same thing is less than 20%. It was also found by another study [18] that the probability of two subjects picking the same term for a given entity ranged from 7% to 18%. It is not surprising therefore the ability of traditional simple keyword search method to handle natural language ambiguities to be very limited.

In traditional keyword and Boolean search, the exact words are obtained from the user and a collection of documents which contain the exact search words will be returned. The

limitation of this approach is that the user input must be clear otherwise it leads to the mismatching problem between the user query and documents. Modern information retrieval begins with recognizing the right meaning of a given word in the context, or simply referred to as concept search. The motivation is to enable a computer to "understand" query and information in data repository before it may reason about a given search query. Formally speaking, a concept search is an automated information retrieval approach that is used to search for information that is conceptually similar to the information provided in a search query, which makes the retrieval process more flexible and intelligent.

A lot of efforts have been made to adopt concept search approach to improve the performance of information retrieval. These efforts can be grouped into two types: query expansion and concept indexing depending on whether the concept search works on query or data repository. Query expansion aims to reformulate the original query by further identifying the information. Although this approach is simply called "expansion", the detailed techniques include not only extending but also deleting, revising and re-weighting. While query expansion works in the front end of the information retrieval process, concept indexing works on the documents in the data repository by exploring their hidden meaning. In this process, documents are treated as a set of concepts surrounded by contents and then some related semantic information is added. By doing this, documents are described in a rich semantic expression instead of a simple string. In most information retrieval systems such as the World Wide Web (WWW) or digital libraries, data repositories are databases containing a large number of items. In addition, the items, usually documents, in these data repositories contain rich information. This kind of data repository draws the attention of most researchers. However, there is another scenario that the items in data repository contain only very limited information. The general retrieval approaches do not fit in this case

because of the limited information in the data repository and the difficulty in the ranking procedure. To solve these problems, we propose a new concept search method utilizing semantic distance to narrow down a query into one concept by answering to: "*What is this query talking about?*", instead of adding more terms into the query to answer: "*What is the exact meaning of every word in this query?*".

Although the data repositories with poor information items are pretty rare, they are still possible to cause some practical problems. This research is motivated by a collaborative research project between researches at the University of Windsor and Cornerstone Intelligent Software Corp. The latter is a leading company in mould design automation software development and research. The aim of this project is to build smart tools that can help mould designers to save time and cost in the process of converting mould specification documents to a computer design model. As a part of the project, the concept search method proposed in this thesis is adopted to implement a search engine to help mould designers finding the maximum correct matches extracted from general English specification documents (query) into a proprietary design database (data repository with limited information items). The initial research is being conducted in the domain of mould design and manufacturing specifications, but we expect to be able to adapt this technology to acquire and manage specifications for any other industry.

Storing extracted information from a specification document into data repository involves four major problems. First, to extract the information from a document, the document should be divided into several pieces to meet the requirement that each piece is describing only one concept. Second is to represent knowledge. Because of the limited information in data repository, a corpus independent knowledge base is needed for concept search. Bhogal in 2007 stated that compared to a general purpose knowledge base, domain

specific knowledge base is more suitable for work-task. One proper knowledge representation should be picked up from Logics, Production Rules, Semantic Network and Frames to represent the domain knowledge. Third, narrowing down a query into one concept is the main challenge during the query reformulation process. Unlike most query expansion techniques which keep almost all the original information in query, our narrowing down query expansion approach revises the original query entirely, which leads to a high risk that totally misunderstands the meaning of the query. So the narrowing down process should be considered very carefully. Finally, as queries are generated from a document automatically, the quality of the query is not assured. Meaningless and redundant information in the initial query should be filtered.

In this thesis, we propose a query revision method to deal with the information retrieval problem when the data repository contains limited information items. And also a concept search model adapting our query expansion method is introduced to solve the practical problem that extracts the information from a document into a database. Moreover, we use genetic algorithms to filter the less important information in the initial query.

1.1 Problem Statement

Concept search is a widely accepted way to overcome the shortages of traditional keyword and Boolean in modern information retrieval systems. A lot of efforts have been made to adopt concept search to improve the performance of the information retrieval process. However, while retrieval systems for large data repository draw most of the attentions from researches, there is another scenario that the items in the data repository contain only very limited information and the general concept search methods are not fit for it. Our concept search method is particularly designed to solve the problems in this case.

1.2 Contributions

We propose a query revision method which concludes a query into one concept for concept search. Previous query revision methods are mainly designed for data repository with rich information but do not fit in the database with limited information. Our method is particularly designed to solve this problem. Also, a concept search model adapting this method is proposed to solve the practical problem about how to extract the information from a document into an existing database. An application development process is introduced as a case study to show how to use this model. As an important part of the model, a concept selection method using genetic algorithms and semantic distance is proposed to filter the meaningless or less important information in the query and matching processes. This method is expected to improve the search performance and be applicable to other query expansion techniques.

1.3 Organization of the thesis

The rest of this thesis is organized as follows: Chapter 2 introduces the background knowledge related to this thesis. Then in Chapter 3 the literature review is given. Some milestone research shows two aspects of information retrieval: query expansion and concept indexing. After that the approach we propose and a model adopting our approach is shown in Chapter 4. In Chapter 5, a detailed case study in mould engineering is given to show how to implement our model to solve a practical problem and experiment results are given to compare our concept search method to classical keyword search. Also the effectiveness of our concept selection process using genetic algorithms is tested. Finally, Chapter 6 concludes our contributions, outlines the advantages of our approach and shortfalls for future work.

Chapter 2

Preliminary

In this chapter, the concept of knowledge and knowledge representation methods will be discussed. Also the three issues of information retrieval including concept search, query expansion and concept indexing will be explained. The definition of Genetic Algorithm will be also introduced.

2.1 Knowledge

Knowledge is defined by the Oxford English Dictionary as expertise, skills or understanding of a subject by a person through experience or education. In the Artificial Intelligence (AI) area, the word "Knowledge" means the information that a computer program needs to solve problems in such a way considered intelligent. The theory of knowledge has related with several areas including knowledge acquisition, knowledge representation, knowledge reasoning and knowledge management, etc. Informally, the knowledge representation and reasoning is how an intelligent agent stores its knowledge and how to use its own knowledge to take behaviors. Followed by the knowledge retrieval (KR) problems, the knowledge

engineering and ontological engineering become eye-catching, that most of the researchers intend to propose a KR method that is more generic and formal. With this trend, more intelligent knowledge- based systems sprout and the ontology concept influences the information retrieval area with the growth of knowledge retrieval.

2.1.1 Knowledge Representation

Knowledge representation is the area of Artificial Intelligence concerned with how knowledge can be represented symbolically and manipulated [8]. In 1971, the representation formalisms proposed in DENDRAL system which is highly specific to the domain of Chemistry. Over time, researchers were more and more interested in standardized knowledge representation formalism and ontology that could streamline the process for creating new expert systems[41].

According to [11], the main knowledge representation methods are as followed: Logic, Production Rules, Semantic Network, Frames, etc. Logic is one of the oldest methods for knowledge representation and usually it can be divided into three types including First-Order Logic (FOL), propositional logic and modal logic. FOL, also known as predicate logic, is the most commonly studied and implemented logic [49]. While propositional logic deals with simple declarative propositions, FOL additionally covers predicates and quantification. It is distinguished from propositional logic by its use of quantifiers: universal quantifier $\forall$ and existential quantifier $\exists$. Modal logic is designed for reasoning about different modes of truth which allow us to specify what is necessarily true, known to be true, or believed to be true. Among these, the most important are what "must be" (necessity) and what "may be" (possibility). Modal logic is gradually receiving more attention by the AI community, but research in modal logic for knowledge representation still has a long way to

go [49]. There are two popular implementations based on FOL: Prolog and the Suggested Upper Merged Ontology (SUMO). Prolog is a practical logic programming language based on a subset of FOL. The SUMO, the largest formal ontology publicly available today, is written in Knowledge Interchange Format (KIF) which is based on FOL.

The rules, called production rules or simply 'productions' are another natural representation of knowledge. The logics discussed above are particularly suitable for representing real world models, and the relationships amongst objects and individuals. In this regard production rules do not fit because it lacks expressive and impossible to represent knowledge in complicated domains. However they are ideal for representing procedural knowledge and easily understood by non-experts. The rules are given the following form:

'*If <condition> Then <actions>*'

If the condition is satisfied, then the rule is said to be applicable and after that the actions are executed. One of the main advantages of production rule systems is their modularity[49], which means each production rule defines a piece of independent knowledge, and the operations for adding or deleting rules are independent of other rules.

In 1960, the knowledge frame, or frame, is proposed as a knowledge representation method. A frame can be seen as a data structure containing many slots. Every slot of the frame is assigned detailed knowledge (value) of a particular object or subject. It implements the concept of object oriented programming concept. A frame is like the class of the OOP and has the reusable property. However, it has a disadvantage because it is hard to code in the computer. The syntax of a frame is given as follows:

'*Frame Name <: slotName1 value1><: slotName2 value2> ...*'

Semantic network is another important method to represent knowledge and is considered as an easier method among knowledge representation methods. Semantic network is a

directed graph using node to stand for object and arc for relation to describe the knowledge as shown in Figure 2.1. This method is also considered as a way which is closest to the way of human beings store the knowledge in their brains and is great for taxonomies. An important implementation of semantic network is WordNet which is a lexical database of English language. WordNet contains nouns, adjectives and adverbs grouped into synonyms called synsets. Synsets are linked by semantic and lexical relations to form a network.

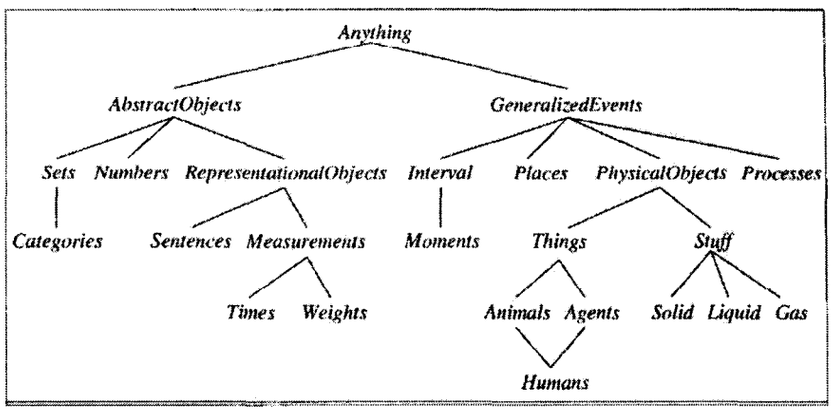

Figure 2.1: Upper knowledge of the world

Description Logics (DL) are a family of knowledge representation languages called description languages [49]. As an extension of semantic networks and frames with formal logical-based semantic, they overcome the limitation in expressiveness of frame systems. Moreover, another advantage of DLs is their close relationship with the Semantic Web community which aims to create a universal medium to represent and share machine-processable data on the Web. Some of the significant technologies of the Semantic Web are: the extensible markup language XML; the resource description framework RDF and the Web ontology language OWL. With the employment by the Semantic Web community, in particular DLs provide the basis for the Web Ontology Language (OWL), DLs are cur-

rently very popular and are actively being researched. With attempts have been made to encode wide bodies of general knowledge, in the 1980s formal computer knowledge representation languages and systems arose. Large scale knowledge representation requires a general-purpose ontology to organize and tie together the various specific domains of knowledge[41]. In order to better present the general concept, ontological engineers have attempted to realize this. An ontology is a description of the concepts and relationships that can exist for an agent or a community of agents. This definition is consistent with the usage of ontology as set of concept definitions, but more general. And it is a different sense of the word than its use in philosophy"[23]. Under the concept of the ontology, the general concept is called upper ontology and the domain-specific ontology is domain ontology which models a specific domain or a part of the world.

2.1.2 Knowledge Engineering

Knowledge engineering (KE) was defined in 1983 by Edward Feigenbaum and Pamela McCorduck [13] as follows: KE is an engineering discipline that involves integrating knowledge into computer systems in order to solve the complex problems normally requiring a high level of human expertise. The goal of the knowledge engineering is to apply the methodological approaches for knowledge-based systems, which turn the process of constructing the knowledge-based systems from an art into an engineering discipline [46].

According to Studer R. et al. [46], the existing overall consensus about the process of building a knowledge based system may be seen as a modeling activity. The knowledge-based system building process aims to realize the ability of the computer problem-solving comparable to a domain expert rather than the cognitive adequate model or inference process. However, in the process of knowledge-based system building, some kinds of knowl-

edge existing subconsciously are always ignored by the experts , while in the expert inference process this kind of knowledge is also important to draw the conclusions and should be built up during the knowledge acquisition phase. From this perspective, the knowledge acquisition process should be viewed as model construction process.

In the process of the knowledge modeling recently, ontology has been considered as the popular approach due to its promise: "a shared and common understanding of a domain that can be communicated between people and heterogeneous and distributed systems. [14] Thus now the main concern of the knowledge engineering is about how to construct ontology. Ontologies can be divided into following five types according to [46] and each of them plays different role in the process of building a domain model[50],[24],[6], [1]:

- Domain ontologies capture the knowledge valid for a particular type of domain (e.g. electronic, medical, mechanic, digital domain).

- Generic ontologies are valid across several domains. For example, an ontology about mereology (part-of relations) is applicable in many technical domains. Generic ontologies are also referred to as super theories [7] or core ontologies [24]

- Application ontologies contain all the necessary knowledge for modelling a particular domain (usually a combination of domain and method ontologies) [19].

- Representational ontologies do not commit to any particular domain. Such ontologies provide representational entities without stating what should be represented. A well-known representational ontology is the Frame Ontology [23], which defines concepts such as frames, slots and slot constraints allowing to express knowledge in an object oriented or frame-based way.

- Method and task ontologies [15], [47]. Task ontologies provide terms specific for

particular tasks (e.g.'hypothesis' belongs to the diagnosis task ontology), and method ontologies provide terms specific to particular PSMs [19].

2.2 Information Retrieval

Information Retrieval is finding material (usually documents) of an unstructured nature (usually text) that satisfies and information need from within large collection (usually store on the computers) [33]. Information retrieval is interdisciplinary which is based on the area of computer science, library science, linguistics, mathematics, etc. In the process of information retrieval, the user issues a query and through the searching process, different degree of relevancy items will be returned by the search engine. Different issues are existing at different steps of the information retrieval. For example, since in most cases the users' queries are in the form of natural language, the problems of natural language that synonym and ambiguity to some extent influence the information retrieval performances. For the data repository, the structure of the content also has impacts on the retrieval results. Therefore various techniques can be applied to improve the performance of the information retrieval. In this section, I will introduce some directions to improve the performance of information retrieval including concept search, query expansion and conceptual indexing.

2.2.1 Concept Search

Keyword and Boolean search is the classical approach. In the process of keyword and Boolean search, the exact words are obtained from the user to look for into the data repository. After submission, a collection of documents which contain the exact search words will be returned. Under this approach, it requires the user to input clear and exact query to

represent what is looked for and only the documents contain the specific words can be returned. Thus, the limitation of this classical approach appears. First the users in some cases are not sure what they are looking for. Moreover, even with the specific terms, the synonym and ambiguity of the natural language always bring with some noises. Synonym means that more than one term can describe a concept. Ambiguity means that one term can express more concepts. For example, if the user is looking for the *Apple* laptop. The query to the search engine can also return lots of documents related to the *fruit apple*. Thus, the classical approach based on full match can drop some documents which are semantically relevant to the query. Besides the problem of synonym and ambiguity, the classical approach also can not deal with the misspelling query scenario and variants on the query terms. In this perspective, the concept search can resolve the problems of the classical approach and through the understanding of the query and its underling concept to improve the performance of the information retrieval.

Formally speaking, a concept search or conceptual search is an automated information retrieval method that is used to search electronically stored unstructured text such as digital archives, email, scientific literature, etc. for information that is conceptually similar to the information provided in a search query. In other words, the ideas expressed in the information retrieved in response to a concept search query are relevant to the ideas contained in the text of the query[55].

2.2.2 Query Expansion

Query expansion is a technique usually used in information retrieval according to [4]. The problem that of natural language synonym and ambiguity influences the efficiency of the information retrieval to some degree. Thus in the process of query expansion, the orig-

inal query will be reformulated to improve the retrieval performance. In the traditional approach, only some simple strategies can be applied. The original query is operated by extracting the nouns and noun phrases, the complete match or best mach will be taken on these extracted key words. However, it's stated by Efthimiadis [12] that the "one term per concept" mere realization might be inadequate to express a concept accurately. Some additional terms should be added to improve the retrieval accuracy or in some cases some terms which are not closely relevant could be removed from the key terms of the query. A substantial improvement can be achieved by using expanded queries which include the techniques of finding synonyms of words, finding all the various morphological forms of words by stemming, correcting spelling errors and automatically searching for the corrected forms or suggesting them in the results and re-weighting the terms in the original query by removing some less important terms. After the reformulation process, according to Vechtomova and Wang [51], the result query expansion terms ideally should have the characteristic of being semantically related to the original query terms and good at discriminating between relevant and non-relevant documents. Also the query expansion operation can take place in both two phases of the search process including the initial query formulation and query reformulation. The following Figure 2.2 shows the different query expansion processes:

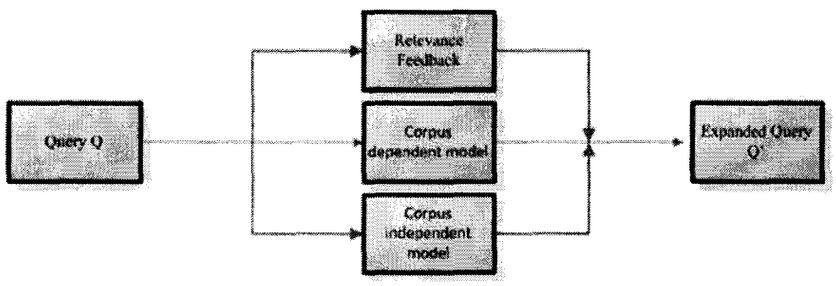

Figure 2.2: Different query expansion processes

It's claimed by Efthimiadis [12] that in the process of query expansion, the source element and the method are the two key elements should be considered. The source element means the source will provide the terms for the expansion. The source can provide the terms for expansion containing the direct search results. Another important type is some forms of knowledge structure. The knowledge structure is independent of the search result and it can either depend on the collection or be independent of it. An algorithms process, characteristics of the collection of documents and an automatically constructed thesaurus can be considered as the collection-dependent knowledge structure. An domain-specific thesauri or a searching thesaurus, global (general-purpose) thesauri, such as Roget's or WordNet and dictionaries can be considered as collection-independent knowledge structure. The following Figure 2.3 shows different elements sources:

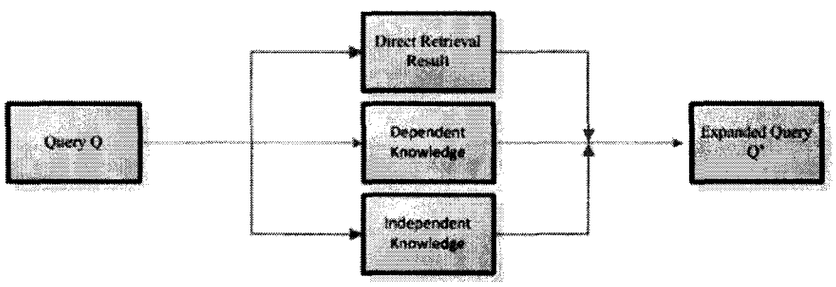

Figure 2.3: Different element sources for QE

2.2.3 Concept Indexing

In the information retrieval, ontology based information retrieval techniques have been used in the past decades. It has also been claimed that ontology based approaches are very promising that it improves the effectiveness of the information retrieval compared with the traditional approaches. In the process of information retrieval, ontology can be used in two

phases. One phase is the front part of the information retrieval, query expansion, that the ontology based technique can help to resolve the problems brought by the natural language including synonym and ambiguity. Ontology has significantly improved the mismatching problems between the user query and the documents. By reformulating the original query, the search results' relevance and accuracy can be improved. Apart from expanding the query with assisted by ontology, the ontology can also be used to index the documents in the database or Web.

Indexing for the corpus can significantly facilitate identifying the relevant knowledge. According to Voss [54], in a concept indexing, any object, conceptualization, idea, person, etc. are described as concepts. A concept indexing maintains the collection of documents and a set of interrelated concepts. Concepts are cross-referenced with explicitly marked pieces of text and with implicitly detected similar pieces of text. With the concept indexing, it can easily describe the content of the document collection, represent the nuggets of the document collection, explore the hidden connections among the documents and discover the relevant documents. Khan and Luo in [30] identified the problem of keyword based traditional indexing approach. The documents returned to the users are the documents which contain the keywords. In this process lots of documents which contain the desired semantic information without the specific keywords are ignored. Thus they proposed to use ontology as the index structure to resolve the above mentioned problem. They also stated the main issue that how to identify the concepts to describe the documents and index them.

2.3 Genetic Algorithms

Genetic algorithms (GA) are used in optimization problem to evolve towards better solutions from a set of abstract representations of candidate solutions. From [20] and [34],

GA was born 30 years ago, originated with an idea of applying the biological principle of evolution to artificial intelligent systems. Genetic algorithms are loosely based on ideas from population genetics and work via the process of natural selection. First, a population of individuals is randomly created as the sample set of potential solutions which then evolves towards a set of more optimal solutions for some problems of interest. Then in the next time step, some poor individuals in the population are tend to die out while good ones mate and propagate their advantageous traits to introduce more potential individuals into the sample set. After this selection process, a new population is created by making copies of more successful individuals and deleting less successful ones. When iterating the cycle of evaluation, selection and genetic operations for many generations, the overall fitness of the generation is generally improved and the individuals in the population are more likely to be the better solutions.

Some terms in the algorithms are defined as in [48]. The *population* is the collection of candidate solutions that we are considering during the algorithms and a single solution in the population is referred to as an *individual.* The *fitness* of an individual represents how good the solution represented by the individual is. Higher fitness means the better the solution. The selection process is analogous to the survival of the fittest in the natural world, which means the individuals with higher fitness have more chances to live and propagate while the poor individuals are tend to die out. There are three kinds of basic genetic operations to make the evolution process (e.g., the generation of a new population). *Reproduction* is used to simply make a copy of the potential individual. During the copy duration, there is a probability of *mutation* and *crossover*. Mutation is the act of randomly altering the value of a gene in an individual (e.g., random bit flips if the genome is represented by a bit string). Crossover is the act of exchanging some gene values between two potential individuals to

produce one or more new individuals (e.g., exchange of corresponding substrings of two bit strings). By transforming the previous set of good individuals to a new generation using reproduction, the mutation and crossover operations generate some new individuals which ideally have a high chance of also being good or even better.

In general, genetic algorithms are very useful particularly for the problems where it is extremely difficult or even impossible to get an exact solution or for difficult problems where the exact solution is not necessarily required. They tend to work better than traditional optimization algorithms because they take advantage of an entire set of solutions spread throughout the solution space to avoid local optima caused by the usage of single-point transaction rules to move from one single instance to another.

Chapter 3

Related Work

Information retrieval contains two important issues including query expansion and indexing. Regarding on these two issues, different approaches to query expansion will be surveyed in the first section from the traditional methods to the current ontology based methods. In the latter part of this chapter, some important approaches to concept indexing will also be discussed.

3.1 Difference Approaches to Query Expansion

In the past research work, different approaches have been proposed to guide query expansion. Efthimiadis in 1996 has divided different approaches into three types which are manual query expansion, automatic query expansion and interactive query expansion. Under the automatic query expansion approach, it can be further divided into two types which are search results based approach and knowledge structure based approach. Based on the source whether collection is dependent or independent, the knowledge structure based approach can also be categorized into two types. Bhogal J. in [5] also identified that query

expansion can be manual, automatic or user-assisted regarding on the process of formulation of the original queries. Manual query expansion process and user-assisted query expansion both need the involvement of the user. In manual query expansion approaches, users select the terms to be added into the new query based on the their experiences. In user-assisted query expansion, the system generates possible query expansion terms and is assisted by the users to select which to include.

3.1.1 Traditional Relevance Feedback Techniques

Salton and Mcgill in [42] claimed that apart from the manual/intellectual query reformulation where the task falls to the searcher, it is possible for the system to take over this task entirely by requiring only some "yes-no" answer from the user. This automatic query reformulation process is called relevance feedback. Rocchino [40] proposed the relevance feedback approach for query modification. He stated that relevance feedback is motivated by the fact that it is easy for users to judge some documents as relevant or non-relevant for their own query. A system can automatically generate a better query with the addition of related new terms to the original query by relevance judgments. In this process, the user is required to judge the relevance of the top several documents retrieved by the system. According to these judgments, the system issues the new query by formulating the initial query to find more relevant documents from the collection. It's claimed that relevance feedback has been shown to work quite effectively across test collections.

The typical automatic relevance feedback operation involves an initial search with a user-supplied query and an initial retrieval of certain documents. Then, from a display (usually titles or abstracts of the retrieved documents) the user identifies and chooses some relevant documents. Those documents are used to modify the query by the following three

methods: reweighing the existing query terms, adding terms that appear useful and deleting terms that do not helpful for the future searching. This process creates a new query which resembles the relevant documents more than the original query does. Based on the relevance feedback query expansion approach, another variant of the relevance feedback is the pseudo-feedback approach. It's stated by Singhal A. [45] that the top few documents returned by an information retrieval system are often on the general query topic, selecting related terms from these documents should yield useful new terms irrespective of document relevance. In pseudo-feedback the information retrieval system assumes that the top few documents retrieved for the initial user query are "relevant", and does relevance feedback to generate a new query. This expanded new query is then used to rank documents for presentation to the user. Pseudo feedback has been shown to be a very effective technique, especially for short user queries. Regarding the relevance feedback approach or the variant pseudo-feedback approach, they both contain the problem of weighting new terms. This kind of weighting tries to answer whether all terms in the expanded query should have equal weights or whether the new terms should have a higher or lower weights.

3.1.2 Corpus Dependent Model Query Expansion Approach

Besides the traditional relevance feedback approach, many researchers have begun to utilize the context information from the document collection. The most frequently used techniques are stemming, clustering and term co-occurrence, etc. Especially, during the late 60s and early 70s, lots of researchers have driven their attentions to the term clustering based method.

Sparck-Jones [28] is the researcher who has conducted the extensive work of query expansion using term clustering. Based on this method, similar documents are clustered

and groups of similar terms are identified before the user issues the query. With adding clusters of related terms, the query is expanded. Jones conducted the experiments and claimed that the retrieval performance is improved by using an automatically obtained term classification. However, in Jones' subsequent research work, it's claimed that the retrieval performance with term clustering approach can only be significantly improved on small collection. The reason for the poor results obtained from the term clustering should result from the properties of the collection that are used to be clustered. Sparck-Jones [29] conducted the detail analysis on the experiment done by Lesk and drew the conclusion that the fault was caused by the insufficient differences in the vocabulary between relevant and non-relevant documents. Later in 1969 and 1990, two branches of the researchers identified the fault of the term clustering approach. It's stated by Lesk [32] that only terms whose meaning is purely local were captured by the cluster and the cluster didn't reflect their general meaning in technical text. And Deerwester et al. [10] addressed that representing document clusters and term clusters together at the same time is difficult.

Term co-occurrence assumes independency of the two terms and it refers to the two terms alongside each other in a certain order in a text corpus. Rijsbergen et al. are the first researchers to conduct research work on systematically evaluate the performances of query expansion using term co-occurrence. They utilized the maximum spanning tree as the source to expand the query. While, it's also identified that the high frequency terms do not discriminate between relevant and non-relevant documents. Also in 2001, Ruthven I. identified the problem that high frequency terms approaches are very influenced by the collection and their relevance to the query.

3.1.3 Corpus Independent Model Query Expansion Approach

Bhogal J. [5] has identified that the problem of traditional relevance feedback techniques and corpus dependent query expansion is that they are content driven. The candidate terms for query expansion are extracted by analyzing corpus content. In the case that the relevant documents are not sufficient and the reasonable set of terms that represent the subject area for the query are not contained, the traditional relevance feedback techniques and corpus dependent query expansion are not working. The collection independent knowledge based approaches don't suffer this drawback. So far, the most popular form of the collection independent knowledge is considered as the Ontology and research work have been conducted on the ontology related areas of information retrieval including the thematic summarization, word sense disambiguation, indexing, text classification, query formulation, cross lingual information retrieval and concept mapping. For collection independent query expansion, the most advantage is that the terms suggested to modify the original query are through the analysis of the general knowledge base rather than other sources which depend on the limited relevant collections. Ruthven I. [39] also identified that collection independent knowledge bases are more precise than traditional approaches in text disambiguation within a specific domain and they are very appropriate approaches for the short query scenario. Since the knowledge bases have been divided into two types including general knowledge and domain specific knowledge. Thus various approaches have been proposed to formulate the original query taking advantage of general knowledge, domain knowledge or both.

Voorhees [52] proposed an automatic indexing method to detect and resolve the sense of the ambiguous nouns in texts or the queries. Attempts have been done on indexing with word senses to exploit the semantics in WordNet to improve the retrieval performances. In this method, the nouns in the text and "is-a" relations identified by the WordNet are used to

select a sense for each ambiguous noun in the text. The author stated that the result of the indexing procedure is a vector. In this vector, some terms represent the word sense rather than word stems. The experiments were conducted and the result obtained by Voorhees showed that sense-based vector is only more effective for some queries and stem-based vector is more effective overall. Through the analysis, Voorhees drew the conclusion that the difficulty in disambiguating senses in short query degrades the sense-based vector approach. By only utilizing the simple "is-a" relation is not sufficient to select the correct sense of a noun from the WordNet and incorrect matches degrades the retrieval performance. Based on his previous work, Voorhees [53] examined the utility of lexical query expansion in TREC collection. By expanding the concepts using the typed link included in WordNet, the retrieval effectiveness is different. When the query completely describes the information to be sought, the retrieval result is a little different. When the query is not a complete description of the information to be sought, the result can be largely improved. However, in this query expansion process, the expansion concepts are chosen by hand rather than automatic generated.

Gonzalo et al. [22] identified the problems existing in the previous research work that only few successful results, query expansion process is manually and bad automatic expansion process, etc. Gonzalo also identified the WordNet's potential advantage for retrieval task: it offers the possibility to discriminate word senses in documents and queries as well as provides the chance of matching semantically related words. They proposed the retrieval strategy to adapt classical vector model based system using WordNet synsets as indexing space. It's claimed by the author that the terms are fully disambiguated and equivalent terms can be identified by using this approach. The author conducted the experiments and obtained the results that synsets indexing is very helpful for text retrieval.

Huang [26] discussed the web information retrieval techniques in search engines and hierarchical categories. Huang defined the web ontology as the hierarchical directories. Hierarchical directories are portals to the web and provide good starting point for the document searching to improve the retrieval performances. By constraining the user to issue the query from the particular portal, retrieval performance can be improved.

Finkelstein et al. [16] identified that several skills can be applied to the basic key-word based retrieval system. In their work, they proposed a new conceptual paradigm for performing search in context. It's stated that this conceptual paradigm can automate the search process and provide the users with more relevant results. Under this approach, search is initiated from a text query marked by a user in a document and is guided by the text surrounding the marked query in that document. It's claimed by the author that the context-driven information retrieval process involves semantic key word extraction and clustering to automatically generate new queries. The new context-based approach improves the relevance of search result by applying natural language processing techniques to the captured context in order to guide the search. Different from the precious existing methods which analyze the entire document or require users' feedback, this approach automatically analyzes the context in the immediate vicinity of the focus text without running over the more distant topics in the source document. These processes capture more background information and was claimed by the author obtain the satisfied results.

In 2003, Navigli and Velardi [35] identified that sense based query expansion never proved its effectiveness and proposed to use ontology to expand query. The author stated that the expansion with only synonyms or heteronyms is not sufficient and other types of semantic information derived from the ontology are much more effective at improving search results. They proposed a word sense disambiguation method based on structure pattern

recognition. Under their approach, they use ontological information to extract the semantic domain of a word instead of using taxonomic relation for sense based expansion. The results drawn from their experiments showed that the words in the same semantic domain of the query words appear as the best candidates for expansion and the type of sense-related information are more useful for the information retrieval. In the same year, Hersh et al. [25] proposed another query expansion method using external knowledge resources. Hersh et al. identified that in the domain of bioinformatics community, there are lots of available databases that can provide rich information. Aggregation of data can be the initial source of information to augment the queries. Results obtained from the experiments are negative and through the analysis, the author drew the conclusion that the use of external information to expand query need to be selective or filtered. More positive results can be obtained by expanding query using external knowledge resources within specific task.

Fu et al. [17] identified the importance of Ontology in information retrieval. In their work, they carried some researches on how ontology can be used to support retrieval of documents which are considered spatially relevant to the users' query. They proposed a new query expansion technique based on both domain ontology and geographical ontology instead of domain independent ontology like WordNet. Also it's claimed by Fu G. that their approach is different from the traditional approach which expands the query by derivation of its geographical query footprint. Their proposed approach is specifically focus on handling the queries which contain spatial terms and fuzzy spatial relationships. Fu took different factors into account to expand the query including encoding spatial terms in the geographical ontology and non-spatial terms in the domain knowledge. Moreover, they also considered semantics of the spatial relationships and the context. The experiments conducted by the author showed that the search results are considerably improved with

the expanded query using proposed approach. It's also claimed by them that the proposed methods work efficiently using realistic ontology in a distributed spatial search environment. At the same time another group of researchers, Nilsson et al. [37], also proposed a technique for query expansion using domain ontology. Nilsson et al. stated the problem of using general-purpose resources such as WordNet results in the little or negative impact on retrieval results. Expanding query with synonyms, hyponyms and hyponyms usually leads too broad interpretation therefore degrades the information retrieval performances. They identified that domain-specific ontology works well for the domain they created and expanding user's query by conservative form or more specific terms can effectively resolve the problem of query expansion using general ontology. The prototype retrieval system they developed using domain-specific ontology for query expansion and translation along with modules to recognize named entities and temporal expression has demonstrated the effectiveness of their approach.

Xu et al. [57] in their work presented the design and evaluation of biomedical literature searching approaches. They aimed to expand queries by three widely used strategies including local analysis, global analysis and ontology-based term re-weighting. For the ontology based re-weighting strategies, the weight of the original query are assigned equally. If an original term has higher term frequency in the initial query with specific major UMLS semantic type, it will be assigned higher weight in the expanded query. Moreover, if a key query term or an expanded term corresponds a major UMLS semantic type, its preferred MeSH term synonym is expanded and owns higher weight. Xu X. conducted the experiments and claimed that ontology-based approach provides the best result and if combined with other strategies, other approaches result will also be improved.

In 2007, Nguyen [36] stated that previous researchers' work on query expansion based

on synonym and similarity approaches. They proposed an approach of ontology development and mechanism of similar noun phrase expansion based on ontology of key property, key member of an object.

Calegari and Pasi [9] proposed the work to perform personalized search. In their work, user profiles are recorded from their browsing patterns to catch more relevant documents. In order to understand the meaning of the search query, a local fuzzy ontology is used to keep trace of the past experiences of the user and expand the original query with the terms that are most frequently inserted in the same query. Page ontology is constructed automatic. The author claimed that personalization used ontology produces better results compared with traditional approaches.

Based on the previous proposed work, Bhogal identified that Domain specific ontologies are more suitable for work-tasks. The terminology in these ontologies is less ambiguous therefore short queries can be expanded with a higher chance of accuracy. General ontologies would be suitable for information type broad queries however the query expansion process may need some guidance or interaction from the user. Although the ontology based approach has achieved huge improvements in retrieval process, it still has some drawbacks of its own. Ruthven I. [39] stated the disadvantage of the ontology based approach is that ontologies are typically difficult to build up and maintained in order to guarantee the necessary level of precision to avoid the decrease of performance.

3.2 Difference Approaches to Conceptual indexing

According to [56], conceptual indexing combines techniques from knowledge representation and natural language processing with classical techniques for indexing words and phrases in text to enable a retrieval system to make connections between the terminology

of user request and related terminology in the information repository. Woods claimed that the conceptual indexing technique resolves the problem of traditional approaches. This technique automatically analyzes the conceptual structure of phrases extracted from the collection and establishes connections between the query and corpus based on the semantic relationships. It also automatically organizes all the words and phrases of the corpus into a conceptual taxonomy. They presented a methodology which organizes conceptual description into a conceptual taxonomy and the index can be used by the retrieval algorithm. It's also claimed by the author that conceptual taxonomy escapes from the limitation of traditional library classification hierarchies by allowing conceptual categories to own more than one parent categories. Moreover, it allows automatic location of the most specific concepts even if the query is not exact. The results obtained from the experiment have demonstrated the effectiveness of this approach which substantially improves the people's ability to find specific information they need.

Later in 1999, Voss et al. identified that marking text in a document is a convenient way to identify bits of knowledge. Based on the marking text, the documents can be categorized for searching convenience. The author also stated that under concept indexing, any object, conceptualization, idea, etc, can be a concept. Within their work, a collection of documents and a set of interrelated concepts are maintained. In their work, Voss et al. summarized that the importance of the concept indexing: the concept indexing satisfies a range of information and knowledge needs which means that the new concepts or relations can be easily introduced. Concept indexing can help to find the hidden connections between the documents and also the number of concept occurrences can indicate the relevance of retrieved documents.

In Khan and Luo's work [30], the ontology used to index the documents is built up au-

tomatically in bottom up fashion. With the clustering algorithms, the first hierarchy is constructed. When documents are on the similar topic, they are assigned to the same concept in ontology. Then the concept recognized is assigned to the node of hierarchy structure. The modified self-organizing tree algorithm is applied to construct the ontology automatically. Different concepts are defined by sub-tree region of ontology. The concepts appearing in different regions are mutually exclusive. Only the region containing a large collection of documents is kept and the concepts in other regions are removed. Within a certain region, the concepts are adjusted based on the semantic distance factors. In this case, the concept which correlates with the higher number of other concepts is kept. The main contributions claimed by khan and Luo are as follows: the modified self-organizing tree algorithm is the new mechanism which can be used to generate ontology automatically to make this new approach scalable. Second, a new automatic concept selection algorithm is proposed to find an appropriate concept for each node in hierarchy.

Kiryakov et. al. [31] presented their view about a holistic system allowing annotation, indexing and retrieval of documents with respect to real-world entities. In the classical IR, the documents are characterized by bags of words and for the last decade lots of considerable efforts have been towards using word-senses. They carried out the work to modify the classical information retrieval method to retrieve the documents on relevance with name entities rather than words. They considered that for semantic annotation, it should be based on specific knowledge about the world rather than general knowledge. Also a upper-level ontology was introduced to assure the efficiency and reusability of the metadata and an extensive knowledge base of entities is maintained based on ontology. In order to demonstrate the vision, KIM platform was developed and the effectiveness of entity-aware IR technique was demonstrated.

Baziz [3] also identified the importance of applied semantics in information retrieval and addressed the problem of representing documents semantics and taking advantage of this in information retrieval process. He proposed another approach for conceptual indexing using ontology. They represented the content of the document by the best semantic network named document semantic core. There are two main steps: the first one is extracting the words and phrases from a document which is guided by the external general-purpose ontology WordNet. Then a global disambiguation of the extracted concepts based on the document lead to construct the semantic network. After above two steps, the concepts represent the nodes of the semantic network whereas similarity measure values between connected nodes weight the links. The resulting cored concepts are used for the document conceptual indexing in information retrieval.

Setchi and Tang [44] addressed the problem that the majority of ontology based indexing are at document level. With the advances in ontological engineering, the document indexing can be at word level. Motivated by this consideration, they proposed a new concept indexing algorithm. This algorithm applies a general ontology named OntoRo and an ontologically tagged corpus named OntoCorp. OntoRo and OntoCorp are used in a two-stage supervised machine learning process which can construct ontology tagging rules. This approach adds ontology information to words and allows full text to be searched, browsed and analyzed at different levels of abstraction. They conducted the experiments and claimed that the results they obtained are encouraging.

Chapter 4

Proposed Method

In this chapter, the problem in our research work is addressed and the analysis of the previous methods is given. Our efforts have been on the new query revision method using semantic distance. Details of our proposed method is introduced. Moreover, a concept search model using our proposed method is given.

4.1 Proposed Query Revision Method Using Semantic Distance

In most information retrieval systems, data repositories are databases containing a large number of items, such as World Wide Web (WWW) or digital libraries. The items in these data repositories are usually documents which contain rich information. This kind of data repository is common and draws a lot of attentions from researchers. To improve the performance of the retrieval systems, a lot of efforts, as introduced in Chapter 3, have been made to adapt concept search principle to overcome the shortages of traditional keyword

and Boolean search.

However, there is another scenario that the items in data repository contain only very limited information. These general retrieval approaches designed for rich information documents do not fit in this case for two reasons. First, concept indexing approaches and corpus dependent model query expansion approaches rely on rich information to analyze query and data repository. Conceptual indexing aims to index or summarize documents in data repository by identifying the concepts. Corpus dependent model query expansion approaches need a data repository with rich information to find similar or related terms such as by clustering the similar documents or term co-occurrence to generate candidate terms for query expansion. Thus for date repository with limited information, these two concept search branches cannot help. Second, although corpus independent model query expansion approaches can be adopted in this problem, previous methods mainly work by explaining or further identifying the meaning of terms in query such as in [52], [22], [16],[35],[57]and [36]. These methods perform well by matching the terms (including newly added terms) in query with the terms in data repository and then the retrieved items are ranked mainly based on the quantity of the matches. The items with more matches are ranked higher in the result list. However, for items with limited information, although the right results to query can still be retrieved by adding some meaningful terms, the ranking task is hard to accomplish due to the fact that the amounts of matches are all similar. In worst case, if each item contains only one term, then each term in query may find one corresponding item and the amount of matches for each retrieved item is always one, which means the ranking process cannot take effect at all. To avoid these problems caused by limited information data repository, we propose a new query revision method utilizing semantic distance to narrow down a query into one concept by answering to:"*What is this query talking about?*", instead

of adding more terms into the query to answer: "*What is the exact meaning of every word in this query?*". Although narrowing down a query into one specific concept is risky because the query is totally revised before the search phrase, it is still a possible way to solve the information retrieval problem for the date repository with limited information.

The main challenge of our proposed revision method is to find a most matching concept for a given query. One possible way to achieve this is by analyzing the syntax structure of the query sentence to understand its meaning. However, this is a very complicated work due to the variety and ambiguity in natural language. For example, "*The man saw the boy with the telescope*": is the *man* or the *boy* who has a telescope? It is even a confusing sentence for human beings and the computers will make it even worse. Thus, instead of analyzing syntax, we adopt the semantic distance to solve this problem. Our proposed approach relies on knowledge base in which concepts are linked by relationships as a network. In the network, the semantic distance represents how closely two concepts are related. If two concepts cannot be connected directly, then more "hop" they need to reach the other means less related they are. The same idea is also used in [21] for keyword proximity search. In our query narrowing down problem, we match the terms in query to the concepts in knowledge base and find the "central" concept of them. This "central" concept may not be the closest concept to any single term in query but it is supposed to be the best candidate concept to represent the query. A sample of this approach is illustrated in the following figure 4.1 .

However, using one concept to represent a query may be too risky so more candidate concepts around the "central" concept are also added into candidate list and ranked by their semantic distances to the "central" concept. After this process, the original query is revised to a new query which contains a list of candidate concepts. Then the concepts in candidate

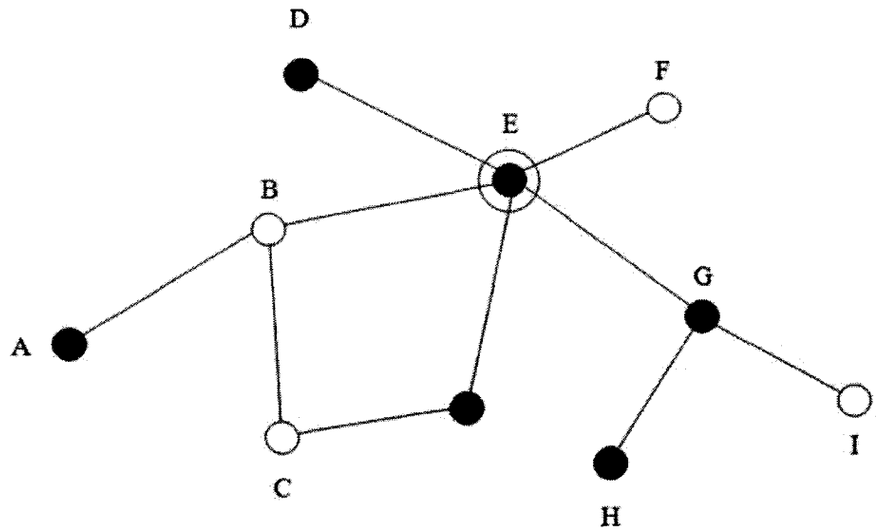

Figure 4.1: Sample of semantic distance

list are searched in data repository one by one and retrieved results are ranked based on the candidate concept's rank.

4.2 Proposed Concept Search Model

This model is proposed to solve the problem that storing extracted information from a document into database. We evolve the search from a heuristic using traditional keyword-based search method and then move towards a more semantic contextual search supported by a domain specific ontology. In our model, knowledge base is added for query revision. By using the information in knowledge base, the query is revised to a list of ranked candidate concepts and then the revised query is searched in data repository to generate the final search results. The overall architecture of our proposed model is illustrated in Figure 4.2.

Based on different functions, the model is divided into three main components including: the Query Generation, the Knowledge Base Creation and the Query Expansion and Search Engine. Details of different components will be discussed in the following sub sections.

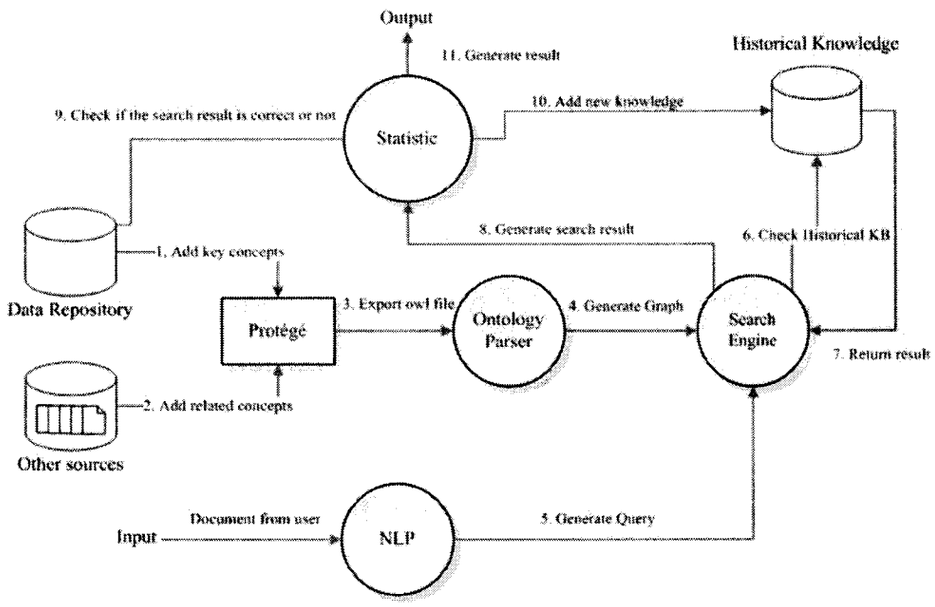

Figure 4.2: Overall architecture of ontology-based concept search model

4.2.1 Query Generation

Most of researches treat documents as data repositories such as in [28],[16] and [17], but none of them takes documents as the query source. In our model, we propose a method to take a document as query source to extract the information into data repository. The overall process to generate queries from a document is illustrated in Figure 4.3. English is the only language supported in our model at this time.

One main challenge for taking a document as query source is to recognize the key

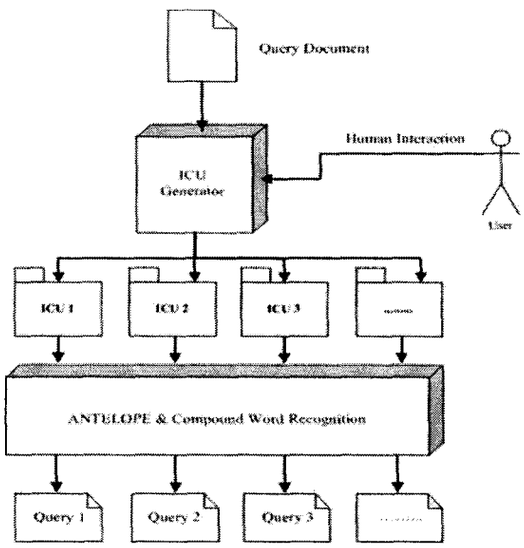

Figure 4.3: Process of query generation

concepts in the document. It is hard to tell how many key concepts exist in one document and which sentences are talking about the same concept. All sentences in a document may describe the same concept or they may describe different ones, or various aspects of the same concept. Separating the sentences about the same concept as individual queries also means separating the features of the concept and losing information, which weakens the recognition ability.

One approach to solve this problem is by calculating the word frequency in the document and assuming the words with higher frequency are more important in this document. This method is a possible way to recognize key words, but not key concepts because one concept may be expressed by different words. For example, "*plane*" and "*aircraft*" are synonyms and can be used to describe the same concept: "*flying machine*". Thus using this method to recognize key concepts is not reliable. On the other hand, key concepts recognition is very important to our model because we view the document as the query source

and the poor key concept recognition results in missing information. We consider grouping more sentences that talk about the same concept, thereby more information about this concept can be extracted and the concept recognition results can be more accurate.

To deal with this problem, we coin the Individual Concept Unit (ICU) which contains one or more of the sentences about the same concept. For the reason that fully automation of sentence understanding is still a challenge, to generate ICUs from a document accurately, user supervision is needed after the initial ICUs are generated from document automatically. First, the document is imported into the system and then divided into individual sentences as initial ICUs automatically. After that, user supervises and refines these initial ICUs by utilizing operations of combining, adding, deleting or splitting.

After the document is divided into several ICUs, each ICU is treated as an individual original query Q. For further query revision, the keywords (usually nouns) are extracted from the original query to generate a new query Q' and meaningless words such as article or determiner are abandoned. It should be mentioned that one important step during nouns extraction is compound word recognition. The use of compound word is a common phenomenon and important characteristic in English. A compound word might represent a new concept from its atom words [27]. For example, word phrase "*log cabin*" can be divided into two words: "*log*" and "*cabin*" or it can also be treated as one compound word "*log cabin*". However, the phrase "*log cabin*" is talking about a kind of cabin, not log. In the former case, the concepts about "*log*" are also found results, which may weaken the concept matching accuracy. Meanwhile, some stemming techniques such as transforming a plural form noun into its base form can be adopted during this process.

4.2.2 Knowledge Base Creation

WordNet, a lexical database of English language, is treated as the knowledge base in some information retrieval systems such as [22] and [52]. However, in our view WordNet is a general knowledge base but its ability to understand and provide detailed information for terms in a specific domain is very limited. This viewpoint is also stated in [5]. Although WordNet can be used to provide some general information of a word (e.g. part of speech), a domain knowledge base is also needed to further identify a word in a specific domain.

Our domain knowledge base is designed to contain concepts and relationships. As described in section 2.1, there are several ways to represent knowledge including Logics, Production Rules, Semantic Network and Frames. However, no single knowledge representation is likely to be optimal for all types of systems or domains. In our search model, the semantic distance is the main attribute to determine the relationship between concepts so the reasoning power of different representations is not the primary concern. In regard to the representation ability, description logics, or more specific, web ontology language (OWL) is adapted as our knowledge representation language the following reasons. First, production rule system is ideal for representing procedural knowledge but not suitable for representing real world models and the relationships among the objects. Second, description logic is an extension of semantic network and frame, which have exceeded both of them in regards to knowledge representation power. At the same time, OWL are more easily understood and utilized by domain experts who are lack of computer technologies with the help of some user friendly tools such as Protege. Although logic is particularity suitable for representing the concepts and the relationships in real world, the most popular editor tool for logics is Prolog which is hard to be utilized by non-expert users. Moreover, OWL has been a standard to represent ontology for its great reusability and wide spread

thus it is very popular and actively researched in information retrieval field. Ontology is a formal specification of a shared conceptualization and an useful ontology should be both useable and reusable. The way to manually create a formal and well organized ontology is explained in [6] and a sample process on creating the ontology for a specific domain is described in a case study in Chapter 5. The shortage of ontology is that it's hard to be used to represent the information for a large domain. Moreover, until now there is still no widely accepted standard for ontology design and analysis. However, for a limited specific domain, ontology is still a good choice to be a suitable knowledge base.

4.2.3 Query Revision and Search Engine

The query revision process is composed of two steps: matching and ranking. Concepts and their relationships parsed from knowledge base and the query generated from documents are used in this process. For each keyword in a query, the matching concepts in knowledge base are found if exist. An important point need to be mentioned here is the matching strategy. Two different matching strategies can be adopted here: exact match or partial match. Simply speaking, for two strings, exact match means the two strings should be exactly the same (i.e. "*program*" does not match "*programmer*") while partial match means one string can be the substring of another (i.e. "*program*" matches "*programmer*"). Partial match has more chances to find related concepts but it also may lead to some mismatching faults. These two approaches are both adoptable in our model so they are both tried in the experiments. After finding the matching concepts, the query revision method described in section 4.1 is adopted to generate a new query which contains a ranked candidate concept list. Because the concept search has been done during the query revision process, the search engine is not important in our model so the traditional word matching search is enough.

4.2.4 Concept Selection

As in our model, the queries are generated from a document automatically so the quality of the query is not assured. Unlike the queries given by user input, automatically generated queries are more possible to have some meaningless or less important information. To solve this problem, concept selection techniques can be adapted to further revise a query Q into a better Q' by deleting the less important information if exists. Also, for one keyword in the query there might be more than one corresponding concepts in the knowledge base. For example, if we want to find the corresponding concepts for the word "*leader*", the concept named "*leader pin*" is also retrieved if partial match strategy is used. In most cases, "*leader pin*" is an unrelated concept with "*leader*" so that this leads to some mismatching problems. Concept selection can also help to filter this kind of mismatching faults.

Concept selection techniques can also be treated as a kind of query revision. While most query revision systems focus on how to add more related information into the query to enhance the search, concept selection focuses on how to filter redundant or less important information from the query. We propose a concept selection approach based on semantic distance. The basic idea of our approach is shown in Figure 4.4. which is a graphically represented knowledge base and the black nodes are the matched concepts to the keywords in the query. In this case, concept L is far away from other selected concepts, which means its relationships to other selected concepts are weak. What we want to achieve by adapting the selection process is to find this kind of weak related concepts in the query and take them out.

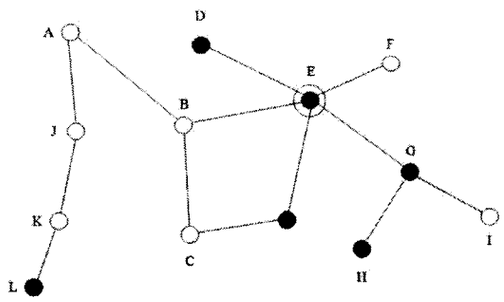

Figure 4.4: Concept selection sample

Chapter 5

Case Study

In this chapter, we present a detailed implementation process for creating an industrial application based on our proposed model as a case study. Also, some experiments are conducted to demonstrate the use of the concept search method and the benefits of concept selection method as adapted during this process.

5.1 Background

This work is motivated by a collaborative research project between researchers at the University of Windsor and Cornerstone Intelligent Software Corp. The latter is a leading company in mould design automation software development and research. During the mould design process, an extensive manual process that starts with libraries of standards of user specific documentations describing various part details to be generated by the mould being designed is involved. It is then up to the mould designers with extensive industrial experience in the domain to comb through the specification documents and translate the client requirements into meaningful design for a mould that matches the desired specifications.

It is typical for domain experts to spend a significant amount of time in manual labor going from specification to a digital design model ready for manufacturing in tool and die shops. To make it worse, it is not uncommon for clients to require a change in the design configuration which adds time delays and excessive costs to the overall project, let alone the design reconfiguration complexities. It is imperative that any change to the design, or a change triggered by the initial change ripple, must conform at all times to the industrial specifications and standards.

The aim of this project is to build smart tools that can help mould designers to save time and cost in the process of converting mould specification to a computer design model. This section reports the initial phase of this project in presenting the tool and underlying algorithms for knowledge extraction. The aim is to help mould designers to find the maximum correct matches extracted from general English specification documents into a proprietary design database. The initial research is being conducted in the domain of mould design and manufacturing specifications, but we expect to be able to adapt this technology to acquire and manage specifications for any other industry.

5.2 Implementation

5.2.1 Query Generation from Mould Design Specification

The keyword generation process aims to take a mould design specification document as input and divide it into several queries containing a list of keywords. This process is made up of four steps: dividing a text document into initial Individual Concept Unit (ICU), extracting keywords from ICU, stemming keywords and recognizing compound words. Each ICU is considered as an individual query in the search phrase. In the first step, a text document

is divided into several initial ICUs automatically based on two possible strategies.

- Sentence Based Splitting: In this approach, each sentence in the text document is considered as an initial ICU. At first glance, "sentence" is a set of words ending with a dot. However, for a filename with extension (paper.pdf) or an IP address (127.0.0.1), this rule is no longer accurate. In a strictly formatted document, we should say that either a final dot is the last character of the input sentence, or it is followed by a space. Nevertheless, this improved rule still fails on name such as 'John F. Kennedy'. Therefore, splitting a document into sentences is a not-so-trivial task for machine [43]. ANTELOP, which stands for Advanced Natural Language Object-oriented Processing Environment, is a natural language processing framework that can provide a sentence splitting function to deal with this problem.

- Index Based Splitting: In some kinds of documents, the closely related information is already grouped and indexed such as *list* or *table* format. In these kinds of formats, the structure of document also implies some relationships between sentences. More specifically, the sentences under the same index are usually describing the same concept as shown in Figure 5.1. For this kind of document format, splitting the document based on index rather than sentence can keep the original sentence relationships in document to initialize ICUs more precisely.

In our case, the mould design specifications are in the *list* format as shown in Figure 5.1. So the index based splitting strategy is adopted as the approach to divide a document into initial ICUs. After that, a domain expert supervised revision work is needed to further improve the quality of generated ICUs. A small tool is provided to help this process. By using this tool, domain experts can review all the initialized ICUs and refine them by simple

1. *Is automatic accelerated ejection required?*
2. *Do not use blade ejectors unless instructed to do so by customer.*
3. *Limit switches are required with hydraulic ejection.*
4. *Do not locate ejector pin too close to deep vertical walls or ribs where part could hang up on pins when ejected, and/or steel might crack.*
5. *Sleeve ejectors should be used on deep bosses. Make sure proper bearing length is maintained inside sleeve. Use sleeve extensions if required.*
6. *All ejector sleeves are to be PCS standard or equivalent. Use PCS pins with sleeves. When PCS pins are not long enough, use performance mould pins. The maximum length of ejector sleeves is to be 14". For conditions which require a longer sleeve, use a sleeve carrier.*

Figure 5.1: A sample specification document in list format

"select and click". There are four kinds of operations provided: adding, combining, deleting and splitting as table 5.1.

Table 5.1: Operations of the tool

Operation	Function
Adding	Define one or more ungrouped sentences as a new ICU
Combining	Combine two or more individual ICUs which talk about the same concept into a new one
Deleting	Delete an existing ICU whose information is not important and can be ignored
Splitting	Split an existing ICU into two individual ICUs if it talks about two individual concepts

Next step is to extract the keywords from ICUs and conduct the stemming process. ANTELOP is the main library used during this step. As ANTELOP provides a lot of functions covering almost all the aspects of natural language processing, the main components we used in this case are "Lexicon" and "Tagging". "Tagging" is used to find the part of speech of a single word (i.e.: is a word a noun, or a verb, or an adjective, or a determiner) based on electronic lexicon (i.e. WordNet). The keyword redundancy problem (i.e. one keyword may appear multiple times in a query) is not considered here because the keyword with

higher occurrence frequency should be paid more attention in the next search phrase. The following figure 5.2 shows an example of the noun recognition process.

```
//Use Tagger to get nouns in a sentence
string simpleTaggerFile = @"data\BrillTaggerLexicon.txt";
ITagger simpleTagger = new SimpleTagger(simpleTaggerFile);
//string sentence = "Rest bottons or back stops are press fit or screwed to ejector clamp
plate and must located behind return pins and ejector pins and close to cylinders for
hydraulic ejector.";
IList<IWord> words = simpleTagger.TagText(sentence);
for (int i = 0; i < words.Count; i++)
{
        if (words[i].Tag.Equals(TagType.EnglishNN)||
words[i].Tag.Equals(TagType.EnglishNNP)||
words[i].Tag.Equals(TagType.EnglishNNPS)||
words[i].Tag.Equals(TagType.EnglishNNS))
  {
    keywords.Add(words[i].Text.ToLower());
    /*Console.WriteLine(words[i].Text);
    Console.ReadLine();*/
  }
}
```

Figure 5.2: Noun recognition process

The next step after keywords are extracted from the sentence is the stemming process. Word stemming can be grouped into four cases depending on different purposes and word types as follows.

- Synonym: For a given word, synonyms are terms which have the same or very similar meanings to the given term. In practical use, the usage of synonym is very normal.
- Noun: Transform a base noun into its plural form ("*box*"-> "*boxes*") or vice versa.
- Verb: Transform a base verb into its gerund form ("*go*"-> "*going*"), third person singular form ("*go*"-> "*goes*"), past tense form ("*go*"-> "*went*"), past participle form ("*go*"-> "*gone*") or vice versa.
- Adjective: Transform a base adjective into its comparative form ("*fast*"-> "*faster*"), superlative form ("*fast*"-> "*fastest*") or vice versa.

In our case, noun is the only type of words we need. And also, synonyms are added in the knowledge base so the only stemming needed here is noun stemming which can be implemented in following format provided by the "Lexicon" component of ANTELOP. Moreover, all the concepts are defined in base form and plural form is not allowed in the ontology. So what we need to do here is transforming all the plural nouns in the sentence into their base form as in the following sample 5.3.

```
//Use Lexicon to get base form of plural word
        ILexicon lexicon = new Lexicon();
        lexicon.LoadDataFromFile(@"data\Proxem.Lexicon.dat", null);
        for (int i = 0; i < keywords.Count; i++)
        {
          IList<IInflectedWord> baseWords = lexicon.GetBaseForms((string)keywords[i],
PartOfSpeech.Noun);
          if (baseWords.Count != 0)
          {
            keywords[i] = baseWords[0].BaseForm;
          }
        }
```

Figure 5.3: Transforming the plural nouns into base form

As mentioned in section 4.2.1, another important work when generating the keywords from sentence is the compound word recognition. Although ANTELOPE also provides a function to recognize the compound word in English, it only works well for some general words such as "*log cabin*" or "*United States*". For a specific domain such as mould engineering in our case, the accuracy of recognition is disappointing because ANTELOPE is based on a general lexicon WordNet, not a professional domain specific glossary. Also, the quantity of compound words in mould engineering domain is limited so it's possible to create our own compound word glossary. By using this glossary, a compound word in the sentence is extracted as a special keyword instead of a couple of individual nouns.

5.2.2 Create Ontology for Mould Engineering Domain

The domain knowledge base in this project is mainly contributed and created by the domain expert from Cornerstone Intelligent Software Corp. An ontology development tool called Protege is used during the work to assist the development. As discussed in section 4.2.2, Protege provides a user-friendly graphical user interface as shown in Figure 5.4 so it can be easily studied and used. There are two main sources of the information used to develop the ontology. The concepts coming from data repository are called key concepts and they are the final targets of our search. Related concepts are found in some mould design textbooks to describe the key concepts and provide additional information to make the ontology more complete. The main operations of Protege used during the development work are defining concept, defining instance, defining relationship and adding relationship between concepts. The hierarchy structure shown in the left side of Figure 5.4 represents the "subclass" relationships between concepts and new concepts can be added here. As the same in object-oriented programming principle, the subclass inherits all the attributes and relationships from his parent class. Instance, also called individual, is a special kind of concept. If a concept is treated as an abstract for a group of objects which have the same features, an instance is one object in the group. For example, "*company*" is a concept to represent the business organization then "*Microsoft*" is an instance of "*company*".

After creating the ontology, an OWL file is exported by Protege for further use. To extract the information from OWL file, an ontology parser is needed to convert OWL file to the graph data structure recognized by the proprietary program of the industrial partner. Jena is one of the most widely used Java APIs for RDF and OWL, providing services for model representation, parsing, querying and some visualization tools. Unfortunately, Jena only supports Java and in our case the programming language is C-sharp. So instead, we

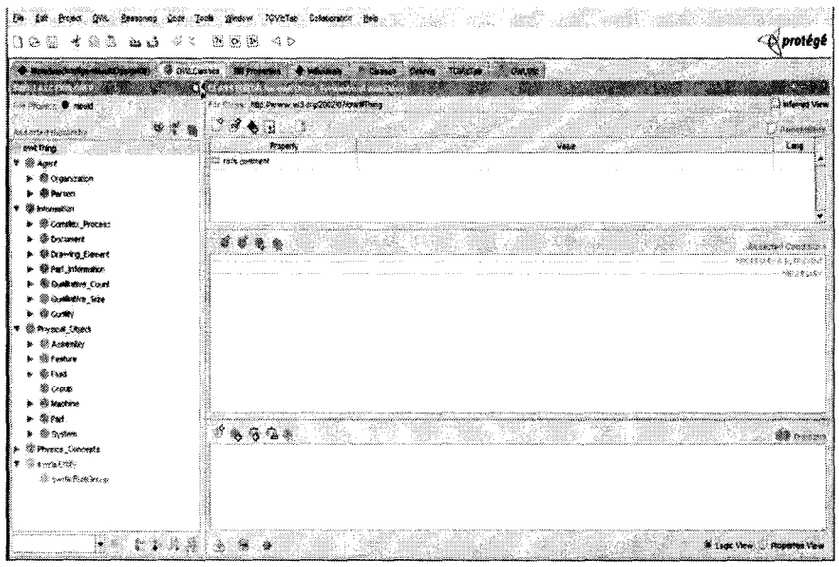

Figure 5.4: Protege User Interface

develop a simple ontology parser to convert an OWL file to a graph data structure containing a one-dimension array to store the concepts (nodes) and a matrix to keep the relationships (links) between nodes. This ontology parser is not designed for general use purpose so it can only deal with limited tags of OWL. The tags can be handled now are listed in Table 5.2.

5.2.3 Query Revision and Search Process

The queries coming from query generator and parsed ontology graph generated by ontology parser are the two inputs of this query revision process. There are four steps to implement the revision process. Firstly, for a given query Q containing a list of keywords $\{ k_1, k_2, k_3 \cdots k_k \}$ and an ontology graph G, all the nodes in G whose node name matches anyone in $\{ k_1, k_2, k_3 \cdots k_k \}$ are selected into a concept group $\{ c_1, c_2, c_3 \cdots c_m \}$. Also, the

Table 5.2: Owl tags used in our framework

Name	Function
<class>	Define a class (concept)
<rdfs:subClassOf>	Indicate if a class is a subclass of another class
<equivalentClass>	Indicate if two or more classes represent the same concept
<rdf:Property>	Define a relationship between classes
<inverseOf>	One property may be stated to be the inverse of another property. For example, hasPart may be stated to be inverse of isPartOf
<rdfs:domain>	Limit the classes or individuals to which the property can be applied
<rdfs:range>	Limit the classes and individuals that the property may have as its value
<Individual>	Define the individuals as instances of classes
<disjointWith>	State classes to be disjoint from each other

same concept may be selected more than once to keep the keyword occurrence frequency information in the original query. In another word, the concept in $\{ c_1, c_2, c_3 \cdots c_m \}$ is not unique so $c_1 = c_m$ may happen. Then in the second step, the semantic distance from c_1 to all the other nodes in G are calculated as $d1n$ $(1<n<m)$ by using Dijkstra's algorithm. Dijkstra's algorithm is a famous graph search algorithm to solve the single-source shortest path problem for a nonnegative edge path costs, producing a shortest path tree. The same step is conducted on $c_2 \cdots c_m$. After all the semantic distances are calculated, for each node a in G, the total semantic distance to $\{ c_1, c_2, c_3 \cdots c_m \}$ is calculated as following expression:

$$D(a) = \sum_{n=1}^{m} d_{an}$$

In the ranking process, the node with less total semantic distance is the more closely related node of the given query. Thus total semantic distances of nodes in G are sorted by ascending order to generate new query candidate list. As described in Section 5.2.2, there are two

kinds of nodes in *G*. Key concepts are nodes from database and they are our final search targets. So the nodes for related concepts in candidate list are filtered and the rest part of the list is our final results list. The pseudo code below 5.5 summarizes the overall revision process.

1. Algorithm **Search_Process** (*Graph G, keywords* $\{k_1, k_2, k_3 \ldots k_k\}$)
2. For all *node n* in *G* do
3. If the node name of *n* matches anyone in $\{k_1, k_2, k_3 \ldots k_k\}$
4. Add *n* into a concept group $\{c_1, c_2, c_3 \ldots c_m\}$
5. For all *concept c* in $\{c_1, c_2, c_3 \ldots c_m\}$ do
6. Find shortest path from *c* to all node *n* in *G* as $P(c, n) = Dijkstra(c, n)$
7. For all node *n* in Graph *G*
8. $D(n) = \sum P(c, n)$
9. Rank all node *n* in Graph *G* by *D(n)* in *ascending* order

Figure 5.5: Overall search process pseudo code

The time complexity of Dijkstra's algorithm is O(n2) so the complexity of the ontology-based search process is O(k∗n2) where n is the amount of nodes in Graph G and k is the number of keywords in the ICU (k<= n). The complexity of keyword-based search is O(k∗n) as a comparison.

5.2.4 Concept Selection Using Genetic Algorithm

As described in section 4.2.4, concept selection is a possible way to improve search during both query generating and searching phrases. The difficulty to accomplish this semantic distance based concept selection is judging the weak related concepts. For a concepts group, weak related concept is defined as the concept which is "far away" from the others. But during implementation, how far can be treated as "far away" is a real problem. To avoid this confusing situation, we introduce another novel approach to accomplish this semantic

distance based concept selection by using genetic algorithm.

In this approach, given a concept group C, the concepts in C are randomly picked up to form random sized combinations. The quantity of the combinations depends on the volume of C. From the genetic algorithms' point of view, a combination is treated as an individual and each concept in combination is taken as a gene of the individual. These randomly created combinations make up of the first generation. Then for the evolution process, a fitness function is needed to decide which combination is better for the search so it can survive. Here we define the fitness as "*The combination whose concepts are more closely related in knowledge base has a higher fitness.*" There are several approaches to implement this work such as minimum sub tree [21]. To simplify the implementation, we use the average semantic distance. Firstly, for a concept combination *Combo*, the central point for the concepts in *Comb* is found by using the same way as described in section 5.2.3. Then the average semantic distance for *Comb* is got by dividing the sum of distances with the amount of concepts in *Comb*.

Based on the fitness, a percentage of combinations with high fitness survive and others are washed out. Then the survived combinations produce the next generation by mutation (randomly change one concept into another from the initial concept group C) and crossover (exchange some concepts with another survived combinations). After several generations, the bad genes are washed out. Or in another word, the weak related concepts in C are taken out and the remaining combinations are expected to be the better query for the search phase.

5.3 Experiments

5.3.1 Experiments of Concept Search

We assess our concept search approach versus classical keyword-based approach with the same domain and queries. Our proposed ontology-based concept search approach has been described in detail in Chapter 4. For the baseline classical keyword-based search, query generation part is implemented in the same way as the concept search. Following the common query generation component, there is no knowledge base and the search is directly conducted on the data repository. The most significant difference between keyword-based search and concept search is the knowledge base layer. Two experiments are set and conducted on both approaches based on different test cases and conditions.

The ejection process of mould engineering is selected as our experiment domain. After the product has cooled in the cavity and the mould has opened, the product must be ejected from the mould [38] and this process is called the ejection process. We select this process as our experiment domain because it is relatively specific and independent. The components designed for this process are shown in Fig 5.6. In our experiments, ontology created for this process contains 92 concepts: 44 key concepts and 48 related concepts. A specification document about ejection process provided by Cornerstone Intelligent Software Corp. is taken as the query document for the experiments. This query document contains 35 sentences and can be grouped into 14 ICUs. Also, the data repository containing 44 concepts about ejection process are provided by the company from their existing database of application. The goal of our search is to identify the concepts in the specification and match them to the concepts in the proprietary data repository. In our experiment, we assume there is one and only one unique match in the data repository for each query.

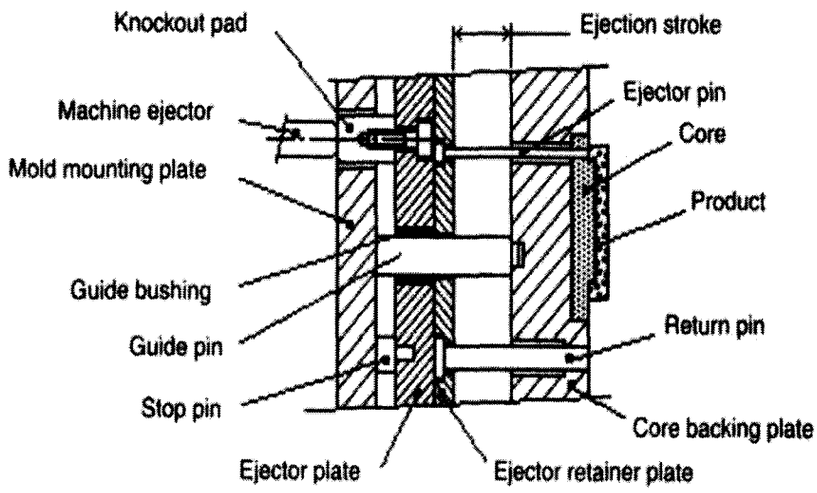

Figure 5.6: Typical mould with ejector pins [38]

The output format of the experiment is shown as in Table 5.3. The results of the query are evaluated by being grouped into three kinds of situations. If the first one on the ranked result list is the right target of the query, it is called "Perfect Match". Then, if the right result is found on any position of the list but not the first one, we call it "Found". And if there is no right result in the result list, it is treated as "Not Found". The aim of our model is to maximize the percentage of "Perfect Found" and minimize "Not Found".

Table 5.3: Sample output of the experiment

Query	Target Concept	Found on Position	Found State
ICU 1	Concept 1	1	Perfect Found
ICU 2	Concept 2	5	Found
ICU 3	Concept 3	NA	Not Found
...	...	...	...

Experiment 1 is the initial experiment to compare our ontology-based concept search

approach to classical keyword-based search. The document taken as the query is formal and ambiguity-free. The statistic comparison result is shown in Fig 5.7.

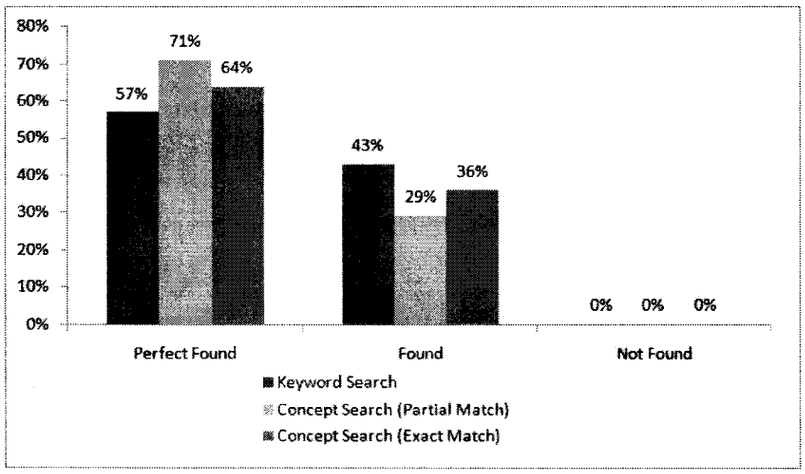

Figure 5.7: Result comparison for a document without ambiguous words

The test cases for experiment 2 are slightly modified. We add some spelling mistakes and replace some of the words in the document with synonyms or other kind of expressions. For example, for an ambiguity-free sentence "*Do not locate ejector pin too close to deep vertical walls or ribs*". We replace "*ejector*" by a wrong spelling "*ejecter*" and "*wall*" by its synonym "*peg*" to form a new sentence "*Do not locate ejecter pin too close to deep vertical pegs or ribs*". This is a very usual case in practical use and in this way the search flexibility and fault tolerant ability can be tested.

As shown in Figure 5.7 and Figure 5.8, our ontology-based concept search approach performs better for both documents with and without ambiguous words. Especially for the document that contains ambiguous words, there are about 28% cases that classical keyword search cannot find the right result, which confirms our statement that the ability of keyword

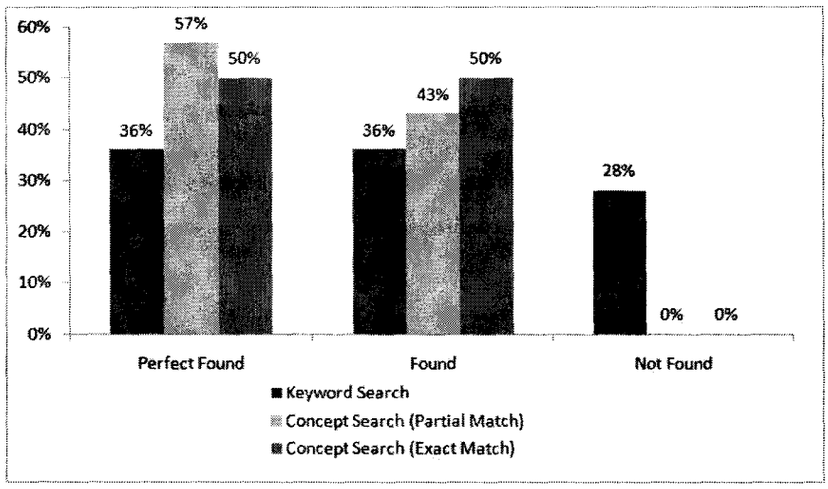

Figure 5.8: Result comparison for a document with ambiguous words

search to handle the ambiguous words is weak. Although our approach still cannot fully solve the words ambiguity problem, the decrease of search accuracy (perfect match) is only 14% while for classical keyword search it is up to 21%. The effectiveness of our ontology-based search approach is shown as our expectation. In practical use, most of documents cannot be formal and ambiguity-free, so our flexible and fault tolerant search model will be more useful than traditional keyword-based search. Also, the results show that in our approach partial match strategy is better than exact match strategy during the search process.In this case, the search ability of keyword-based search is severely decreased while our ontology-based can still work well.

5.3.2 Experiments of Concept Selection

As discussed in section 4.2.4, concept selection process in our model is used to filter the less important words in automatically generated query and the "fake" matched concepts during

partial match process. In the knowledge base of previous experiments, the occurrence of "fake" matched concepts is very low so we use another knowledge base contributed by Cornerstone Intelligent Software Corp. to show the effectiveness of our concept selection method. The new knowledge base contains 208 concepts and 20 ICUs generated from 41 sentences are used as test cases. The following figure 5.9 shows the result of our approach after using concept selection. The results shown in Figure 5.9 indicates that after using

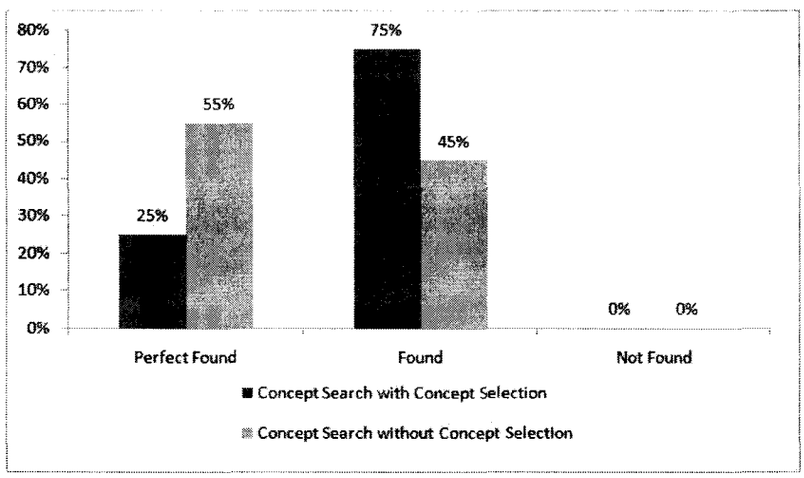

Figure 5.9: Concept search result after using concept selection

our concept selection method the search performance is even worse, which is out of our expectation. This problem may cause from our fitness function. We define the fitness in our method as "*The combination whose concepts are more closely related in knowledge base has a higher fitness.*" Under this definition, the combinations that contain fewer concepts are more likely to have higher fitness and survive. In worst cases, the final candidate combination contains only two concepts no matter how many concepts are in the initial query. Although this kind of candidate combination does filter some less important information,

some useful concepts are also neglected. To solve this problem, we add the restriction that the amount of the concepts in final candidate combination must be at least larger than half of the initial concepts number and it may vary for different cases. Then the experiment result after this modification is shown in the following figure 5.10.

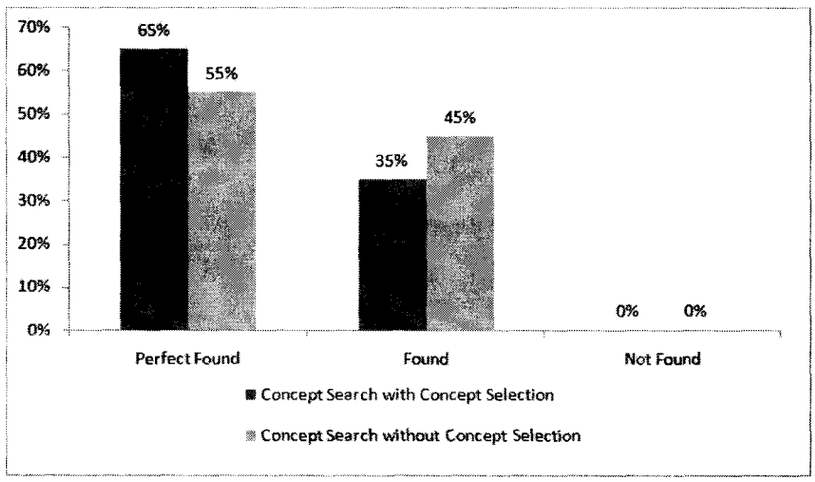

Figure 5.10: Concept search result after using modified concept selection

Figure 5.10 shows the benefits to use our concept selection method based on semantic distance. Although the improvement is slight and the percentage of how many concepts should be kept is not seriously considered, the concept selection process still has a potential to improve our concept search model.

Chapter 6

Conclusions

This thesis proposes a query revision method by narrowing down a query into one concept using semantic distance. Based on this method, an ontology-based concept search model which can be considered as an evolution of classical keyword-based search technique is introduced to solve the problem about how to extract the information from a document into an existing database. According to the results reported from the experiments, our approach improves the search accuracy for 14% compared to the classical keyword-based search. For the documents with ambiguous words, our model works muck better than the keyword search for 21% increase on accuracy and 28% improvement on coverage. Consequently, this demonstrates that our approach can appropriately solve the ambiguity problem brought by the practical use and is more flexible and fault tolerant. Also, the concept selection method has a potential to improve the concept search. Our proposed model is a generic model which can also be applied in other specific domain with the changing of the domain knowledge. Although the case study was limited to the current industry's specific domain in mould design, one can easily apply the search and ranking methods into any other domain ontology due to the portability and relative independence of the ontological layer.

Our model is designed to solve the practical problem, however there are still some shortages which limit its commercialization. The first one is about the revision method. Our query revision method is designed particularly for the data repositories with limited information but not fit for the large data repositories since it is too risky, which may seriously reduce its scale of usage. Second, the search performance of our model heavily relies on the quality of the ontology. Although ontology has the benefits of strong representation power, easily used tools and outstanding reusability, how to create a high quality ontology is still a open problem. Moveover, now the ontology creation process is still manual so even our model can automate the revise and search process, a lot of human evolvements are also needed.

Future work will focus on two aspects. One is expanding our ontology to increase the domain size to test our approach in a larger scale for larger experiment samples. The other is to improve the components of the model to get a more effective concept search approach. More improvements can be made including using more features of ontology such as relation weights and constrains, adopting more advanced natural language processing techniques to exploit semantic information such as verbs, numbers or locations and taking historical knowledge into account. Our approach still has a large room for improvement to achieve higher accuracy, while the keyword-based search is hard to be improved and further exploited.

Bibliography

[1] A. Abu-Hanna and W. Jansweijer. Modeling domain knowledge using explicit conceptualization. *IEEE Expert: Intelligent Systems and Their Applications*, 9(5):64, 1994.

[2] M. J Bates. Subject access in online catalogs: A design model. *Journal of the American Society for Information Science*, 37(6):357376, 1986.

[3] M. Baziz, M. Boughanem, and N. Aussenac-Gilles. Conceptual indexing based on document content representation. *Information Context: Nature, Impact, and Role*, page 171186, 2005.

[4] M. Beaulieu and S. Jones. Interactive searching and interface issues in the okapi best match probabilistic retrieval system. *INTERACT COMPUT*, 10(3):237248, 1998.

[5] J. Bhogal, A. Macfarlane, and P. Smith. A review of ontology based query expansion. *Information processing & management*, 43(4):866886, 2007.

[6] W. N Borst. Construction of engineering ontologies for knowledge sharing and reuse. 1997.

[7] W. N. Borst, J. M. Akkermans, and J. L. Top. Engineering ontologies. *International Journal of Human-Computer Studies*, 46(2-3):365406, 1997.

[8] R. J Brachman and H. J Levesque. *Knowledge representation and reasoning. 2004.* San Francisco: Morgan Kaufmann Publishers.

[9] S. Calegari and G. Pasi. Personalized ontology-based query expansion. In *IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology, 2008. WI-IAT'08*, volume 3, 2008.

[10] S. Deerwester, S. T Dumais, G. W Furnas, T. K Landauer, and R. Harshman. Indexing by latent semantic analysis. *Journal of the American society for information science*, 41(6):391407, 1990.

[11] T. Efraim, M. L Ephraim, and W. James. *Information Technology for Management: Making Connections for Strategic Advantage. New York: J.* Wiley, 1999.

[12] E. N Efthimiadis. Query expansion. *Annual review of information science and technology (ARIST)*, 31:12187, 1996.

[13] E. A Feigenbaum and P. McCorduck. *The fifth generation: artificial intelligence and Japan's computer challenge to the world.* Addison-Wesley Publishing Co., Reading, MA, 1983.

[14] D. Fensel. Ontologies: Silver bullet for knowledge management and electronic commerce. 2001. *Berlin: Spring-Verlag.*

[15] D. Fensel and R. Groenboom. Specifying knowledge-based systems with reusable components. In *in Proceedings of the 9th International Conference on Software EngineeringKnowledge Engineering (SEKE-97*, 1997.

[16] L. Finkelstein, E. Gabrilovich, Y. Matias, E. Rivlin, Z. Solan, G. Wolfman, and E. Ruppin. Placing search in context: The concept revisited. *ACM Transactions on Information Systems (TOIS)*, 20(1):116131, 2002.

[17] G. Fu, C. B Jones, and A. I Abdelmoty. Ontology-based spatial query expansion in information retrieval. *On the Move to Meaningful Internet Systems 2005: CoopIS, DOA, and ODBASE*, page 14661482, 2005.

[18] G. W. Furnas, T. K. Landauer, L. M. Gomez, and S. T. Dumais. The vocabulary problem in human-system communication. *Communications of the ACM*, 30(11):971, 1987.

[19] J. H Gennari, S. W Tu, T. E Rothenfluh, M. A Musen, et al. Mapping domains to methods in support of reuse. *International journal of human computer studies*, 41(3):399424, 1994.

[20] D. E Goldberg. *Genetic Algorithms in Search and Optimization*. Addison-wesley, 1989.

[21] K. Golenberg, B. Kimelfeld, and Y. Sagiv. Keyword proximity search in complex data graphs. In *Proceedings of the 2008 ACM SIGMOD international conference on Management of data*, page 927940, 2008.

[22] J. Gonzalo, F. Verdejo, I. Chugur, and J. Cigarran. Indexing with WordNet synsets can improve text retrieval. *Arxiv preprint cmp-lg/9808002*, 1998.

[23] T. R Gruber et al. A translation approach to portable ontology specifications. *Knowledge acquisition*, 5:199199, 1993.

[24] G. Van Heijst, A. T Schreiber, and B. J Wielinga. Using explicit ontologies in KBS development. *International Journal of Human Computer Studies*, 46:183292, 1997.

[25] W. R Hersh, R. T Bhupatiraju, and S. Price. Phrases, boosting, and query expansion using external knowledge resources for genomic information retrieval. In *TREC*, page 503509, 2003.

[26] L. Huang. A survey on web information retrieval technologies. *ECSL. State University of New York, New York*, 2000.

[27] G. Jiang, K. Ogasawara, A. Endoh, and T. Sakurai. Context-based ontology building support in clinical domains using formal concept analysis. *International journal of medical informatics*, 71(1):7181, 2003.

[28] K. Sparck Jones. *Automatic keyword classification for information retrieval*. London, 1971.

[29] K. Sparck Jones. An evaluation of query expansion by addition of clustered terms for a document retrieval system. *Information Storage and Retrieval*, 9(6):339, 1973.

[30] L. Khan and F. Luo. Ontology construction for information selection. 2002.

[31] A. Kiryakov, B. Popov, D. Ognyanoff, D. Manov, A. Kirilov, and M. Goranov. Semantic annotation, indexing, and retrieval. *The SemanticWeb-ISWC 2003*, page 484499, 2003.

[32] M. E. Lesk. Word-word associations in document retrieval systems. *American Documentation*, 20(1):27438, 1969.

[33] C. D Manning, P. Raghavan, and H. Sch\tze. An introduction to information retrieval.

[34] K. Meffert and N. Rotstan. A brief introduction to genetic algorithms. `http://jgap.sourceforge.net/doc/gaintro.html`, April 2010.

[35] R. Navigli and P. Velardi. An analysis of ontology-based query expansion strategies. In *Workshop on Adaptive Text Extraction and Mining*, page 4249, 2003.

[36] T. Nguyen and T. Phan. An ontology-based approach of query expansion. In *Proceedings of the 9th International Conference on Information Integration and Web-based Application & Services (IIWAS 2007), Jakarta, Indonesia*, 2007.

[37] K. Nilsson, H. Hjelm, and H. Oxhammar. SUiScross-language ontology-driven information retrieval in a restricted domain. In *Proceedings of the 15th NODALIDA conference*, 2005.

[38] R. Rees. *Mould Engineering*. Distributed by Hanser Gardner Publications, Inc., Ohio, 2002.

[39] I. Rithven, M. Lalmas, and K. van Rijsbergen. Empirical investigations on query modification using abductive explanations. In *Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval*, page 181189, 2001.

[40] J. J. Rocchio. *The SMART Retrieval System - Experiments in Automatic Document Processing*. Prentice Hall, 1971.

[41] S. Russell and P. Norvig. Artificial intelligence: A modern approach. Prentice-Hall. *New York, New York*, 2002.

[42] G. Salton and M. J McGill. *Introduction to modern information retrieval*. McGraw-Hill New York, 1983.

[43] Proxem semantic (r)evolution. Proxem semantic (r)evolution. http://www.proxem.com/Default.aspx, April 2010.

[44] R. M Setchi and Q. Tang. Concept indexing using ontology and supervised machine learning. *Trans. on Engineering, Computing and Technology*, 19:221226, 2006.

[45] A. Singhal. Modern information retrieval: A brief overview. *IEEE Data Engineering Bulletin*, 24(4):3543, 2001.

[46] R. Studer, V. R Benjamins, and D. Fensel. Knowledge engineering: principles and methods. *Data & Knowledge Engineering*, 25(1-2):161197, 1998.

[47] R. Studer, H. Eriksson, J. Gennari, S. Tu, D. Fensel, and M. Musen. *Ontologies and the configuration of problem-solving methods*. Citeseer, 1996.

[48] S. M Thede. An introduction to genetic algorithms. *Journal of Computing Sciences in Colleges*, 20(1):115123, 2004.

[49] K. Trentelman, DEFENCE SCIENCE, and TECHNOLOGY ORGANISATION EDINBURGH (AUSTRALIA). Survey of knowledge representation and reasoning systems. 2009.

[50] G. van Heijst. The role of ontologies in knowledge engineering. *PhDthesis, University of Amsterdam*, 1995.

[51] O. Vechtomova and Y. Wang. A study of the effect of term proximity on query expansion. *Journal of information science*, 32(4):324, 2006.

[52] E. M Voorhees. Using WordNet to disambiguate word senses for text retrieval. In *Proceedings of the 16th annual international ACM SIGIR conference on Research and development in information retrieval*, page 171180, 1993.

[53] E. M Voorhees. Query expansion using lexical-semantic relations. In *Proceedings of the 17th annual international ACM SIGIR conference on Research and development in information retrieval*, page 6169, 1994.

[54] A. Voss, K. Nakata, and M. Juhnke. Concept indexing. In *Proceedings of the international ACM SIGGROUP conference on Supporting group work*, page 10, 1999.

[55] Wikipedia. Concept search. http://en.wikipedia.org/wiki/Concept_search, April 2010.

[56] W. A Woods. Conceptual indexing: A better way to organize knowledge. *Sun Microsystems, Inc. Mountain View, CA, USA*, 1997.

[57] X. Xu, W. Zhu, X. Zhang, X. Hu, and I. Y Song. A comparison of local analysis, global analysis and ontology-based query expansion strategies for bio-medical literature search. In *IEEE International Conference on Systems, Man and Cybernetics, 2006. SMC'06*, volume 4, 2006.

Appendix A

A Detailed Sample of Search Process

In this part I will give a detailed example to describe how our concept search model works. A part of the specification document taken as the query source is shown as the following figure:

1. Is automatic accelerated ejection required?
2. Do not use blade ejectors unless instructed to do so by customer.
3. Limit switches are required with hydraulic ejection.
4. Do not locate ejector pin too close to deep vertical walls or ribs where part could hang up on pins when ejected, and/or steel might crack.
5. All ejector pins are to be PCS standard or equivalent.
6. Use a std. PCS ejector pin for return pins #NP-47 36'' max. Length Min. 4 req'd. per tool, use 6 when possible.
7. Sleeve ejectors should be used on deep bosses.
8. Make sure proper bearing length is maintained inside sleeve.
9. Use sleeve extensions if required.

Figure A.1: A specification document sample

Given a specification document, the sentences describing the same concept are grouped into one individual concept unit. In this case, sentences #4 to #6 are talking about the same concept *ejector pin* so they are grouped into one individual concept unit to form the initial query: *Do not locate ejector pin too close to deep vertical walls or ribs where part*

could hang up on pins when ejected, and/or steel might crack. All ejector pins are to be PCS standard or equivalent. Use a std. PCS ejector pin for return pins #NP-47 36" max. Length Min. 4 req'd. per tool, use 6 when possible.

Then the next step is to extract the keywords from the query. Here we select nouns as our keywords so a keyword list is generated from the query as follows: *ejector, pin, walls, ribs, part, pins, steel, ejector, pins, PCS, standard, equivalent, ejector, pin, return, pins, 47, 36, length, 4, tool, use, 6* A stemming process is used to transform all the plural form nouns into their base form then we can get the updated keyword list: *ejector, pin, wall, rib, part, pin, steel, ejector, pin, PCS, standard, equivalent, ejector, pin, return, pin, 47, 36, length, 4, tool, use, 6* After the stemming process the compound words in the keyword list are identified to replace the single words. In this case the compound words are *ejector pin* and *return pin* and the keyword list is finalized as follows: *ejector pin, wall, rib, part, pin, steel, ejector pin, PCS, standard, equivalent, ejector pin, return pin, 47, 36, length, 4, tool, use, 6* After keywords are generated, the next phrase is the query revision process. The first step in this process is to find the matching concepts for the keywords. To simplify the demo process, a small knowledge base is used. Here we give the example by using the exact matching strategy so the matching concepts are listed as follows: *ejector pin, pin, ejector pin, ejector pin, return pin* Since the number of the matching concepts in this case is small, the concept selection technique is not adaptable here. Then in the knowledge base, *ejector pin* and *return pin* are the subclasses of *pin*. So the semantic distances between these concepts are:

S (ejector pin <-> pin) = 1

S (return pin <-> pin) = 1

S (return pin <-> ejector pin) = 2

The semantic distance between a concept and itself is 0. So the total semantic distance for *ejector pin* is 3 (0 + 1 + 0 + 0 + 2) while it is 4 (1 + 0 + 1 + 1 + 1) for *pin* and 7 (2 + 1 + 2 + 2 + 0) for *return pin*. The total semantic distances for other concepts in the knowledge base are also calculated but the minimum one is still *ejector pin*. Also the concept *ejector pin* comes from the database as keyword concept so it is ranked first on our candidate list to represent the original query. Then the concepts in the candidate list are searched in the proprietary database to retrieval the final results.

Appendix B

Test Cases for the Experiments

In this part the test cases used in the experiments are listed.

B.1 Test Cases for Ontology Based Concept Search Model

1. Is automatic accelerated ejection required?

2. Do not use blade ejectors unless instructed to do so by customer.

3. Limit switches are required with hydraulic ejection.

4. Do not locate ejector pin too close to deep vertical walls or ribs where part could hang up on pins when ejected, and/or steel might crack.

5. All ejector pins are to be PCS standard or equivalent.

6. Use a std. PCS ejector pin for return pins #NP-47 36" max. Length Min. 4 req'd. per tool, use 6 when possible.

7. Sleeve ejectors should be used on deep bosses.

8. Make sure proper bearing length is maintained inside sleeve.

9. Use sleeve extensions if required.

10. All ejector sleeves are to be PCS standard or equivalent.

11. Use PCS pins with sleeves.

12. When PCS pins are not long enough, use performance mould pins.

13. The maximum length of ejector sleeves is to be 14".

14. For conditions which require a longer sleeve, use a sleeve carrier.

15. Ejector sleeve identification shall be located on the bottom of the pin head and adjacent to the c'bore for the pin head.

16. Use letters to identify ejector sleeves.

17. Flow dividers will be used on ejector systems and will be rated for 3,000 psi continuous service.

18. Use JSB "D series" Rotary Flow Dividers. Maximum pressure available from machine is 2,150 psi.

19. Flow dividers will be placed on the lower side of the mould and positioned on the ejector rail or core/cavity block and not on the QMC or clamp plates.

20. Ejector retainer plate is normally .75" to 1.00" thick.

21. Ejector plate is normally 1.50" to 4.00" thick.

22. Ejector plates over fifty pounds require handling holes. Two per side on two opposite sides.

23. Rest buttons or back stops are press fit or screwed to ejector clamp plate and must be located behind return pins and ejector pins and close to cylinders for hydraulic ejection.

24. Always place a rest button under a under pin.

25. Design rest buttons between ejector screws. Use PCS rest buttons.

26. Ejector guide bushings must be to customer standards and must have provision for lubrication.

27. If graphic impregnated bushings are permissible no lubrication is required.

28. Length of bushings should allow as much bearing as possible.

29. Return pin shown with mould in open position.

30. The minimum diameter of return pins will be one inch unless otherwise specified by Saturn Manufacturing Engineering.

31. Use six return pins for moulds over 28 inches in length.

32. Return pins on any tool shall not obstruct part removal.

33. Return pin contact surface against cavity shall be flat and perpendicular to mould travel. Top of the pin to be chamfered.

34. A minimum of four pins per mould are required.

35. Four ejector guide pins will always be used. One guide pin will be offset at the "0" corner to prevent disassembly.

B.2 Test Cases for Concept Selection

1. Ejector guide bushings must be to customer standards and must have provision for lubrication.

2. If graphic impregnated bushings are permissible no lubrication is required.

3. Length of bushings should allow as much bearing as possible.

4. Return pin shown with mould in open position.

5. The minimum diameter of return pins will be one inch unless otherwise specified by Saturn Manufacturing Engineering.

6. Use six return pins for moulds over 28 inches in length.

7. Return pins on any tool shall not obstruct part removal.

8. Return pin contact surface against cavity shall be flat and perpendicular to mould travel. Top of the pin to be chamfered.

9. All locating rings shall be 3.990" nominal size (DME or equivalent), to be confirmed by the Polybrite Project Engineer.

10. Nozzle radius for sprue bushings to be .750" spherical radius with a high polished surface finish (please see Figure 1).

11. Sprue height to be kept to a maximum of 2-3 inches whenever possible.

12. All sprue bushings shall be located to keep from rotating.

13. Sprue orifice diameter should 0.030" larger than the machine nozzle orifice diameter.

14. Sprue bushings with extensive draft (4' - 5' taper for T.P.O material) or a long sprue, resulting in a heavy sprue to runner section, are permitted to have one .250" x 45E rib per runner to support the sprue upon ejection.

15. Runners on all moulds to be round or modified trapezoid.

16. Size of runner to be determined for specific mould design and material by the Polybrite Project Engineering in conjunction with the resin supplier.

17. All runners shall be constructed to a balanced lay-out, unless conditions justifies an unbalanced pattern.

18. A cold slug "chamber" shall be provided at the sprue puller and at the ends of the runners.

19. All runners shall be smooth and free of nicks and other imperfections.

20. Runner shut-off switches or blocks, made out of P-20/4140 material, can be incorporated in the runner design lay-out and have to be accessible while in the moulding machine.

21. The maximum angle for angular lifters is 12 degree. In cases where a greater angle is required, the application must be referred to the Polybrite Project Engineering for approval

prior to completion of the lifter design.

22. The maximum ejections stroke of the specified moulding machine is to be used to minimize lifter angle unless otherwise specified by the Polybrite Project Engineer. A minimum length bearing bushing of 2 times the rod diameter is to be used to for all angular lifter slides.

23. All bushing should be graphite impregnated ALBr. The maximum allowable lifter rod to bushing clearance on thick mould half sections is 0.005".

24. All angular lifters are to be fastened to the lifter slide via a self-locking screw, i.e., nylok or equivalent.

25. Clearance hole included through ejector plate and clamp plate in line with the lifter bar for disassembly or lifter without major disassembly of ejector mechanism.

26. Lifters across parting lines are not to be used without approval from the Polybrite Project Engineer.

27. All lifters are to be consecutively numbered to assist Polybrite with lifter maintenance programs. Each lifter component (lifter base, shaft face, etc.) is to have the lifter number stamped on it.

28. The actual lifter shaft length is to be stamped on all lifter shafts to aid in the manufacture of replacement shafts should breakage occur.

29. Lifters that weigh over 50lbs are to have handling holes (drilled and tapped) for removal and installation purposes.

30. Holes to be plugged (with a copper plug) to prevent plastic from entering threads.

31. Lifter housing to be one solid piece construction (P20), inserted in 1/2" deep pocket in ejector plate to be gibbed and plated in the ejector plate.

32. Adequate cooling (water channels if necessary) where large areas are in contact with

moulded plastic. If the size of the lifter/insert is not adequate for cooling, alternative materials with a higher heat transfer rate (i.e. Aluminum Bronze) must be used.

33. All hydraulic cylinders to be "Miller". Standard size cylinders shall be designed if possible to utilize interchangeability between moulds and reduce inventory.

34. A flow divider has to be installed when two cylinders or more are moving one assembly simultaneously, Delta/Barnes or equivalent.

35. Hydraulic system to be tested at 3,000 P.S.I; However, the system must fully operate at a pressure of 2,000 P.S.I.; leak free.

36. Hydraulic lines are to be drilled in the mould shoe, back plates, parallels, etc. to avoid the use of external lines as much as possible.

37. Stand-off "legs" shall be incorporated in the design to prevent cylinders, etc. from being damaged. Wherever possible, hydraulic cylinders should be mounted on the bottom or side (lowest point of gravity) of the tool to avoid leakages onto the part surface.

38. Mold to have 3/16" wide /deep hydraulic /water diversion channel if required. Hydraulic lines to be piped through steel to limit hard line exposure and leaking.

39. Hydraulic manifolds, required when more than one cylinder is needed shall be installed on the bottom non-operator side of the mould.

40. All hosing shall be designed so as to be easy to install and remove without dismantling other components.

41. Hosing must be located so as not to interfere with removal of other mould components such as slides and cores, nor should it in any way interfere with water connections and should be fastened securely to mould wherever possible.

Vita Auctoris

Ding Chen was born in Shanghai, China. He received his bachelor degree in software engineering at Beijing University of Posts and Telecommunications in 2007. Currently, he is completing his master in computer science at the University of Windsor and expects to graduate in Summer 2010.