The Hong Kong Polytechnic University

Department of Computing

Acquisition of Domain Concepts and Ontology Construction

Cui Gaoying

A Thesis Submitted in Partial Fulfillment of the

Requirements for the Degree of Doctor of Philosophy

October 15, 2009

CERTIFICATE OF ORIGINALITY

I hereby declare that this thesis is my own work and that, to the best of my knowledge and belief, it reproduces no material previously published or written, nor material that has been accepted for the award of any other degree or diploma, except where due acknowledgement has been made in the text.

_____(Signed)

Cui Gaoying_____(Name of student)

Abstract

Ontology is an area of research that gets off to a flying start and attracts inceasing attention in many branches of computer science as well as other relevant subjects. As a knowledge structure, ontology can be applied to different areas where domain knowledge is needed. Avoiding the weak points of manual methods of constructing ontologies in the top-down direction, such as time consuming and difficult to update, automatic or semi-automatic ontology construction methods usually start from the bottom with the acquisition of domain concepts from a proper corpus and then build the relations among these concepts.

In this research, a series of studies have been conducted and the whole work can be divided into two aspects include the work for bottom-up construction of ontology and top-down construction of ontology. For bottom-up ontology construction, FCA method is used first to build an domain ontology, the focus of this method is to select an appropriate attribute set for the given object set. Considering the limitation of FCA method, Wikipedia, the world's largest online encyclopedia, is exploited for the acquisition of the corpus. Then, based on the Wikipedia, concepts and attributes are extracted according to the context information and instance information in Wikipedia, such as the {{Infobox}} structures, definition sentences, and category labels. For top-down construction of ontology, domain terms that are labels of concepts are classified through given Part-of-Speech tags as prilimiary classification information.

Publications Arising from the Thesis

[1] **Cui, G.Y.**, Lu, Q., Li, W.J., and Chen, Y.R., 2007. "Attributes Selection in Ontology Taxonomies Acquisition with FCA", in the *Proceedings of the Chinese Lexical Semantics Workshop* (CLSW2007), pp. 210-215, Hong Kong, May 21-23, 2007.

[2] **Cui, G.Y.**, Lu, Q., and Li, W.J., 2008. "Preliminary Chinese Term Classification", in the *Proceedings of the 6th Workshop on Asian Language Resources* (ALR6-IJCNLP workshop), Hyderabad, India, January 11-12, 2008.

[3] **Cui, G.Y.**, Lu, Q., Li, W.J., and Chen, Y.R., 2008. "Corpus Exploitation from Wikipedia for Ontology Construction", in the *Proceedings of the International Conference on Language Resources and Evaluation* (LREC 2008), Marrakech, Morocco, May 28-30, 2008.

[4] **Cui, G.Y.**, Lu, Q., Li, W.J., and Chen, Y.R., 2008. "Attributes Selection in Chinese Ontology Acquisition with FCA", in the *International Journal of Computer Processing of Oriental Languages* (IJCPOL), Volume 21, No. 1, pp. 77-95.

[5] **Cui, G.Y.**, Lu, Q., Li, W.J., and Chen, Y.R., 2009. "Automatic Acquisition of Attributes for Ontology Construction", in the *22nd International Conference on the Computer Processing of Oriental Languages* (ICCPOL2009), pp. 248-259, Hong Kong, Mar. 26-27, 2009.

[6] **Cui, G.Y.**, Lu, Q., Li, W.J., and Chen, Y.R., 2009. "Mining Concepts from Wikipedia for Ontology Construction", in the Workshop NLPOE (Natural Language Processing and Ontology Engineering) for the 2009 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology (WI/IAT 2009), Milan, Italy, Sep.15-18, 2009.

[7] Chen, Y.R., Lu, Q., Li, W.J., and **Cui, G.Y.**, 2007. "Discovering Kind-of Relation for Ontology Construction", in the *Proceedings of the Chinese Lexical Semantics Workshop* (CLSW2007), Hong Kong, May 21-23, 2007.

[8] Chen, Y.R., Lu, Q., Li, W.J., Li, W.Y, Ji, L.N., and **Cui, G.Y.**, 2007. "Automatic Construction of a Chinese Core Ontology from an English-Chinese Term Bank", in the *Proceedings of the ISWC2007 Workshop OntoLex07 - From Text to Knowledge: The Lexicon/Ontology Interface*, pp. 78-87, Busan, Korea, 2007.

[9] Chen, Y.R., Lu, Q., Li, W.J., and **Cui, G.Y.**, 2008. "Chinese Core Ontology Construction from a Bilingual Term Bank", in the *Proceedings of the International Conference on Language Resources and Evaluation* (LREC 2008), pp. 2125-2132, Marrakech, Morocco, May 28-30, 2008.

[10] Chen, Y.R., Lu, Q., Li, W.J., and **Cui, G.Y.**, 2009. "A Novel Method to Improve Chinese Core Ontology Construction with Terms Sharing Suffixes", in the *Proceedings of the 10th Chinese National Conference on Computational Linguistics* (CNCCL 2009), pp. 370-375, Yantai, China, July 24-26, 2009.

Acknowledgements

My deepest gratitude goes first and foremost to my chief supervisor, Prof. Qin Lu, for the guidance and encouragement she gave me in every stage of my studies. Without her constant and illuminating instruction, this thesis could not have reached its present form. During the period of my studies, she not only taught me how to overcome difficulties in my research, but also caused me to understand what a person of exemplary virtue is.

Second, I would like to express my heartfelt gratitude to Dr. Wenjie Li, my co-supervisor, for her valuable advice and warm care in the past three years. It is also my great pleasure to thank Dr. Grace Ngai and Dr. Yan Liu, for their insightful comments and helpful advice in my confirmation and guided study.

I also owe my sincere gratitude to my friends and labmates in the Chinese Computing Lab, who tirelessly gave me their help and support.

My biggest thanks must go to my parents for their constant love and the great confidence that they have shown in me throughout these years.

Table of Contents

List of Figures

List of Tables

Chapter 1
Introduction

1.1. Background and Motivations

Ontology is an area of research that has been attracting more and more attention in many branches of computer science and other relevant subjects. In the 1990s, ontology began to receive wide attention from researchers in the field of computer science. In artificial intelligence, an ontology is considered to be a controlled vocabulary that describes objects and the relations between them in a formal way (Chandrasekaran et al. 1999). This representation vocabulary provides a set of terms with which to describe the facts in some domain, while the body of knowledge using that vocabulary is a collection of facts about a domain. An ontology can also be considered to be a shared understanding of some domain of interest (Uschold and Gruninger 1996) in the area of knowledge engineering and natural language processing (NLP). Figure 1 displays an overview of ontology that expresses the intent and extent of ontology. The left part of Figure 1 shows the semantics contained in an ontology. The contents of an ontology can express, construct, and formalize the ontology. The ontology includes concepts and corresponding relations derived from the attributes of concepts. Each concept has one or more instances attached to it. The concepts can also be considered to be some sets of instances in different granularities.

Attributes of concepts are concepts in themselves, the values of which are in fact their instances. The right part of Figure 1 provides three different levels of ontologies: the top-level ontology, mid-level ontology, and domain ontology. The top-level ontology is composed of the most general and foundational concepts, which can meet the practical need of a model that has as much generality as possible to ensure reusability across different domains (Smith and Welty 2001). The domain ontology is an ontology containing all domain-relevant concepts and relations between these concepts. It is an ontology to conceptualize a specific domain. The mid-level ontology is a bridge between the very abstract top-level ontology models and the rich detail of various and specific domain ontologies.

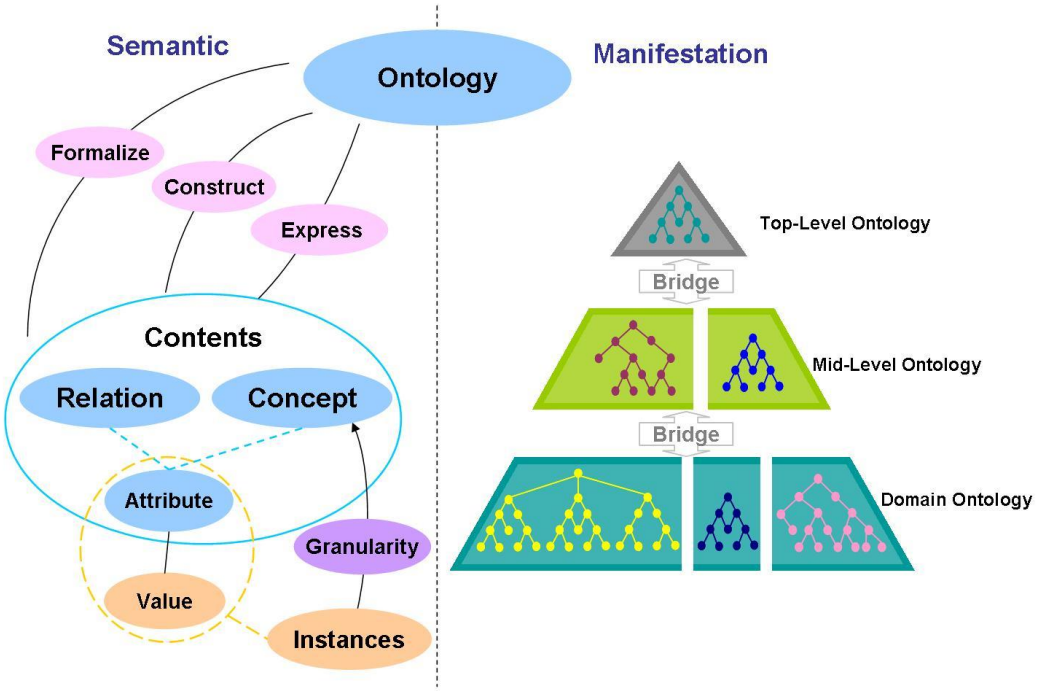

Figure 1 Overview of an Ontology

An ontology can be constructed manually or automatically. In the manual

construction of an ontology by experts, no corpus is needed as a resource. However, manual methods are costly and time-consuming, and it is also difficult to update the ontology. The suggested upper merged ontology (SUMO) is one such ontology. It was built by a working group with collaborators from the fields of engineering, philosophy, and information science (Niles et al. 2001).

In the automatic construction of an ontology, the methods can be classified in terms of two directions: top-down methods and bottom-up methods. In the top-down approach, some upper-level ontology is given and algorithms are developed to expand the ontology from the most general concepts downwards, to reach the leaf nodes where instances of concepts can be attached as leaves (Chen et al. 2007). Top-down method can make use of existing ontologies but most top-down method are employed to construct ontology manually because it is not easy to top-down build an ontology automatically and the disambiguation process is complex. By contrast, in the bottom-up approach, some domain corpus is employed to extract concepts, attributes, and relations without prior ontological knowledge. To generate a comprehensive ontology, the required domain corpus must have a good coverage of domain knowledge. Existing works have exploited different sources to serve as a corpus for the construction of an ontology. Some early works used corpora that had been manually established by domain experts (Baker et al. 1998). But manual work is usually time consuming and labour intensive. The texts are often automatically or

semi-automatically selected from books, magazines, and news organizations (Khan et al. 2002). But these corpora are not so easy to extend because the knowledge contained in the corpora is fixed by time and regions and cannot be easily updated. Others have tried to exploit a corpus from the internet, such as using the results of the Google search engine (Cimiano et al. 2004). In fact, the internet is a good source for collecting corpus data. But the construction of an ontology requires domain-specific information. Information on the classification of articles over the internet may not be very clear and it is not so easy to obtain an appropriate domain-specific corpus from internet search results. Thus, an automatic method to identify domain-specific corpuses is needed.

After acquiring a domain corpus, the general procedure in the construction of an ontology can be classified into two categories: concept acquisition and relation extraction. Current studies on concept acquisition focus on terminology extraction because a terminology is a collection of terms that are used to represent the specific concepts in a specific domain. Compared to relation extraction, terminology extraction is relatively easier work, and has been studied by many people in different domains. Most terminology extraction algorithms and systems (He et al. 2006; Zhang et al.2006; Ji et al. 2007) tend to produce a list of terms, as the results of terminology extraction and information on the classification of terminology are lacking in these works. Also, there should be different levels of terms according to their ability to

express domain knowledge. In fact, different types of terms tend to serve different roles in an ontology. Therefore, terms representing different kinds of concepts should be classified first before further steps are taken in the construction of an ontology.

Many corpus-based works build ontological structures based on a set of concept terms, which are assumed to be known or acquired by a terminology extraction algorithm from a given domain corpus (Zhou 2007; Maedche and Staab 2001; Navigli et al. 2004). Thus, efforts can be focused on identifying relations among the concept terms with some emerging tools and integrated ontology development environments, such as OntoLearn (Navigli et al. 2004), Consys (Missikoff 2002), and Text-to-Onto (Maedche and Staab 2001). However, in real corpora, concepts and their instances are usually mixed together. Concept instances sometimes appear even more often than their corresponding concepts. For example, “Microsoft” and “IBM” are instances of the concept “*company*”, and such instances may occur more frequently in certain types of corpora such as Wiki. Thus, in corpus-based ontology construction, methods for mining concepts and corresponding attributes must be developed to bridge the gap between the ontology as a concept-level structure and the corpus as an instance-rich resource.

1.2. Objectives and Scope of the Study

Considering that the purpose is to construct an ontology that uses the least amount of human resources, and taking into account the limitations and problems of previous

approaches on acquiring domain concepts and building an ontology, the main objectives of this dissertation focused on four issues: (1) to investigate an approach to exploit a domain corpus as the resource for the construction of an ontology, (2) to classify concept terms for further organizing concepts into an ontology, (3) to extract concepts from an instance-rich corpus and acquire concept attributes to further build relations between concepts, (4) to classify concept terms by assigning part-of-speech tags as preliminary classification infromtion for top-down ontology construction.

Theoretically, this research can be applied to different languages and domains. In scope, this study focuses on doing the following: (1) Building a domain ontology using the bottom-up method, FCA method is employed and the selection of attribute set for given objects is the focus of using FCA method. In the Formal Concept Analysis (FCA) method, a partial ordering relation is defined first, and attributes are used to identify relations between objects. This work is independent of language, but the experiments focus on the Chinese language. (2) Exploiting Wikipedia in English as a corpus by traversing a domain structure along the category labels in pages and ranking visited nodes by their domain-relevance scores. This is because Wikipedia in English is the largest version of Wikipedia, and therefore better than any other language version of Wikipedia for acquiring a complete domain structure. (3) Mining concepts by collecting different features of concepts in Wikipedia article pages, such as {{Infobox}} structures, category labels, definition sentences, and rankings of the

frequency of concept candidates, and combining them using the SVM model. (4) Acquiring attributes by making use of syntactic information in the contexts of given concepts and the instance information of concepts, and comparing the attribute to see which kind of information is more efficient. All of the work on mining concepts and attributes are independent of language but considering the lack of Chinese Wikipedia resources, the experiments focus on the English Wiki. (5) Giving preliminary information on classification to the extraction results of Chinese terms by applying a domain term tagging algorithm. This is because Chinese is different from English in its smallest unit of expression. In English, the smallest unit is a word split by spaces, but in Chinese that unit is a character. Also, the form of English words can help to reduce the ambiguity of a term when building an ontology. For example, “operate” is a verb and “operation” is a noun without ambiguity, but in Chinese this information is represented by only one word “操作”. Therefore, the tagging of terms is more necessary in Chinese and more influences should be considered than in English, such as word segmentation. This study focuses on the tagging of Chinese terms because the method can be simplified and moved to the English language.

1.3. Methodologies and Contributions

To achieve the objectives listed in the previous section, the following methodologies and algorithms are investigated and elaborated as follows:

1. Build domain ontology using the FCA method and select appropriate attributes set

for given objects

Formal Concept Analysis (FCA) is one formal method of acquiring generalization and specialization relation and building the structure of an ontology by making use of the inclusion relation between the attribute sets of concept terms. In using FCA to acquire relations for ontology construction, the focus is to select appropriate attribute set for given objects. Different context words including nouns, verbs, adjectives, adverbs, and their combination are investigated as possible candidate attributes of given concept terms as objects. The domain core lexicon is also considered a candidate because each entry in this core lexicon should be frequently used in its domain and, furthermore, must have strong descriptive power for other terms in the domain. This means that other terms can be built based on items in the core lexicon.

2. Exploit an appropriate domain corpus as a resource for constructing an ontology

The domain corpus to be employed in constructing an ontology must have good coverage of domain knowledge. Wikipedia, the world's largest online encyclopedia, is a good candidate as a resource for a domain corpus. However, Wikipedia is a data resource that cannot be considered domain specific without further information on classification. In this study, an algorithm is proposed to rank the domain relevance of Wikipedia pages starting from one given page name as the most relevant node to a given domain. The main idea is to follow a classification tee of a domain to identify

domain-relevant nodes from the hyperlinked pages of wikipedia by making use of category labels in Wikipedia pages. Only the contents of articles strongly linked to this classification hierarchy are considered qualified for inclusion in the domain corpus.

3. Mine concepts and attributes by making use of context information and instance information in Wikipedia pages

To bridge the gap between ontology as a concept-level structure and the corpus as an instance-rich resource, different kinds of semantic information in Wikipedia are explored and combined to mine concepts. {{Infobox}} Structures, category labels, definition sentences, and article names in Wikipedia pages, which are categorized according to instances belonging to the same concept, along with frequency rankings, are compared and combined through an SVM model to acquire reasonable concepts from Wikipedia pages.

In constructing an ontology, a concept is usually associated with a set of attributes through which relations between concepts are connected. As descriptions of concepts, attributes can be automatically acquired through the context of the concept terms in a corpus. In addition, semantic information annotated in encyclopedic corpora such as Wikipedia can help to minimize the problem of sparse data, which is more apparent in a general domain corpus for automatically extracting attributes. Both syntactic information, such as dependency relations existing in contexts of given concept terms, and the instance information of concepts can be employed to acquire

attributes for concepts from Wikipedia pages. Attribute types can also be identified according to the instance information of concepts, to disambiguate the attributes that are assigned the same name but have different semantic meanings.

4. Carry out a preliminary classification of concept terms for mapping concepts into an ontology

In building an ontology in top-down direction, ambiguity arises if terms are used as substitutes of domain concepts. For example, if one term can serve as a noun as well as a verb, there will be ambiguity if part-of-speech information is missing. Therefore, if the terms are classified before building the ontology, there will be less ambiguity in the work of constructing the ontology. Part-of-speech (POS) tags usually carry some linguistic information on terms, so POS tagging can be seen as a kind of preliminary method of classification to help construct concept nodes into the ontology. This is because features or attributes related to the concepts of different POS types may be different. In this study, a term POS tagging approach is proposed to simplify the classification of domain terms for the convenience of ontology construction.

1.4. Organization

This dissertation is organized as follows. **Chapter 2** gives an overview of related concepts and contains the literature review. **Chapter 3** attempts to build a domain ontology with the FCA method and discusses different kinds of attributes for given objects. **Chapter 4** investigates the algorithm for travelling and exploiting the domain

corpus from Wikipedia article pages. **Chapter 5** explores the different kinds of semantic information in Wikipedia pages and uses an SVM model to combine them together for mining concepts. Concept attributes are also extracted based on conctext information and the instance information of concepts to in Wikipedia article pages. **Chapter 6** proposes the algorithm for classifying concept terms by giving different POS tags to them. Finally, **Chapter 7** summarizes the whole work and discusses possible future directions for this study.

Chapter 2
Related Concepts and Literature Review

2.1. Related Concepts and Resources

The concepts and resources related to ontology construction and employed in this dissertation are introduced as follows:

1. **Concept:** In the psychological view, concepts are considered as mental representations. They are the constituents of propositional attitudes such as beliefs and desires. In the logical or semantic view, concepts are abstract objects, which are the constituents of the mental propositions. Considering both views, concepts are mental representations typed in terms of the senses they express (Margolis and Laurence 2007). In WordNet[1], a concept is defined as an abstract or general idea inferred or derived from specific instances. Terms label or designate concepts. Several partly or fully distinct concepts may share the same term (this is the definition in Wikipedia[2] of "concept"). One concept can also be expressed by several terms with similar meanings, referred to as the synset in WordNet. Concept learning refers to a learning task in which a human or machine learner is trained to classify objects by being shown a set of example objects along with their class labels (this is the definition in Wikipedia of "concept learning").

[1] http://wordnet.princeton.edu/perl/webwn
[2] http://en.wikipedia.org

Bruner et al. (1967) defined concept attainment (or concept learning) as "the search for and listing of attributes that can be used to distinguish exemplars from non exemplars of various categories". In this dissertation, these definitions lead to the corresponding task of domain concept acquisition.

2. **Instance:** Instance has several synsets in WordNet. In the area of ontology construction, an instance should be considered an item of information that is typical of a class or group and the synset taken to refer to an "example, illustration, instance, representative" in WordNet. An instance is usually called an object in computer science. Instances usually share attributes of their corresponding concepts and fill different values to these attributes. It should be pointed out that instances of concepts are normally not considered to be part of an ontology. If they are appended in an ontology, they should appear only as leaves by being attached to corresponding concepts.

3. **Terms**[3]**:** Terms are words and compound words that are used in specific contexts. They are considered to be the labels or designations of concepts that are particular to one or more subject fields or domains of human activity. Terms are the individuals of terminology, which have more ambiguity than concepts when building an ontology, because the same term can serve different roles in the same ontology, according to its semantic meaning.

4. **Core terms:** Core terms are the terms that can represent the most basic concepts

[3] http://en.wikipedia.org/wiki/Terminology

and form all the other terminology of their domain. A core term for a specific domain should possess the following four characteristics. It should be (Ji et al. 2007a):

1) Domain specific: It must be a terminology in the specified domain;
2) Independent: It must be a complete and independent word;
3) Combinable: It must have a strong ability to form other, different terminologies in the domain;
4) Minimal: It should be of a minimal term in length in the sense that it is not formed by other, shorter considered core terms. In general, the length of a core term should be short.

5. **Attributes:** In WordNet, an attribute is considered to be an abstraction belonging to or a characteristic of an entity, from which objects or individuals can be distinguished by attributes with different values. In an ontology, attributes are the properties, features, characteristics, or parameters that concepts can have or represent during their being or interactions with other concepts. Instances of concepts share the attributes of concepts by filling different values to these attributes. For example, the concept *company* has the attributes "company name", "location", "company type", "industry", and so on. "*Microsoft*" and "*Bank of China*" are two instances of "*company*". Both of them share these attributes and fill them with different values. The attribute values corresponding to different

instances of *company* are listed in Table 1.

Concept		*Company*	
Instances		*Microsoft*	*Bank of China*
Attributes and their values	*company name*	*Microsoft Corporation*	*Bank of China Limited*
	location	*Redmond, Washington in the United States*	*Beijing, People's Republic of China*
	company type	*public company*	*public company*
	industry	*computer software*	*banking*

Table 1 Instances of the same concept fill attributes with different values

6. **Relations:** Relation[4] is a logical or natural association between two or more things; relevance of one to another. In mathematics, a **binary relation on** a set A is a collection of ordered pairs of elements of A. In other words, it is a subset of the Cartesian product $A^2 = A \times A$. More generally, a **binary relation between** two sets A and B is a subset of $A \times B$. An **n-place relation**[5] is defined on a Cartesian product of n sets, and is represented by a set of ordered n-tuples. Usually, the n-place relation can be decomposed or transformed into several binary relations, so sometimes the relation in ontology construction refers to the binary relation.
7. **Ontology:** The definition of ontology originally comes from philosophy. Ontology (the "science of being") is a word, like metaphysics, that is used in many different senses (Heylighen 1995). It is sometimes considered to be identical to metaphysics, but it is more properly used in a more specific sense, as that part of metaphysics that specifies the most fundamental categories of

[4] From The American Heritage® Dictionary. http://www.mathacademy.com/pr/prime/index.asp
[5] From the PRIME mathematics encyclopedia. Copyright © 1997-2009, Math Academy Online™ / Platonic Realms™.

existence, the elementary substances or structures out of which the world is made. Ontology is thus the analysis of the most general and abstract concepts or distinctions that underlie every more specific description of any phenomenon in the world, for example, time, space, matter, process, cause and effect, and system.

It was John McCarthy who first recognized the overlap between work done in philosophical ontology and the activity of building the logical theories of AI systems. Already in 1980, McCarthy affirmed that builders of logic-based intelligent systems must first "list everything that exists, building an *ontology* of our world" (McCarthy 1980). Gruber (1995) defined an ontology as a formal specification of a conceptualization for knowledge sharing.

8. **Wikipedia:** Wikipedia is an international online project to create a free encyclopedia in multiple languages. Using Wikipedia software, thousands of volunteers have collaboratively and successfully edited articles. Within three years, the world's largest Open Content project contained more than 1,500,000 articles, more than any other encyclopedia. However, not enough research has been conducted on the project or on Wikis at all.[6] Most reflections on Wikipedia have been raised within the community of its contributors. Wikipedia is particularly interesting to those conducting research on cyber metrics because of the richness of semantic information and because its data is fully accessible. One

[6] See http://de.wikipedia.org/wiki/Wikipedia:Wikipedistik/Bibliographie (retrieved on April 10, 2005) on Wikipedia and http://www.public.iastate.edu/~CYBERSTACKS/WikiBib.htm (January 30, 2005) on Wikis.

can analyse a wide variety of topics, ranging from author collaboration to the structure and growth of information, with little effort at data collection and with a high degree of comparability (Cimiano et al. 2004).

Since Wikipedia is an open-content and collaboratively edited online encyclopedia, it has expanded from around 1 million articles (November 2006) to more than 3 million articles or more to date. Wikipedia is a rich resource of knowledge and semantic information, with hyperlinks to other entries and relevant classification information explicitly given by the contributors. Also the content of Wikipedia is oriented towards science and technology and science, making it a good candidate as resource for the construction of an ontology.

The whole structure of Wikipedia can be considered to consist of an interconnected network of articles. There are several types of article pages in Wiki, including *article* pages, *category* pages, *image* pages, and so on. Every *article* page in Wikipedia is referenced via a unique name and URL. For synonyms, *Redirects* pages that direct forward the users to another article can be created. For each *article page* there is a *Talk* page for discussions. Articles on the project itself belong to a special Wikipedia namespace. Uploaded *image* or other *media* files can be described at pages in the *media* namespace. Each logged-in user has a *User*-page where he can introduce himself. The *Talk* pages of user pages are essential for personal communication within the project. Furthermore, there is the *Template* namespace for text modules that can be

used in multiple articles, the *Category* namespace for categories that can be assigned to articles, and the *MediaWiki* and *Help* namespace for localization and documentation of the software. Figure 2 displays the distributions of different kinds of pages in Wiki:

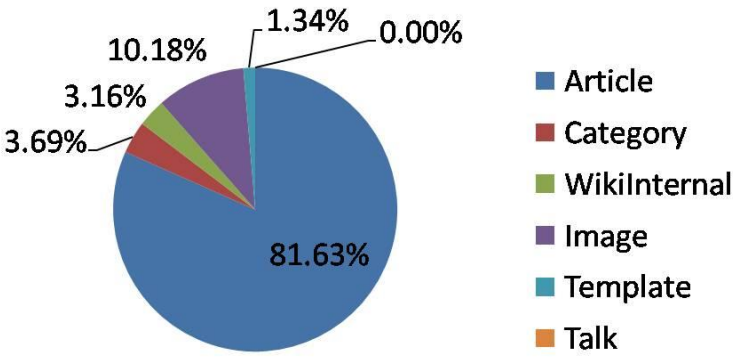

Figure 2 Distributions of Pages in Wikipedia

9. **WordNet:** WordNet is an online lexical reference system whose design is inspired by current psycholinguistic theories of human lexical memory (Miller et al. 1990). English nouns, verbs, adjectives, and adverbs are organized into synonym (equivalent word) sets, each representing one underlying lexical concept. Different relations link the synonym sets. As of 2005, the database contained about 150,000 words organized in over 115,000 synsets for a total of 203,000 word-sense pairs; in compressed form, it was about 12 megabytes in size.

WordNet mainly contains information from five kinds of words, namely: 1) Nouns: organized as topic hierarchies; 2) Verbs: with entailment relations; 3) Adjectives: in N-dimensional hyperspace; 4) Adverbs: in N-dimensional hyperspace; 5) Function words: probably stored separately as part of the syntactic components of

language.

WordNet distinguishes between nouns, verbs, adjectives, and adverbs because they follow different grammatical rules. Every synset contains a group of synonymous words or collocations (a collocation is a sequence of words that go together to form a specific meaning, such as "car pool"); different senses of a word are in different synsets. The meaning of the synsets is further clarified with short defining glosses. Most synsets are connected to other synsets via a number of semantic relations. For example, antonymy and synonymy are two common relations in WordNet. Semantic relations apply to all members of a synset because they share a meaning and are all mutually synonyms. Meanwhile, words can be connected to other words by lexical relations, including antonyms (opposites of each other) and derivationally related words.

WordNet also provides the polysemy count of a word: the number of synsets that contain the word. If a word participates in several synsets (i.e., has several senses), then typically some senses are much more common than others. WordNet quantifies this by the frequency score: in several sample texts all words were semantically tagged with the corresponding synset, and then how often a word appeared in a certain sense was counted.

10. **Formal Concept Analysis (FCA):** FCA is a formal method for data analysis and knowledge representation (Baker et al. 1998). It is a useful tool for automatically

acquiring taxonomies or concept hierarchies from texts.

FCA is composed of two sets of data. The first one is a so-called **object set** and the other is an **attribute set**. FCA can help to identify a binary relationship between the data of the two sets. The relationship is used to form a formal context according to a so-called formal concept lattice that satisfies the **partial ordering relationship**. The definitions of formal context and formal concept in FCA are given below according to Baker et al. (1998).

A **formal context** is a triple (G, M, I), where G is a set of *objects*, M is a set of *attributes*, and I is the relation on $G \times M$.

A **formal concept** of the context (G, M, I) is a pair (A, B), where $A \subseteq G$, $B \subseteq M$, $A' = B$ and $B' = A$, where $A' := \{m \in M \mid (g,m) \in I, \forall g \in A\}$ and $B' := \{g \in G \mid (g,m) \in I, \forall m \in B\}$. A is called the *extent* and B the *intent* of the formal concept.

A **partial ordering relationship**, denoted by $\leq$, for two formal concepts (A_1, B_1) and (A_2, B_2) , with regard to the inclusion of their extents or the inverse inclusion of their intents, formalized by: $(A_1, B_1) \leq (A_2, B_2) <=> A_1 \subseteq A_2$ *and* $B_2 \subseteq B_1$.

The whole formal concept lattice satisfies the partial ordering relations. Obviously, the FCA model can be used to represent ontology where the formal concepts in FCA correspond to the concept set ***C*** for a specific language ***L*** and a specific domain ***D***. The main issues in using FCA include: (1) the selection of an attribute set to describe the object set after the latter has been determined, (2) the

acquisition of the mapping between ***R*** and the partial ordering relationship in FCA. In the partial ordering relationship $(A_1, B_1) \leq (A_2, B_2)$, the two objects have a so-called type-subtype relation. A_1 is called the **subclass** of A_2 and A_2 is called the **superclass** of A_1. Object A_1 is the subclass of A_2 if and only if B_2 is included by B_1. If A_1 is the subclass of object A_2 and A_2 is also the subclass of A_1, then the relation between A_1 and A_2 is called an **equivalence relation**.

2.2. Literature Review

2.2.1. Ontology Construction with FCA Method

If the FCA method is to be employed to help build an ontology, two sets of information are needed. One is called the object set, which is a collection of given concept terms. The other is a collection of descriptive information for the objects, referred to as the attribute set. Different kinds of attributes have been investigated for fitting FCA into studies related to ontology, such as the design and construction of ontologies, the updating and merging of ontologies, and so forth. Objects (concept terms for the construction of ontologies) used in FCA can be in different granularities. Words, sentences, or even documents can be used as objects. Attributes, on the other hand, must be contextual elements that are most descriptive of these objects, such as words and phrases in the context of the objects. Relations between objects can then be shown by identifying the relationships between their attributes.

The work of Cimiano et al. (2003) was based on the assumption that verbs pose

strong selective restrictions on their arguments. By using a dependency parser, verb-object dependencies were extracted from the contexts, with the head words as objects and the verbs with the added postfix “able” as attributes for ontology construction with FCA. An example is the object named “apartment”, the attribute of which can be “rentable”, where the word “rent” is from the context of “apartment”.

A method for the semi-automatic construction and updating of ontologies with the help of FCA was proposed by Obitko et al. (2004). The objects used in this work are nouns, and only adjectives are selected as attributes. The method used by Obitko et al. generated the initial ontology structure using several objects and their attributes, such as “river” with the attribute “flowing”, and “lake” with the attribute “stagnant”. The structure was visualized using FCA. Potential objects such as “pond” were then added into the lattice and attributes as “natural”, “artificial” were also imported to extend the structure of the ontology until it was completed.

The work by Li et al. (2005) was the first to apply FCA to the construction of ontologies for Chinese. Li’s work focused on how to select sources of data and attributes for FCA to construct a domain-specific ontology. She proposed a framework to facilitate the manual modification of an ontology and used Information Gain (IG) to select sememes from HowNet as attributes. In a comparative experiment, a large-scale general corpus was used as the data source and only verbs were taken as the attributes used in the context of the tested object terms. This was based on the assumption that

verbs pose stronger selective restrictions on their arguments or that using verbs can avoid generating complex relations.

In the work of (Haav 2003), objects in FCA are actually domain-specific texts describing domain-specific entities. Take, for example, the following: “This apartment is in a family house in a quiet street, with a parking space, small bedroom, and big dining room, and is close to public transportation. Call to set up an appointment.” It is an advertisement in the real estate domain, and the attributes are confined to noun phrases only, such as “family house, parking space, and so on”. In fact, the constructed ontology shows the relationships between these descriptive sentences and phrases. There is no terminology being identified in this work.

The ontologies by Jiang et al. (2003) and Stumme and Maedche (2001) are constructed in a similar way as those of Haav. Jiang et al. developed an ontology construction system that integrates FCA with a natural language processing (NLP) module for medical documents, in which formal objects are medical documents and the compound medical phrases and concepts extracted from the NLP module serve as formal attributes. Stumme and Maedche described a bottom-up method to merge ontologies with the assistance of FCA. The formal objects of the FCA were documents, and the attributes were existing concepts in ontologies that had already been obtained. Basically, the FCA model was used by Jiang et al. and Stumme and Maedche for the classification of documents rather than for the acquisition of ontologies.

2.2.2. Wikipedia as a Resource

Wikipedia as a Web resource has been employed in many studies. Statistical studies and analyses have been carried out on the ratio between the number of edits and unique editors in Wikipedia articles (Lih 2004). Statistical studies on the structure and content of Wikipedia have also been conducted by Voss (2005). Bellomi and Bonato (2005) analysed Wikipedia's link structure and cultural biases, using two metrics, HITS and PAGERANK, to gain insights on the macro-structure of the organization of the corpus and on cultural biases related to specific topics.

Some researchers have also tried to add semantic relation links and attributes to Wikipedia (Völkel et al. 2006) and to measure semantic relatedness using Wikipedia as a resource (Strube and Ponzetto 2006). The work of Völkel et al. provided an extension to be integrated into Wikipedia , which allowed the inputting of links between articles and the specification of typed data inside the articles in an easy-to-use manner. Strube and Ponzetto presented methods of using Wikipedia for computing semantic relatedness, and compared this to the use of WordNet on various benchmarking datasets. The results showed that the performance from computing semantic relatedness from Wikipedia was better than a baseline given by Google counts.

Wikipedia was also exploited by Denoyer and Gallinari (2006) as the corpus in a content-oriented XML retrieval area. The corpus from Wikipedia used in the work of

Denoyer and Gallinari was mainly composed of eight collections in eight different languages: English, French, German, Dutch, Spanish, Chinese, Arabic, and Japanese. Only the article pages were kept and all the other kinds of Wikipedia pages such as "Talks", "Template", and others were removed. Besides these eight collections, additional collections were also provided for other IR/Machine Learning tasks such as categorization and clustering, machine translation, multimedia IR, entity search, and so forth.

Wikipedia categories were discussed in the work of Chernov et al. (2006) in connection with the task of extracting semantic relationships between the categories for building a semantic schema for Wiki. The aim is to improve Wiki's search capabilities and provide contributors with meaningful suggestions for editing Wikipedia pages.

2.2.3. Corpus Exploitation

The first step in building an ontology with the bottom-up method is to select an appropriate domain corpus as a resource for ontology construction. Some researchers prefer to use a corpus that has been manually constructed by linguistic experts, such as the British National Corpus (BNC) (Baker et al. 1998), a 100-million-word text corpus of written and spoken English drawn from a wide range of sources. BNC was compiled as a general corpus (text collection) in the field of corpus linguistics. Many studies on ontology construction make use of existing corpora derived automatically

or semi-automatically from books, magazines, and news organizations, such as the text document corpus of Reuters (Khan and Luo 2002). Reuters released a corpus of Reuters News stories from the year 2000 to 2004 (called the ***Reuters21578 corpus***) for research and development purposes, such as in the areas of natural language-processing, information-retrieval (IR), and information extraction (IE). The *Reuters21578 corpus*, which is a collection of newswire stories marked up in XML, originally collected and labeled by Carnegie Group, Inc. and Reuters, Ltd. has previously been seen as a standard real-world benchmarking corpus for the IR/IE and other communities. Attempts have also been made to collect a corpus for ontology construction by making use of search engines over the Web and selecting a corpus from the search results (Cimiano et al. 2004). Cimiano et al. developed a system based on the corpus collected from the internet, to try to implement a self-annotating Web. The PANKOW system proposed in this study is a pattern based method which starts from instances in given web pages to acquire corresponding concepts and then annotates the instance-concept pair information into text. As PANKOW is a pattern based method and starts from proper nouns as instances to concepts, it's not easy to get a corpus with considerable size and covering a whole domain with the PANKOW.

All of the abovementioned methods had shortcomings in one aspect or another. These included the cost in time and human resources for manual corpora, limitations in time and region for periodical and journal corpora, and the lack of appropriate

classification information for corpora from the internet.

2.2.4. Concept Mining

For ontology construction, terms as labels of concepts are usually extracted first. If terms are automatically extracted from a corpus, concepts and instances labelled by terms are usually mixed together. It is not easy to mine concepts in a considerable scale from a corpus without predefined semantic annotation. First, it is not easy to get exact labels of concepts from plain text. A bag-of-words approach may be used to express concepts. Gelfand et al. (1998) extracted concepts from unstructured text by identifying relationships between words and organizing them into a Semantic Relationship Graph (SRG) with the help of WordNet. In Gelfand's work, concepts are expressed by SRG structures of relevant words in documents. The work in their study proves that concepts can be equal to corresponding organized word groups; however, in their work, appropriate names were not assigned to these word groups.

Second, in the approach of general concept mining from the Web, it is usually necessary to make use of search engines to get relevant Web pages and then mine corresponding domain concepts. In their work, Liu et al. (2003) mined concepts and their definitions with the help of search engines. For a given term as a topic, its sub-topics or salient concepts are word phrases emphasized by specific html tags with high frequency. The acquired pages in these sub-topics were informative pages returned by some search engines containing definitions extracted according to a set of

rules. This method is topic oriented and cannot work for general domain concepts.

Third, syntactic patterns can also be used to get concepts through relations determined by these patterns. Sumida et al. (2006) extracted concept-instance relations in Japanese from simple noun sequences with the help of a full-text search engine to filter unqualified ones. In the method proposed by Sumida et al., noun sequences composed by one concept plus one or more instances are acquired from HTML documents. They are then filtered using rules and patterns, along with the result from a search engine. A similar work in English by Fleischman et al. (2003) was compared with this paper. However, in their method, Fleischman et al. relied on explicit clues such as punctuations and capitalizations, as well as on help from a morphological analyser for distinct proper nouns and other nouns. In both papers, concepts are tied with instances in simple noun sequences.

More efficient resources with annotated information, such as Wikipedia, have also been explored for mining concepts. Wikipedia is an information-rich resource with internal and external hyperlinks and relevant classifications annotated manually by contributors. This makes it a good corpus for information extraction and text mining. Gregorowicz and Kramer (2006) associated terms with concepts using the Disambiguation labels in Wikipedia pages and article titles to build a network of terms sharing the same lexical term as a substring in Article pages. This lexical term is then considered the representative for the concept of these title terms. For example, the

lexical term “Java” is used in many article titles. Thus, there is a Disambiguation page named “Java(disambiguation)” linking to “JavaScript”, “Java(band)”, “Java Platform”. Consequently, “Java” is considered a concept. However, the generated network and corresponding concepts have not been clearly evaluated in this work. In the former example, the term “Java” is in fact an instance of the concept “programming language” and it is not a qualified concept by itself. The annotated information denoting the relations between concepts and instances are still not well explored in Wiki.

Generally speaking, it is difficult to mine concepts from plain text and general Web pages because they are inefficient for supplying semantic information and suffer from the problem of data sparseness. A corpus containing structural information and annotated information, such as Wiki, is more efficient because it contains annotated information and plenty of instances. Currently, there is no clear way to make good use of different annotated information and to mine concepts from a given corpus. How, then, to make good use of different structural information contained in an instance-rich corpus to extract concepts is a practical question worth addressing.

2.2.5. Attribute Acquisition

The methods of acquiring attributes for concepts can be categorized into supervised and unsupervised methods with the help of many tools from different areas such as IE, classification, parsing, and so on. Almuhareb and Poesio (2004) compared the results

from clustering, depending on attributes and values (instances of attributes) to acquire concept vectors. Simple text patterns were applied to automatically extract both value-based and attribute-based descriptions of concepts into tables for concept clustering. The comparative result proved that even though there are fewer occurrences of attributes in a corpus than the instances (values) of these attributes, attributes are still better descriptors of concepts than attribute values. In addition, it was shown that the best attributes as descriptions to concepts are those having more general attribute values. Subsequently, Poesio and Almuhareb (2005) brought forward their classification of concept attributes. Their approach is based on which two supervised classifiers are trained to identify concept attributes from candidate attributes acquired from the Web. It uses morphological information, an attribute model, a question model, and an attributive-usage model to classify attributes. Up to 500 text pattern instances for each of the candidate attributes from the Web are used for finding attributes. The classification tool CLUTO[7] is used to cluster attributes using these vectorial representations built from collected data. Yoshinaga and Torisawa (2007) proposed an unsupervised method using patterns to acquire a set of attribute-value pairs (AVPs; e.g., director, W. Wyler) for a given object (e.g., "Ben-Hur") from semi-structured HTML documents. Class attributes used by many Web authors are acquired first through patterns, and then the acquired class attributes are exploited to induce patterns for extracting AVPs from Web pages.

[7] Karypis Lab, University of Minnesota. From http://glaros.dtc.umn.edu/gkhome/views/cluto.

The instance-driven extraction method was investigated by Pasca and Van Durme (2007) for extracting attributes from query logs. Their work is based on the theory of Dowty et al. (1980) that a class (concept) is traditionally a placeholder for a set of instances (objects) that share similar attributes (properties). Thus, the attributes of a given class can be derived by extracting and inspecting the attributes of individual instances from that class. In the work of Pasca and Van Durme, the target class is first specified as a set of instances. Then, for the given set of target classes, candidate attributes are selected, filtered for higher quality, and those that have passed all the filters are ranked.

2.2.6. Term POS Tagging

In general, existing studies on POS tagging can be classified into the following categories. The basic idea behind Term POS Tagging is to make use of information from the words themselves. A number of features based on prefixes and suffixes and spelling cues like capitalization have been adopted in studies on this subject (Mikheev 1997; Brants 2000; and Mikheev 1996). Mikheev (1996 and 1997) presented a technique for automatically acquiring rules to guess possible POS tags for unknown words using their starting and ending segments. Through an unsupervised process of rule acquisition, three complementary sets of word-guessing rules would be inducted from a general-purpose lexicon and a raw corpus which included prefix morphological rules, suffix morphological rules, and ending-guessing rules. Brants

(2000) used the linear interpolation of a fixed-length suffix model for handling words in his POS tagger, named TnT. For example, an English word ending in the suffix –able was very likely to be an adjective.

Some methods make further use of word morphology. They exploit more features besides morphology or take morphology as supplementary means (Samuelsson 1993; Toutanova et al. 2003; and Tseng et al. 2005). Samuelsson (1993) used n-grams of letter sequences ending and starting each word as word features. The main goal of using the Bayesian inference was to investigate the influence of various sources of information and the ways of combining them, on the ability to assign lexical categories to words. The Bayesian inference was used to find the tag assignment T with highest probability $P(T|M, S)$, given morphology M (word form) and syntactic context S (neighbouring tags). Toutanova et al. (2003) demonstrated the use of both preceding and following tag contexts via a dependency network representation. They made use of some additional features such as lexical features, including joint conditioning on multiple consecutive words and other fine-grained modelling of word features. Tseng et al. (2005) investigated a variety of morphological word features, such as the tag sequence features from both the left and right side of the current word for POS tagging, and implemented them in a Maximum Entropy Markov model.

Other researchers regard this POS tagging work as a multi-class classification problem. Many methods used in machine learning, such as the Decision Tree, Support

Vector Machines (SVM), and k-Nearest-Neighbors (k-NN), have been used for guessing the possible POS tags of words (Sun et al. 1997; Orphanos and Christodoulakis 1999; Nakagawa et al. 2001). Sun et al. (1997) presented a POS identification algorithm based on the k-nearest-neighbors (k-NN) strategy for Chinese word POS tagging. With auxiliary information such as the existing tagged lexicon, the algorithm can find out the k nearest words that are mostly similar with the word that needs tagging, to decide the tag of the targeted word. Orphanos and Christodoulakis presented a POS tagger for Modern Greek, which focused on a data-driven approach for the induction of decision trees as disambiguation or guessing devices. The system was based on a high-coverage lexicon and a tagged corpus capable of showing off the behaviour of all POS ambiguity schemes and characteristics of words. The main point of using the Support Vector Machine, which is a widely used (or effective) classification approach for solving two-class pattern recognition problems, is to select appropriate features and train effective classifiers. Nakagawa et al. (2001) made use of substrings and surrounding context as features for SVM and achieved a high degree of accuracy in POS tag predictions.

In fact, none of the above can exactly accomplish the task of term POS tagging because terms are very different in granularity from general words. However, all of these methods for tagging general words are valuable as references for tagging terms into different part of speeches.

Chapter 3
Build an Ontology using the FCA Method

An ontology is in fact the hierarchical organization of concepts. Formal Concept Analysis (FCA) is an effective tool for the acquisition of ontology structure and corresponding visualization can be made to see the structure directly. Some existing methods of using FCA to acquire the structure an ontology tend to use a single type of attribute such as verbs or nouns or adjectives in the context of a given concept term. As introduced in former chapter, for using the FCA method, an objective set as well as their attribute set need to be acquired in a head. In this chapter, terms as labels of concepts are given first as the object set and an appropriate attribute set need to be identified before using the FCA method. In Section 3.1 of this chapter, general acquisition method of context words as attribute set has been given. In section 3.2, context words of given concept terms are investigated, including nouns, verbs, adjectives, and adverbs, as well as their combination to determine which kinds of attributes are most appropriate for given concept terms. A core term list with statistical information and acquired seperately is also applied to compare with the context information. Section 3.3 gives experiments and discussions in general domain and IT domain for comparison. Section 3.4 summarizes the whole chapter on selecting appropriate attributes for constructing domain ontology with the FCA method.

3.1. General Method for Acquiring Attributes from Context

It is obvious that when concept terms are considered as objects in the FCA model, the attributes should be acquired mainly from the context of these terms, which has been described in Chapter 2. For example, to identify the attributes describing the concept term “硬盘”(hard disk), it is natural to look at the sentences where the term occurs and to look for content words in the context to serve as attributes. For example, suppose the term “硬盘”(hard disk) appears in the following two example sentences:

1. “在分割硬盘时允许移动硬盘分区表.” (When dividing a hard disk, the partition table of the hard disk is allowed to be moved.), and
2. “可大大提高硬盘存储照片的数量.” (That can greatly increase the number of photos that can be stored in a hard disk.).

It is natural to take certain context words such as “分区”(partition), “存储”(store), and others as the attributes. It is easy to see that content words play more important semantic roles in any specific context. Thus, in principle, nouns and verbs are the best candidates. However, adjectives and adverbs can also be considered as attributes because they directly modify nouns and verbs, respectively, and further qualify them in certain aspects. This work investigates the selection of content words including nouns, verbs, adjectives, and adverbs within the context windows of object terms as attribute candidates.

The selection of attributes should take into account different factors, such as the

distance from an object concept term, the frequency of co-occurrence with the concept term as object, and so forth. Because the selection of attributes is for ontology construction, in this work the criterion for selection is based on statistics that are similar to those of previous works, where the content words of selected types are examined based on their co-occurrences with the domain-specific terms. Only words with a certain level of statistical significance in co-occurrences are considered to be descriptive attributes. A context window in the range of [-5, 5] is used for any term object to obtain a list of word bigram co-occurrences, as follows.

For each concept term object t_i in the object term set T, a triplet $<t_i, w_j, n_{ij}>$ is used to represent the statistics of co-occurrence between an object term t_i and an attribute word w_j in the context of t_i. n_{ij} indicates the frequency of their co-occurrence. After the statistics are collected, all of the triplets are sorted in a descending order according to n_{ij}. The system has a threshold parameter N. For all $<t_i, w_j, n_{ij}>$ with $n_{ij} \geq N$, w_j is considered a significant descriptive word in the context of t_i, and thus is a member of the selected attribute set.

The connection between a concept term t_i and an attribute word w_j is then represented by a binary membership variable pair, $\mu(t_i, w_j)$. If there exists a triplet $<t_i, w_j, n_{ij}>$ with its $n_{ij} > N$, $\mu(t_i, w_j)$ is set to 1; or 0, otherwise. Then, a concept lattice can be built based on the $\mu(t_i, w_j)$ for all t_i in T to form partial ordering relations based on the FCA model, which was first introduced in the work of Cimiano et al. (2003). Each

relation between concept terms is represented by a $<t_i, t_j>$ tuple. Then, a $<t_i, t_j>$ tuple list is generated containing all of the terms in the term set *T*. $<t_i, t_j>$ is an ordered pair of terms according to the definition of a partial ordering relationship, where t_i is the subclass of t_j. If both $<t_i, t_j>$ and $<t_j, t_i>$ are in the list, this means that there is a relationship of equivalence between t_i and t_j.

In this study, the ConExp (Concept Explorer) system,[8] a Java API of open source software, is used to transform the concept lattice into a visualized FCA diagram. ConExp helps us to visualize the taxonomy of the ontology that is generated. If the terms in *T* do not appear in the corpus, they will not be shown in the FCA diagram. Since those terms have no statistically significant context words, they are considered non-attribute terms and are shown in the FCA diagram only as isolated nodes.

3.2. Attribute Selection for the FCA Method

In this section, the aim is to take terms acquired from some terminology extraction systems, and then investigate the content words including nouns, verbs, adjectives, adverbs as well as their combination in a corpus to identify the relationships among these terms. A so-called Core Term List, which is acquired separately, is also compared with context words to determine whether it qualifies as an attribute set for concept terms. The Core Term List contains terms that comply with some specific standards as qualified domain terms. The objective is to build an ontology for

[8] Copyright (c) 2000-2006, Serhiy Yevtushenko and contributors, available at http://sourceforge.net/projects/conexp.

domain-specific terms that have been acquired. The method is make use of the FCA model and determine what are the more appropriate context words to serve as attributes for ontology construction.

The selection of context words as attributes is based on statistical information about the context words. The acquisition of the context words and their statistical information is a complex process. However, there are certain words that are known to be domain specific and can be used directly rather than going through the complex selection process. For example, in the sentence “蓝牙技术是一种无线通讯协议”(Bluetooth is a wireless communication protocol), if the words “无线”(wireless), “通讯” (communication) and “协议(protocol) are known IT terms, it is natural to use them as attributes to describe the term “蓝牙”(bluetooth). Therefore, if there is already a qualified lexicon, they can be used directly as descriptive attributes to construct an ontology using FCA, as long as they are used in the context of the term object.

In this study, a so-called Core Term List is used as a separate attribute set for the ontology. The Core Term List is taken directly from a core lexicon, itself obtained from a separate research work of Ji et al. (2007a). According to the definition in Ji et al.’s work, a core lexicon in a specific domain should contain the most fundamental terms used in that domain. Each entry in a core lexicon should be frequently used in its domain. Furthermore, they must have strong descriptive power for other concept terms in the same domain, which means that other concept terms can be built based on

the items in the core lexicon. For example, in the IT domain, 数据(data) and 系统(system) should be core lexicon items because they can be used to form many other IT concept terms such as 数据库(data base), 操作系统(operating system), and others. In this work, the Core Term List contains 2,471 term entries taken directly from the IT domain core lexicon in the work of Ji et al. (2007a).

The way the items in the core lexicon are used is very similar to the method given in using context words. For a given concept term as object, if a core lexicon term appears in its context window of [-5, 5], it qualifies as an attribute for constructing an ontology with FCA only if its frequency is higher than the current given threshold *N*.

3.3. Experiments and Evaluation

3.3.1. Data Set and Experiments

Two different Chinese corpora are used in this chapter for attribute selection and further evaluation. The first corpus, referred to as the *General Corpus,* is a 1G corpus of the *People's Daily* from the years 1993 to 1998, segmented and tagged by researchers from Beijing University. The second corpus, referred to as the *PCW Corpus*, is a 52M corpus from the Chinese magazine *PC World* from the years 1990 to 1996, also segmented and tagged. For comparison to previous works, the object concept term set *T* here is the same as that used in the work of Li et al. (2005), which contains 49 IT terms as listed in Appendix C. The Core Term List contains 2,471 manually verified Chinese core terms of the IT domain.

The attribute selection algorithm given in 5.1.1 is applied on both sets of data when constructing an ontology using the FCA approach. For each corpus, four separate experiments are conducted to consider words for each type of POS (nouns, verbs, adjectives, and adverbs) as descriptive attributes. There is another experiment to investigate the combination of different types of POS. Experiments to determine whether the Core Term List is a suitable attribute set are also conducted for both corpora. For those sets of context words of terms, the ones with enough frequencies will be considered to be qualified attributes.

3.3.2. Evaluation Method

As the object term set *T* is fixed, the main task of evaluation is to determine the correctness of the automatically generated partial relationships between these concept terms as objects. Before an evaluation is conducted, a subjective answer set is prepared. Two IT domain experts are first asked to separately identify the partial ordering relationships for all of the terms in *T*. The results will then be consolidated to produce the answer set agreed upon by both experts after a review of the instances in which they differ. The final answer set contains 146 partial ordering term pairs from *T*. For example, for the two words "软硬件(hardware-software)" and "计算机(computer)", some people may consider that "软硬件(hardware-software)" is a subclass of "计算机(computer)". Others may think that "计算机(computer)" is a piece of hardware; thus, it is a subclass of "软硬件(hardware-software)". In this study,

the word “计算机(computer)” is considered in the general sense. Thus, the answer set takes the pair <“软硬件(hardware-software)” and “计算机(computer)”> as a partial ordering, where “计算机(computer)” is a super class of “软硬件(hardware-software)”, but not the other way around.

Then, for each t_i, in T, the corresponding concept lattice will be generated based on the selected attribute w_j, where $<t_i, w_j, n_{ij}>$ is a qualified entry according to the experiment part. Once the lattice is generated, it can then be applied through the FCA model to generate the partial ordering relationships. The evaluation is then conducted for each partial ordering relationship $<t_i, t_j>$ in FCA if $<t_i, t_j>$ is also in the manually prepared answer set for the calculation of *precision* and *coverage*. Here, the term *coverage* is used rather than *recall* because the set of concept terms under investigation is given as object term set T and other terms in the corpora are not evaluated. The performance of the extracted attributes is calculated through the formula (4) for *f-measure*:

$$f-measure = \frac{2*precision*coverage}{precision+coverage} \times 100\% \qquad (\mathbf{1})$$

For the purpose of reducing the influence of the corpus coverage that is used in the experiments, we do not calculate the relationships generated by those object concept terms that do not occur in the corpus as a resource for evaluation. For example, “CPU”, “ASCII”, and “单板机 (Single Board Computer)” did not appear in the PCW Corpus; thus, they are not considered in the evaluation of the PCW Corpus.

3.3.3. Experimental Results and Analysis

1. Results of Different Attribute Sets

Figure 3 shows the performance of six attribute sets on the General Corpus based on different threshold values. The six attribute sets are: (1) noun only, (2) verb only, (3) adjective only, (4) adverb only, (5) content word as combination, and (6) Core Term List only.

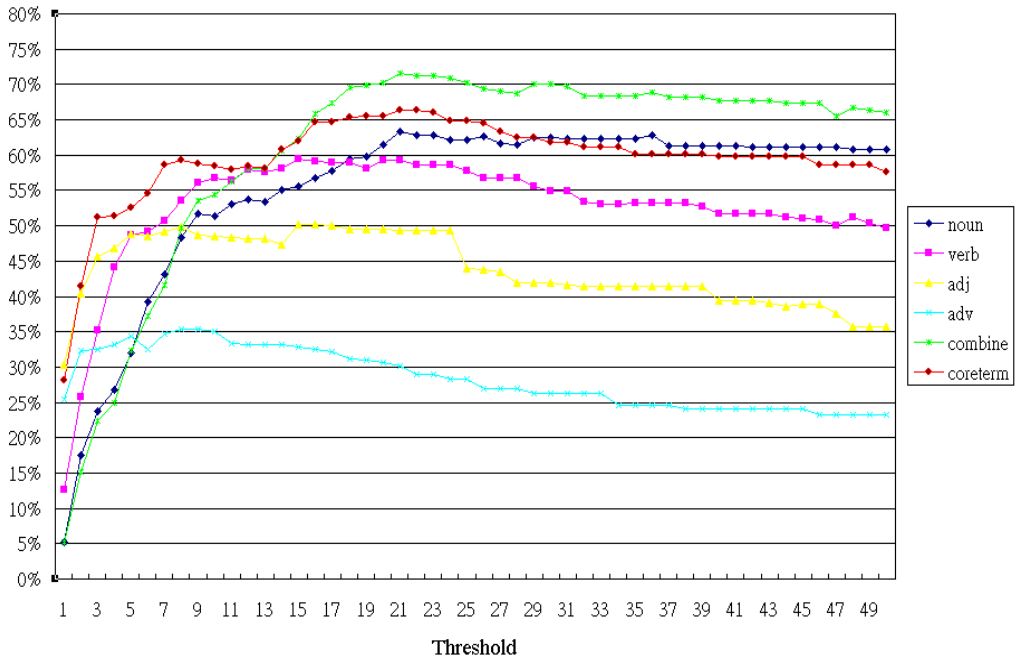

Figure 3 *F-measure* values according to threshold *N* on the General Corpus

It can be seen from Figure 3 that the highest peak point is reached by the combined content word attribute set with 71.67% in the *f-measure*. The Core Term List has the second highest peak at 66.35%, yet its performance declines to below that of nouns once the threshold reaches 30. It is interesting to note that the single noun attribute performs better than single verb attributes when *N* reaches about 18. This implies that verbs perform better than nouns with smaller threshold values. It is not

difficult to understand that adjectives and adverbs are the two worst performers with *f-measures* of 50.18% and 35.35%, respectively. From a semantic viewpoint, adjectives and adverbs are not directly associated with concepts as they are auxiliary words to nouns and verbs. So, in practice, their performances are very limited and they are not likely to be used alone.

Even though the combined content words performed best, their performance was the slowest to pick up. At low threshold values, they perform even worse than adverbs, which indicates that they require high threshold values for their power of discrimination to pick up. In terms of the speed at which the power of discrimination picks up, the Core Term list is the best performer. This is easier to explain. The Core Terms had already been obtained, and therefore do not need much "training" to work well. Verbs also have a relatively fast pick-up rate. This may explain why many works choose verbs as attributes. But, contrary to common belief, nouns have a better overall performance than verbs, especially those with higher threshold values.

Considering all of the single attribute sets, nouns have a better average performance. This indicates that nouns are more discriminative with high occurrences. It is not difficult to understand that nouns are better attributes, as they are directly associated with concepts. In the General Corpus, when the threshold *N* reaches 15 and 21, the experiments using nouns and verbs as the selected attribute set get the highest *f-measure* performance, respectively. The performances of points 15 and 21 are

compared based on more detailed observations. These observations show that the words that are in the verb attribute set and have a high frequency of co-occurrence are mostly general-purpose verbs, such as “用(use)”, “有(have/has)”, and so on. The more domain-specific verbs such as “识别(identify)”, “查询(query)” have a relatively low frequency of occurrence and thus are less likely to be used as attributes especially when the threshold moves up. However, the nouns serving as attributes with high frequency are more domain specific, such as “系统(system)”, “信息(information)”, “数据(data)”, and so forth. This leads to a situation in which there is a more obvious decline in performance when verbs rather than nouns are used as attributes.

The experiment using the combined attribute set produced the best performance, although the performance of the combined attribute set peaked much more slowly than that of other attribute sets. This is not surprising, as the combined attribute set basically takes the highest co-occurrence without considering POS tags. Experiments show that if frequency is the only consideration, all types of content words can in fact serve as attributes. A combination of different types of context words enhances the descriptive and discriminating power of the selected attribute set.

The combinations of adjective-noun and adverb-verb pairs are also considered and evaluated. But the performances of these two combinations are no better than those using nouns and verbs as single attributes. Thus, we did not show the results of the performances here, and they are not included in subsequent analysis. A poorer

performance is most likely due to the problem of sparseness of data.

The IT domain Core Term list performs well in the General Corpus, and its performance is better than that of the noun attributes. This is because the Core Term list contains the most fundamental and most frequently used terms in the IT domain. Thus, the Core Term list has no conflict with the data in the IT domain. In fact, many core terms are frequently used in the General Corpus, such as the domain-specific concept term “图像处理(image processing)”, which is concatenated by the two core terms “图像(image)” and “处理(processing)”.

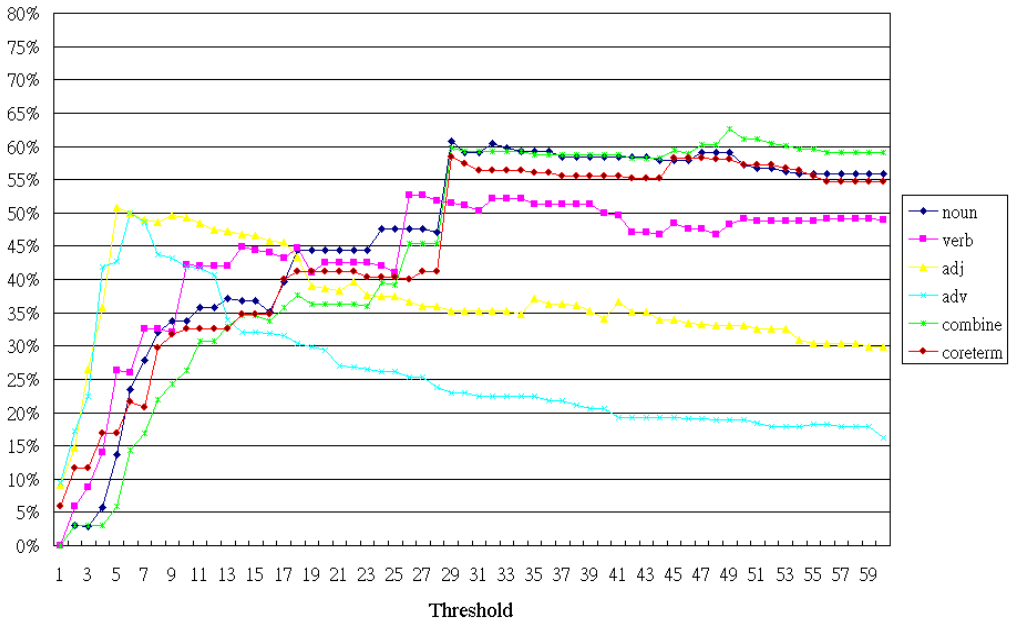

Figure 4 *F-measure* values according to threshold *N* on the PCW Corpus

Figure 4 shows the results of the performance on the PCW Corpus using the same six attribute sets. In fact, the performance trends for all six attribute sets are similar to those for the General Corpus, with two obvious exceptions. First, the pick-up trend is much less stable for the PCW Corpus than for the General Corpus, as shown in Figure

4. Second, the pick-up speed is much slower for the former than for the latter. That is why this experiment has a longer range of *N* from 1 to 120 at intervals of 2 as compared to the range 50 at intervals of 1 in Figure 3.

As shown in Figure 4, the highest peak point is also reached by the combined attribute set, with an *f-measure* value of 62.61%. Next is the noun attribute set with 60.71%. The experiment considering only verbs attains just 52.63% as its peak point, followed by adjectives with 50.75% and the adverbs with 50%. For both corpora, the various attribute sets reach their peak points in almost the same order. Only the curve of the Core Term list trails the curve of the noun attribute slightly because the Core Term list is more relevant in the IT domain corpus.

The reasons for a slower curve pick-up and a less stable increase in the PCW Corpus compared to the General Corpus can be explained by more detailed observations. The number of words that can be used as attributes in the PCW Corpus is much larger than that of the General Corpus. The average number of attributes one object can have in the General Corpus is about 63, while in the PCW Corpus it is about 319, which means that there are more relevant context words under the same threshold values in the PCW Corpus than in the General Corpus. On the other hand, this also means that there are more potential noises in the PCW Corpus as far as the FCA model is concerned, which makes the performance in the PCW Corpus less stable. For example, when the threshold is set as 1, the number of average combined attributes

from the General Corpus for one object is 500, while from the PCW corpus it is 2,000. In fact, this also explains why it takes higher threshold values in the PCW Corpus than in the General Corpus to weed out the attributes that are making more noise than contributing to performance.

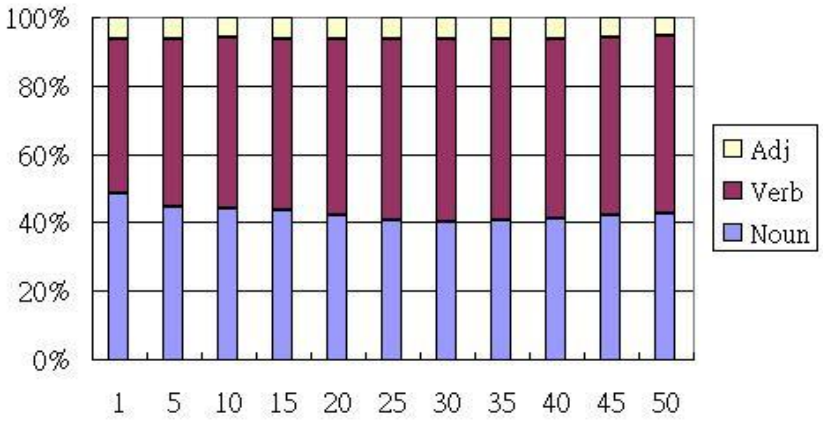

Figure 5 Distribution of different word types on the General Corpus

For more detailed analyses, Figure 5 and Figure 6 show the distribution of the different types of words under different threshold values on the General Corpus and the PCW Corpus, respectively. The distribution in Figure 5 indicates that the percentages of different word types are quite stable, with about 52% as verbs, 42% as nouns, and 6% as adjectives. This in a way justifies the intuition that verbs are commonly selected as attributes. However, 52% is only slightly over the 50% mark, which clearly indicates that using verbs alone is not enough. By looking at the memberships of actual words served as attributes in different threshold values, it becomes clear that different word types behave quite differently. The number of noun attributes is reduced from 4,955 to 82 when the threshold *N* is changed from 1 to 50,

which is similar to that of verb attributes, which goes from 3,683 to 87. However, before the highest point of threshold *N*, most of the domain-specific nouns such as “芯片(chip)” and “软件(software)” are still in the attribute set. On the other hand, most of the verbs that are more relevant to IT domain, such as “读取(read)”, “存储(store)”, and “开发(develop)”, have disappeared. Adjectives as attributes are in the same state as verbs. This explains why the *f-measure* value of nouns remains steady when *N* increases, but the *f-measure* values of both verbs and adjectives trend downward with an increased *N* value.

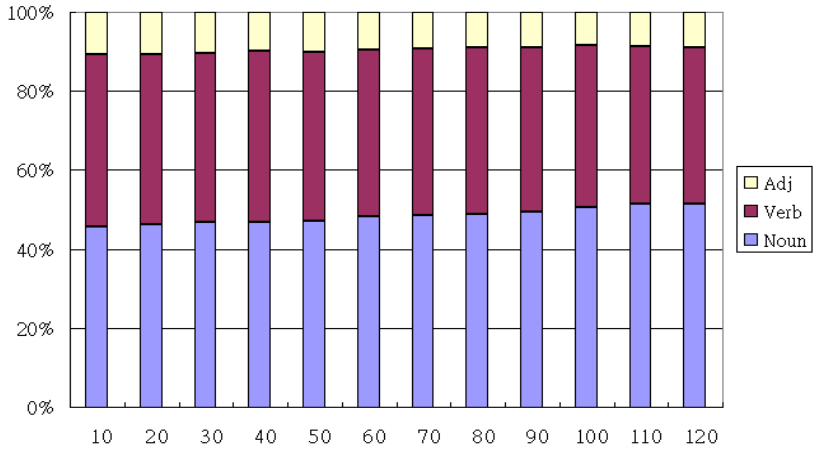

Figure 6 Distribution of different word types on the PCW Corpus

Figure 6 for the PCW Corpus as domain-specific data shows a different trend. Not only is the percentage of nouns larger, but a trend of increase in nouns as attributes can be seen from 45% to more than 51%. The percentage of verbs dropped from 43% to 39%, and that of adjectives from 10% to 8%. This is a good indication that in a domain-specific corpus, nouns play a more important role than other kinds of content words, and that their importance cannot be overlooked.

2. Average Performances on Different Corpora

Comparing the *f-measure* values in Figure 3 and Figure 4, it can be pointed out that the performances observed on the IT corpus are poorer than on the General Corpus. This goes against the common belief that words in a domain-specific corpus should be of better quality and more discriminative than words in a general corpus in describing domain-specific terms. To explain this issue, a more detailed analysis on precision and coverage is needed. Table 2 and Table 3 show the average levels of precision and coverage of the two corpora, respectively.

Corpus	Average Precision				
	Noun	Verb	Adjective	Combined	Core Term
General Corpus	55.47%	47.79%	33.25%	77.11%	56.76%
PCW Corpus	63.41%	51.99%	27.46%	75.01%	62.88%

Table 2 Average precisions of both corpora

Corpus	Average Coverage				
	Noun	Verb	Adjective	Combined	Core Term
General Corpus	60.50%	60.25%	67.01%	53.77%	65.54%
PCW Corpus	40.15%	45.79%	68.54%	36.23%	40.33%

Table 3 Average coverages of both corpora

Table 2 shows that the precisions of applying candidate attribute sets on the PCW Corpus are higher than or comparable to those of the General Corpus. But when considering the coverage to the standard answer, the General Corpus is better, as shown in Table 3. That means that the performance degradation of the PCW Corpus is

caused by lower coverage.

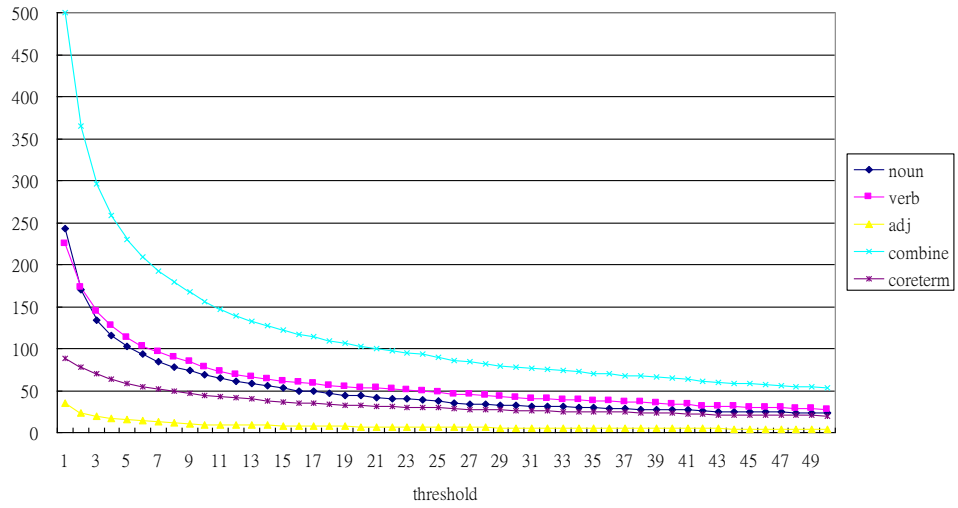

Figure 7 Attribute/object ratios on the General Corpus

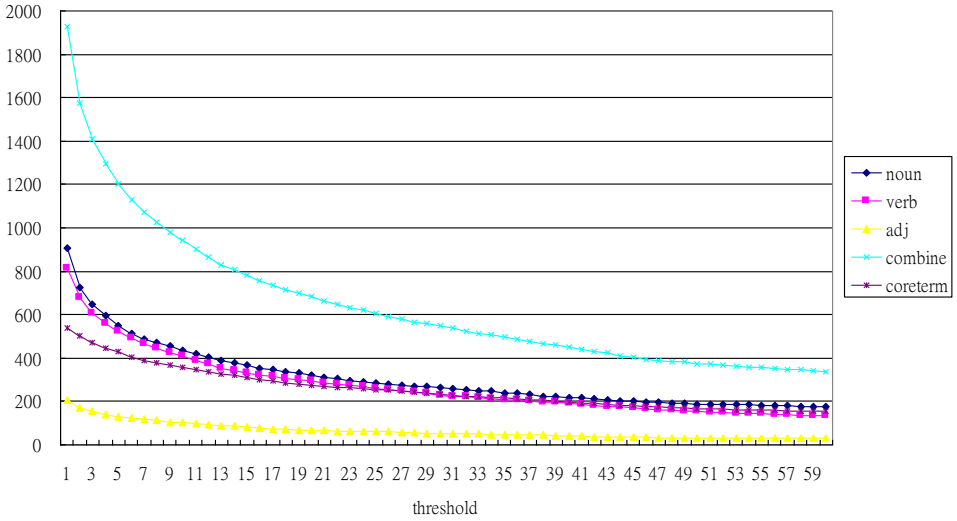

Figure 8 Attribute/object ratios on the PCW Corpus

To further examine the attributes used for identifying relationships between concept terms, a more detailed statistical analysis was done on the two corpora using a ratio between attributes and object, referred to as R_{AtoO}. This ratio is defined by the total number of attributes divided by the total number of term objects. R_{AtoO} shows on

average the number of attributes used to describe each concept term as an object. The statistical data on R_{AtoO} for the five different attribute sets applied on the General Corpus and the PCW Corpus are shown in Figure 7 and Figure 8, respectively.

Even though Figure 7 and Figure 8 exhibit similar trends for the five attributes sets that are being examined, the scale of the figure for the PCW Corpus is much bigger. Taking the curves of the combined attribute set in Figure 7 and Figure 8 as an example, the ratios are descending on both corpora. However, the maximal ratio on the General Corpus is up to 499 when the threshold value is 1. By comparison, on the PCW Corpus, the maximal ratio is up to 1,930, which is about four times that in the General Corpus. The best performances on both corpora are achieved when the ratio becomes less than 100. This means that when there are too many attributes, the performance is lower. This is consistent for all of the other attribute sets as well. Table 4 gives a summary of R_{AtoO} for both corpora.

Corpus	Average Attributes/object Ratio(R_{AtoO})				
	Noun	Verb	Adj	Combined	Core Term
General Corpus	52	60	8	119	34
PCW Corpus	308	275	65	648	257
R_{AtoO}-*PCW*/R_{AtoO}-*General*	5.9	4.6	8.1	5.4	7.6

Table 4 Average attributes to object ratios of both corpora

It can be seen from Table 4 that on average, every concept term as object has more attributes to describe it in the PCW Corpus than in the General Corpus. This is, in a way, consistent with the common understanding that there is more context

information in a domain-specific corpus than in a general corpus. However, the more attributes each concept term object has, the more likely it is that the attributes of different concept term as objects will intersect, rather than one being a subset of another, making it more difficult to satisfy the required partial ordering relationship. In other words, fewer partial ordering relationships can be identified using the FCA model in a domain-specific corpus than in a general corpus. This explains the relatively low coverage of relations between concept terms in the PCW Corpus compared to the General Corpus. For example, in the General Corpus, there is a partial ordering relationship from the term "硬盘(hard disk)" to "软硬件(hardware-software)" when using the combined attributes. But such a relationship cannot be identified in the PCW Corpus using the FCA analysis. This is because all of the combined attributes of "软硬件(hardware-software)" from the General Corpus are included in the attribute set of "硬盘(hard disk)". But when using the PCW Corpus, there are some attributes of "软硬件(hardware-software)" that do not occur in the attribute set of "硬盘(hard disk)", such as "平台(platform)" and "大型机(mainframe computer)", which leads to an overlapping of attribute sets instead of one attribute set including another.

This suggests that sometimes more attributes rather than fewer can make the construction of an ontology more difficult, as too many details can make the construction of a simple hierarchical structure difficult. Future work can be done to

further prune out smaller details if FCA is used. Other methods such as the imposition of further restrictions using syntactic chunks can also be explored.

3.4. Chapter Summary

This section builds a domain ontology with the FCA method. The domain specific terms are given as objects and for the attribute set, the context words are investigated, including nouns, verbs, adjectives, and adverbs, as well as their combination. Experiments on a general corpus and a domain-specific corpus show that the combined attributes of content word types give the best performance. Another experiment is conducted to examine context words that qualify as domain-specific terms from a so-called Core Term List, which is acquired separately. In fact, the Core Term List approach gives a result comparable to that of the combined content words. Thus, simple domain knowledge acquired apriori can serve as a good alternative attribute set, especially in cases of limited resources.

Besides, the results of experiments on the corpus from general domain and the IT domain demonstrate that the domain specific corpus does not supply more efficient attributes from the context for ontology construction with the FCA method. That is because the coverage of domain specific corpus is not satisfactory for building domain ontology. So a domain specific corpus is important for ontology construction and if an aumatic method for identifying domain specific corpus is proposed, the construction of domain ontology will have a solid base.

Chapter 4
Acquisition of Domain Corpora

As discussed in the summary of Chapter 3, when building an ontology with a bottom-up method, a domain specific corpus is usually required for acquiring concepts and building a corresponding hierarchy of domain. The domain specific corpus must have a good coverage of domain knowledge to generate a comprehensive ontology. Existing works have not done much to exploit the Web to identify domain specific corpus for ontology construction.

As the world's largest online source of encyclopedic knowledge, Wikipedia covers millions of articles and is still expanding continuously. Each article of Wikipedia is connected through hyperlinks in its main body to other Wikipedia entries. However, simply following these hyperlinks to find related articles in a domain is not appropriate because hyperlinks do not necessarily point to articles in the same domain. On the other hand, the category information for each article declared manually by contributors provides more relevant information on domain specificity in classifying article types. However, each article can belong to different categories. Suppose an IT domain corpus is the target of acquisition. As an example, the article page named "*Women, girls and information technology*" has the category labels "*Category:Women*", "*Category:Computing and society*", "*Category:Information Technology*", and others. Obviously, this article is labelled to be related to the IT

domain. Yet it is not a qualified IT domain-specific article.

In this chapter, a novel approach is proposed to trace domain-relevant articles in Wikipedia as a domain-specific corpus by making use of the category labels as classification information that is available in the article pages. The main idea is to generate a domain hierarchy from the hyperlinked category labels in Wikipedia pages. Only articles strongly linked to the given hierarchy are selected for inclusion in the domain corpus. The structure of Wikipedia is reorganized in Section 4.1. Section 4.2 introduces the proposed traversal approach, which makes use of linked category labels in Wikipedia pages to produce the hierarchy as a directed graph and obtain a set of pages in the same connected branch. Section 4.3 describes the ranking and filtering function of acquired nodes, which is based on the breadth first search algorithm for the classification tree. Section 4.4 supplies the experiments and corresponding discussions for selecting the scoring scheme and identifying the root nodes. Section 4.5 summarizes the whole chapter.

4.1. Structure of Wiki

Among the six basic types of Wikipedia pages (ordinary article pages, category pages, image pages, template pages, talk pages, and Wikipedia pages), only article pages and category pages are relevant in this work. According to the category labels declared in each article page, Wikipedia can be considered to be a directed graph where the articles are nodes and the category information are edges/links to other articles and

category nodes. In this study, basic terms used in graph theory are referenced here to define related objects in Wikipedia as follows:

First, all ordinary article pages and category pages are considered to be **nodes** in a graph and are named by their Web page titles.

A **directed edge** is defined by a 2-tuple **edge**(P_i, P_j), where P_i and P_j are two nodes and P_i contains the category information link to P_j. edge(P_i, P_j) is called the **out-edge** of P_i and also the **in-edge** of P_j.

A **Wiki-graph G**, is a directed graph defined by a 2-tuple, **G = <V, E>**, where **V** is a not empty set containing Wikipedia ordinary articles and category pages as nodes, called the **ode set**; E is a set of directed edges, called the **edge set**.

In a Wiki-graph, the **in-degree** of one node is defined as the number of in-edges of this node, and **out-degree** is defined as the number of out-edges of one node.

Generally speaking, a directed graph forms a network topology. According to the connectivity theory, if we start from a node P_r in the graph, all nodes connected to P_r can be reached following the directed edges. Generally speaking, given a Wiki-graph G, all of the traversed nodes of P_r form another graph G' with a set of V' and E' such that **G' = <V', E'>**, which is a connected branch of G.

A **Classification Tree T = <V', E"> ** with a selected root node P_r is defined as a spanning tree of G', G' = <G', E'>, where G' is a connect branch of G and E"$\subseteq$E', Pr $\in$V'. For $\forall P_i \in V'$ and $\forall P_j \in V'$, edge(P_i, P_j)$\in$E" if and only if edge(P_i, P_j) can be

reached along the in-edges starting from P_r.

For example, if the Wiki-graph is traversed from a given category node (say "Category:Information Technology" for the IT domain or "Category:Biology" for the biology domain) P_r, the article nodes that can be reached through category labels are in fact the terminal nodes and all of the traversed nodes form a tree-like structure. This structure can be considered to be a classification tree starting from a properly selected node. All of the edges connected to a node P_i are either out-edges from P_i to the category pages that P_i belongs to or in-edges pointing from some other category pages to P_i. $N_{out}(P_i)$ denotes the total number of out-edges from P_i and $N_{in}(P_i)$ denotes the total number of in-edges to P_i.

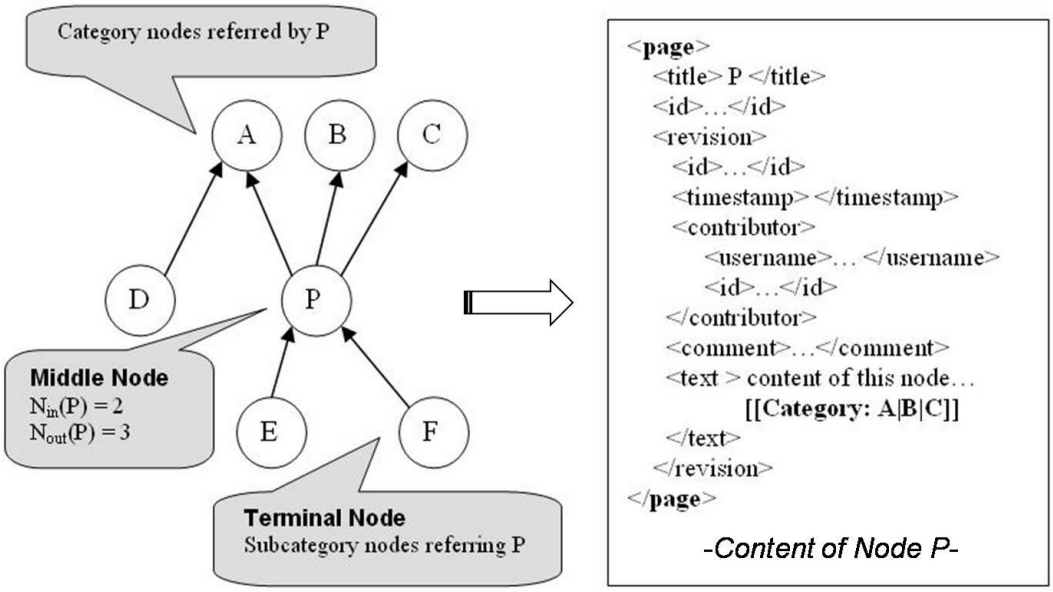

Figure 9 A Fragment of a Directed Category Graph from Wikipedia

A fragment of a Wiki-graph is displayed in Figure 9 with page P as the current node. As node P belongs to three categories A, B, C, and pages E, F belong to category

P, the number of out-edges $N_{out}(P)$ is three ($N_{out}(P) = 3$) and in-edges $N_{in}(P)$ is two ($N_{in}(P) = 2$). The right part of Figure 9 shows the content of a terminal node F. Different starting nodes will lead to different classification tree structures. If P is a terminal node, its $N_{in}(P)$ should be zero.

4.2. Classification Tree Traversal

In this section, the process of traversing the classification tree in fact involves growing a spanning tree of a connected branch in Wiki-Graph from a specified root node. Theoretically, both a depth-first search and a breadth-first search can be used for traversing a spanning tree. A breadth-first search is used in this algorithm because it traverses the tree one level at a time starting from the root. This makes it easy to show the relations between the visited nodes and the root node in a naturally hierarchical way.

```
Algorithm: CT-BFS
Input: Wiki-graph G and a root node P_r
Output: classification tree T with ranked nodes

For each node P_i in G {
    visited(P_i) = False;
} EndFor
visited(P_r) = True;
T.V = {P_r};
T.E = { };
W(P_r) = 1;
Push P_r into queue Q;
While (Q.empty( )==false) {
        P_c = Q.pop( );
        For each in-edge E_i of P_c {
            P_n is the other end node of E_i;
            If (visited(P_n)==false and P_n not in Q){
              T.E = T.E U {E_i};
               Push P_n into Q;
            }Endif;
        } EndFor
        W(P_c) = Scoring(P_c);
        Visited(P_c) = True;
        T.V = T.V U {P_c};
} EndWhile
Return tree T with ranked nodes.
```

Figure 10 Pseudo codes of the CT-BFS Algorithm

First, two hash tables are used to store all of the reverse pairs of pages with their categories, respectively. Then, the pseudo code shows the algorithm in Figure 10 to generate the classification tree of the Wiki-Graph using a breadth-first search (CT-BFS). As given in the algorithm, for a given root node P_r (how this node is selected will be discussed later), all other node pages in the same connected branch of P_r must be pointed to P_r either directly or transitively through the in-edges of P_r. Therefore, the CT-BFS algorithm starts from P_r to traverse the spanning tree for all of the nodes along the in-edges, one level at a time, to all reachable terminal nodes. No circle can form because no node will be visited twice. Using a breadth-first search, the

shortest route from the root node to each terminal node will always be selected if there are multiple routes between them. Each node is given a score after it has been visited according to the different scoring schemes to be discussed in Section 4.3. The scoring is based on a calculation of the node's relevance to the domain.

4.3. Ranking Nodes in the Classification Tree

During the traversal of the classification tree, each node is given a score on the relevance of the node to the specific domain. Once the traversal is completed, the terminal nodes, which are the article pages, are ranked according to the domain relevance scores. Pages over a certain threshold are considered domain relevant. The threshold value of ranking is an experiment-dependent parameter in the algorithm. The score can consider either in-edges or out-edges. Even though Wikipedia pages can belong to multiple categories, it is easy to see that the more out-edge nodes a node P_c has that are pointing to the classification tree, the more likely the node is to be domain specific. For a given P_c, suppose that it has a total of $N_{out}(P_c)$ number of out-edges. Among them, m out-pages point to the classification tree. P_i is the ith out-page of P_c in the classification tree, where i = 1, …, and m with P_i's score W_i obtained from the previous iteration of the CT-BFS algorithm. To initiate, $W_r = 1$ for root node P_r, and $W_i = 0$ if P_i is not on the traversal path to this level. Three scoring schemes are proposed with consideration of different proportions of N_{out} (P_c) and $N_{in}(P_i)$, where $N_{out}(P_c)$ means how many nodes are P_c's category nodes and $N_{in}(P_i)$ means how many nodes

take P_i as one of their category nodes. The numbers of in-edges and out-edges of each node in Wikipedia are independent of the classification tree, and they are acquired and stored in the two hash tables and are used during the traversal of the classification tree. The formulas for calculating the score W_c of P_c during a traversal are shown below.

S_1: $$W_c = \frac{1}{N_{out}(P_c)+1} \times \sum_{i=1}^{m} W_i \quad (2)$$

S_2: $$W_c = \sum_{i=1}^{m} (W_i \times \frac{1}{N_{in}(P_i)+1}) \quad (3)$$

S_3: $$W_c = \sum_{i=1}^{m} (W_i \times \frac{1}{(N_{in}(P_i)+1) \times (N_{out}(P_c)+1)}) \quad (4)$$

In the scoring scheme S_1, the W_c of P_c takes the sum of the scores of P_c's out-edges pointing to the classification tree against the total number of P_c's out-edges. In S_2, the score of P_c is a consideration of the sum of P_c's out-edges in the classification tree against the total number of the in-edges of P_is, which are P_c's upper-level nodes pointing to the classification tree. S_3 scores P_c according to the sum of the out-edge nodes in the classification tree divided by both the total number of its out-edges and the total numbers of the in-edges of those upper-level nodes in the classification tree. In summary, S_1 and S_2 consider the influence of out-edges and in-edges respectively, whereas S_3 combines both factors.

4.4. Experiments and Evaluation

The experiments are conducted on the English Wikipedia with a cut-off date of November 30th, 2006, which contains about 1.1 million article pages and category

pages. In order to ensure that the method can work on different domains, two specific domains are selected for the evaluation of the proposed algorithm and further ranking schemes. They are the domains of IT and biology. In the IT domain, the root node is set as “Category:Information Technology”. Using the CT-BFS traversal algorithm, the classification tree that was obtained can reach 549,486 nodes from the 1.1 million nodes. As for the biology domain, the root node is set as “Category:biology”, which reached 549,433 nodes when traversing the classification tree. Considering that Wikipedia has article pages in many other domains, it is obvious that there is a heavy overlapping of the two sets of traversed pages. Thus, a selection scheme must be applied to choose the most relevant pages of a single domain.

4.4.1. Scoring Scheme Selection

The scoring schemes are applied in separate experiments starting from the selected domain-specific root category nodes for the two domains. The results are then ranked, respectively. For evaluation, the ranked nodes are sampled at intervals of 1,000 for the first 20,000 nodes, then at intervals of 20,000 from 20,000 to 100,000. At each sampling point, 10 consecutive nodes are taken manually by two people knowledgeable in both IT and biology. Thus, for each domain, there are a total of 250 samples for evaluation. Precision is used as the measure of performance. Table 5 shows the sampling evaluation results on the IT domain and Table 6 shows the corresponding results for the biology domain. For each table, the first 20 columns

show the qualified domain nodes as intervals of 1,000 according to the top 20,000 nodes, and the last five columns are qualified results from every 20,000 nodes from the 20,001th to 100,000th nodes.

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25
S1	7	0	0	0	0	0	0	0	0	0	0	0	7	0	0	7	0	9	8	0	10	1	3	10	0
S2	10	10	10	10	9	7	9	10	10	10	10	10	1	10	10	0	0	9	8	7	0	10	0	0	0
S3	10	10	10	10	9	10	10	10	10	10	8	10	1	10	10	10	7	10	10	10	8	10	0	0	0

Table 5 Evaluation Result of Different Schemes in the IT Domain

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25
S1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	10	2	10
S2	10	10	10	8	10	6	7	6	7	4	10	10	0	8	10	9	0	5	9	5	10	0	0	0	0
S3	10	10	10	10	9	9	9	6	10	10	9	7	10	10	3	6	9	9	8	9	2	0	0	0	0

Table 6 Evaluation Result of Different Schemes in the Biology Domain

As shown in Table 5, for the results in the IT domain, the descending tendency is quite apparent in S_3 and S_2 according to the ranks, yet it is not so apparent in S_1. This means that the exclusive employment of out-edges (in S_1) is not sufficient. This tendency is even more apparent in Table 6 for the biology domain because among the sampling results from the top 20,000 nodes of S_1 in Table 6, there are no domain-relevant pages until after 60,000. In fact the results from both tables show that S_2, which considers the influence of in-edges to upper-level nodes, performs better than a consideration of the out-edges from the current node only. Furthermore, the results of S_3 have fewer fluctuations than those of S_2 during the descending tendency. This means that S_3 is most probably more appropriate for selecting domain-specific

nodes than S_2 because both the total number of in-edges and total number of out-edges are factored in. On the whole, the sampling results in Table 6 are in a very similar situation to those in Table 5, but there are a few differences between Table 5 and Table 6. Besides the difference in S_1, there are fewer biology-relevant items than items in the IT domain. After the top 60,000 nodes in the IT domain, not many nodes can really qualify as domain-specific articles; while in the biology domain, the boundary has been advanced to 40,000.

Generally speaking, a good scheme for reflecting domain specificity should show the ranking results in a descending order and should have good overall precision with regard to domain-relevant pages. The overall precisions of the different schemes applied on the top 20,000 nodes on both the IT domain and the biology domain are shown in Table 7.

Schemes	Precision in IT domain	Precision in Biology domain
S_1	19.0%	0.0%
S_2	76.5%	72.0%
S_3	92.5%	86.5%

Table 7 Overall Precisions of Different Schemes

The precisions of the different schemes show that S_3 is the best scheme for getting the pages that are most relevant to the selected domain. On the basis of this fact, experiments show that the text parts of the first 20,000 articles can be taken to form a domain corpus with good confidence, using S_3 for both corpora. For the IT domain, the generated corpus size is 98M; while for biology domain, the size of the generated

corpus is 101M. These are reasonable sizes for a domain corpus without any need for manual selection.

4.4.2. Root Node Identification

The selection of the root node is very important to the generation of the classification tree indicating a given domain. Whether the root node is a proper one can be verified by comparing the structure of the tree to an existing acknowledge classification system such as the LACC structure. Although the choice of root is not automatic, in practice, before you build an ontology, you need to first know which domain you are trying to build the ontology. The name of the domain, say IT, "Information Technology" would be suffice to start the traversal. However, two other nodes, "Category:Communication" and "Category:Electronics", can also be taken as the root node to be applied to the algorithms. It is interesting to note that almost the same number of nodes (549,486, 549,485, and 549,483) is reached by all three different starting nodes, although the classification trees are different and the ranking results are different. Here, S_3 is taken as the scheme, and the node "Information technology management" ranked 33 if the starting node is "Category:Information Technology". The same node will be ranked 425,335 if the starting node is "Category:Electronics", which in fact is not appropriate. What this experiment told us is that the choice of the root node does make a difference, as out of the 549,486 reachable nodes, only about 20,000 top-ranked pages are used. It is understood that the classification information

provided in Wikipedia is supplied by a contributors' manual, which does not have strict rules about following any reference classification. In fact, because the hyperlink structure for these classification links in Wikipedia is a network, it is likely that almost all of the domain-specific nodes can be reached if the traversal starts from any node. This does give rise to the need to further validate the appropriateness of the selected root node.

For this evaluation, the classification structure provided by the Library of American Congress Classification (LACC) is used as the external reference and assumed to be the correct classification. It should be noted that, as a classification for books, LACC has a rather flat structure. For the 32 IT-related categories in LACC, for example, the hierarchical structure is not complete and there are partial trees involved, as given in Appendix A for reference. The relations refer to the links defined in LACC, and there is a total of 26 such relation links for the IT category. The validation compares the classification tree produced by using S_3 in terms of (1) the coverage of the terms in the classification tree with respect to all of the domain trees in LACC and (2) the violation of the hierarchy with respect to that of LACC. The comparison is done by a manual check, so that abbreviations and plurals are considered.

Table 8 displays a summary of the results of the evaluation. Out of the 32 IT-relevant categories in LACC, 26 appear in Wikipedia as either a category node or article node. The S_3 algorithm identifies 21 of these 26 categories in its classification

tree. As for the categories in LACC that do not appear in Wikipedia (a total of six), this situation arises because the LACC category names are more general high-level names, which are not directly used by the contributors. However, the corresponding lower-level categories or names are in Wiki. For example, for the LACC category name “Internet domain names”, the category nodes “World Wide Web”, “Internet protocols”, and other more detailed terms are used instead in Wiki. Out of the 28 pairs of categories in LACC that hold hypernym relations, 23 appear in Wiki. Using S_3, 20 such relations are maintained in the same order in the acquired classification for the IT domain. The other three relations do not exist in the classification tree. For example, in LACC, “Usenet” is under the classification of “Networks”; on the other hand, the page titled “Usenet” in Wikipedia can link to “Networks” by following the category nodes “Wide area networks”, “Networks by scales”, “Computer networking”, and “Networks”. That means that the relation between “Usenet” and “Networks” from LACC and Wikipedia are consistent.

Root Node	IT		Biology	
	Terms	Relations	Terms	Relations
LACC	32	28	31	25
Wiki	26	23	21	17
Domain Classification Tree	21	20	20	15

Table 8 Comparisons of Domain Classification Trees with LACC

As to the biology domain, there are 31 relevant nodes in LACC. Among them, 21 occur in Wikipedia as article nodes and 20 are contained in the classification tree.

Although some items cannot be found in Wiki, such as “Cytology” and “Animal biochemistry”, their synonyms can be found in the classification tree from Wiki, called “cell biology” and “biochemistry”. Out of the 25 hypernym relations in LACC, 17 such relations are in Wiki, and 15 are maintained in the acquired classification tree. The other two cannot be found in the classification tree. For example, the item “Cyanobacteria” is below the classification branch of “Microbiology”, a major subfield of “Biology” in LACC. In the Wikipedia classification tree, the node “Cyanobacteria” can also be linked to “Biology” according to the sequence of category nodes “Bacteria”, “Prokaryotes”, “Microorganisms”, and “Microbiology”. Obviously, the relation between “Cyanobacteria” and “Microbiology” acquired from the biology domain classification tree of Wikipedia also conforms to that in LACC. Comparing the two domains of the experiment, both the node coverage and relation coverage of the biology domain are lower than those of the IT domain. That may be because biology is a comparatively traditional and stable domain, while IT, as a new area of science and technology, can involve more interdisciplinary and applied areas.

Root Node	Electronics			
	For Electronics		For IT	
	Terms	Relations	Terms	Relations
LACC	34	42	32	28
Wiki	30	36	26	23
Domain Classification Tree	23	30	14	2

Table 9 Comparisons of the Classification Tree with LACC with the Root Node:Electronics

A further examination was also conducted to compare the generated classification tree with “Category:Electronics” as the selected root node for the IT domain and the electronics domain in LACC, as shown in Table 9.

When compared with the electronics classification in LACC (see Appendix B for details), “Category:Electronics” is a good choice as the root node because most of the nodes as well as the relations can be found in LACC. However, when this classification tree is compared to the IT categories in LACC, it can be seen that only 14 nodes out of the 26 in Wikipedia are found. Furthermore, out of the 23 category relations, only 2 of them are maintained in the same way. In other words, this classification tree is much more screwed in comparison to the classification of the IT domain.

The comparisons with the LACC classification indicate that the choice of root node must be representative of the domain. Otherwise, the classification tree that is generated will not be representative of the domain, putting the quality of the acquired articles at risk. However, if one chooses a popularly used general term to represent the domain, the classification hierarchy of LACC is usually maintained by the generated classification tree, and thus the acquired corpus remains domain specific.

4.5. Chapter Summary

In this chapter, a novel approach was proposed to select an appropriate domain-specific corpus from Wikipedia for constructing an ontology. A classification

tree was acquired from traversing the category and article nodes in Wikipedia using the proposed CT-BFS algorithm. Three different schemes were evaluated by taking into consideration in-edges, out-edges, and their combinations. Two domains, the IT domain and the biology domain, were selected for the evaluation of the ranking results of the acquired nodes. The results of the evaluation show that the scheme that considers both types of edges gives the best performance and the corpus using this scheme is of good quality and can readily be used without the need for manual selection. This confirms that Wikipedia is a good Web resource as a domain-specific corpus if proper selection and scoring algorithms are applied. Also, the quality of the algorithm is dependent on the choice of the root node in the traversal. A general rule of thumb is to use the most representative and general term to name the domain, and the result should be quite reasonable.

Chapter 5
Concept Mining and Attribute Extraction

As concepts are the basic components for expressing and organizing the knowledge in an ontology, the main task of this chapter is to mine concepts for ontology construction. In this chapter, different kinds of annotated information in Wikipedia are explored to be candidates of features for applying the SVM model to mine concepts corresponding to given article page names as instances from Wikipedia as a corpus. The annotated information includes Wikipedia {{Infobox}} Structures, definition sentences, and Category Labels. In Section 5.1, different kinds of annotated information are introduced and individually employed to acquire concepts. A simple combination of these annotated data and a SVM model are applied to improvement the performance of concept mining. In Section 5.2, the context information and the instance information in the wikipedia article pages are employed to extract attributes for concepts. Section 5.3 summaries the work of the whole chapter.

5.1. Concept Mining from Wikipedia

Below, Figure 11 gives an example of a Wikipedia page containing an {{Infobox}} Structure, definition sentence, and Category Labels. The background is a real Wikipedia page named "*Atlas Shrugged*". The left part of the front square with blue borders displays the structure of a general article page, which is considered an instance

page. The contents of this page are given on the right side, which shows details of the information contained in a general article page.

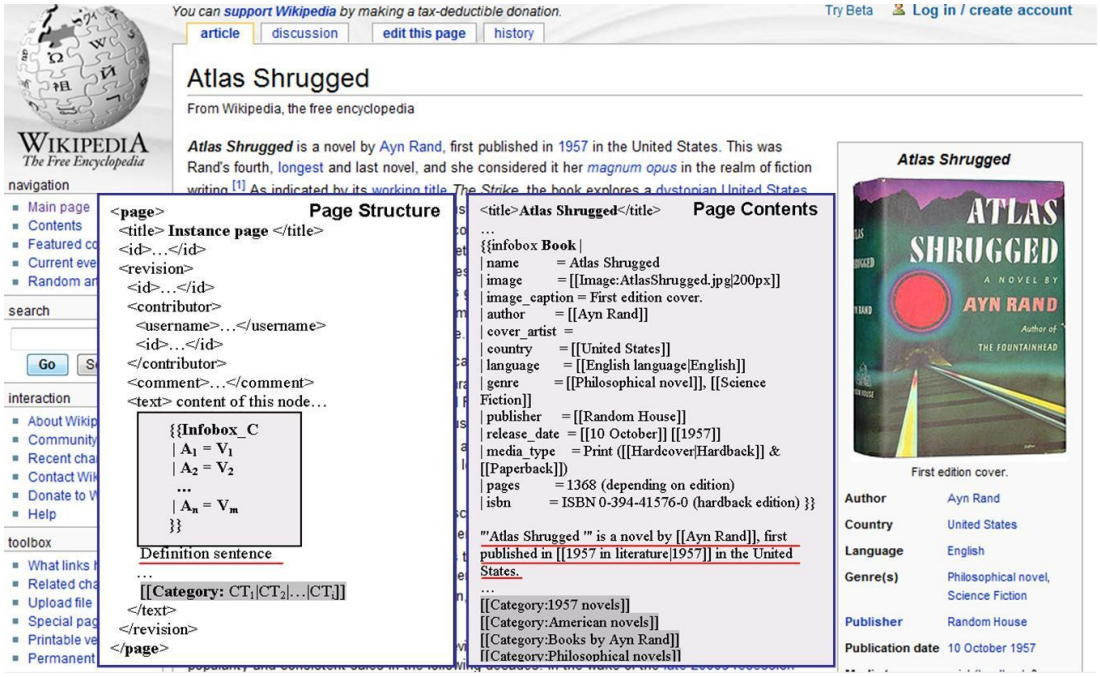

Figure 11 Different Resources Contained in a Wikipedia Page

Figure 11 indicates that the page entitled "*Atlas Shrugged*" is an instance page of the concept "*book",* because it contains an {{Infobox *book*}} Structure quoted by double levels of squares and starts from the key word Infobox. The first sentence underlined in red is the definition sentence, which points out that "*Atlas Shrugged*" is a "*novel*". The Category Labels highlighted in gray colour indicate the topics of this page as "*Category: 1957 novels*", "*Category: Books by Ayn Rand*", and others. Just as above, each article page can belong to one or more corresponding categories by including the [[Caterogy:Name]] labels into its text. These labels link different kinds of pages together as a hierarchy, which can be considered to be a directed graph,

referred to as a classification tree. In fact, this hierarchy of Wikipedia pages is organized by the topics of article pages and hyponym relations. Among these relations, the category labels that tie together article pages with their upper-level category pages indicate the concepts that are potential candidates for corresponding article pages as instances.

In a word, all of the annotated semantic data above, including structures and labels, which are contained in Wikipedia pages, more or less indicate the concept-instance relation. If all of the article pages are taken as instances, concepts can be identified according to the concept-instance relations indicated by these Wikipedia resources. The methodology for making use of different resources in Wikipedia will be introduced in the following subsections.

5.1.1. The {{Infobox}} Structures

In this chapter, the method that uses {{Infobox}} Structures will be employed as the baseline because of its high precision and low coverage, which is not a balanced performance. An {{Infobox}} Structure is a formatted table present in some article page *P* and labelled by a common subject $concept_p$ in the form of {{Infobox $concept_p$}} or {{Infobox_ $concept_p$}} to indicate that it is an instance page with reference to the $concept_p$ to which it belongs. Pages with the {{Infobox}} Structures pointing to the same concept are instances of the same concept. More than one {{Infobox}} Structure can be present in an article page. Thus, an instance can also be associated with

multiple concepts. For example, the article page "*Arnold Schwarzenegger*" is an instance of both {{Infobox *Actor*}} and {{Infobox *Governor*}}. An {{Infobox}} Structure can be extracted according to the XML and Wikipedia tags in Wikipedia pages. In all article pages, only 17% contain the {{Infobox}} Structure, which is a quite low level of coverage for all article pages as instances in Wiki.

5.1.2. Definition sentences

Because of the low coverage of {{Infobox}} Structures, definition sentences in Article pages are also considered in this work for concept extraction. Although there is no strict formatting rule in Wiki, the content of an Article page usually starts with a sentence defining the instance or concept that the article describes. Most of the time, the sentence will also provide the hypernym concept associated with the given instance or concept. For example, the definition sentence for the Article page "*Atlas Shrugged*" displayed in Figure 11 is "*''Atlas Shrugged'' is a novel by Russian-born writer and philosopher [[Ayn Rand]], first published in 1957 in the [[United States|USA]]*." This sentence states that "*Atlas Shrugged*" is an instance of the concept "*novel*".

Even though a definition sentence is the first sentence in the text part of an Article page, a number of pre-processing steps must be taken to remove all unnecessary information. The cleaning up includes the removal of (1) all html structures except needed tags such as <page>, <title>, and <text> relevant tags; (2)

other annotation tags such as italics and internal links; and (3) {{Infobox}} Structures and other structured contents quoted by {{ and }}.

In principle, the first full stop by the punctuation mark "." (period) should signify the end of the first sentence. But, in practice, a further analysis is needed to identify anomalies. For instance, abbreviations can be used with the period mark at the end such as "*Prof.*" and "*Mr.*". Other cases, such as "*Ph.D.*", "*No.1*", and "*U.S.*" also should not be treated as a full stop. So a pre-processing module to correctly identify the first full stop and ignore these abbreviations is necessary.

Concept identification is done by analysing the syntactic structures of the definition sentences to identify the main verbs first. Then, the noun or noun phrases following the main verbs are extracted as the concepts. In general, definition sentences can be classified into two types according to the main verbs used in the sentences. One includes all sentences using the be-verbs such as "is", "was", "are", and "were". The other is for sentences using non-be-verbs. Be-verbs normally indicate the *is-a* relationship. Thus, the nouns or noun phrases directly after the be-verb can be considered to be the corresponding concepts of the subjects. Non-be-verbs are of various different kinds and the relationship between the subject and the noun phrases after the verbs depends on the verbs that are used. Some may reflect the *is-a* relation; others may not. If the corresponding noun phrases of non-be-verbs are taken as the corresponding concepts, the precision can be affected, as will be shown in the

evaluation part.

Theoretically, a dependency parser is a good choice to analyse definition sentences no matter the main frame of the sentence is a “be-verb” structure or a “non-be-verb” structure. It is because the parser can get the objects of the main verb more precisely. However, in practical use, the large size of corpus and the run time of the parsers determined that it is more practical to use a POS tagger and patterns to mine object candidates for the main verb in the sentence. The relations indicated by “non-be-verb” structures can be classified according to a small set of result of dependency parser. The simple structure with direct objects can be process with the POS tagger and patterns. As the task of identifying concept just needs to look for the nouns or noun phrases directly following the verbs, the use of a full parser will be time consuming and unnecessary. Thus, a POS tagger which has much better precision is applied along with a regular expression to identify the noun/noun phrases after verbs. POS taggers usually tag general noun phrases as NN(S) and proper nouns as NNP(S). Our algorithm prefers to select NN(S) as concepts because the plural format of a general noun indicates this noun is more like a concept than an instance. Instances usually have no plural format in real use. Only when NN(S) is not present, NNP(S) will be taken as the concepts, because sometimes the noun/noun phrase indicating the relevant concept to the given Wikipedia article as instance can be labeled as NNP(S) by the POS tagger. For example, “Thad Cochran” is a “senator” of United States,

where the “senator” is tagged as NNP because its first “s” is capitalized in the phrase “United States Senator”. Also, in the cases that the extracted nouns or phrases following the verbs start with “type of”, “a kind of”, “name of”, and “one of”, the noun phrases after “of” will be extracted as the target.

5.1.3. Category Labels

Categories are contained in Wikipedia in two different ways. First, they appear in Article pages as Category Labels to indicate information about the topics in Article pages. There are also Category pages, with the title of a page being a Category Label, and the page listing all of the sub-categories under the current label. If all Wikipedia pages are organized as a hierarchy by Category pages, the article pages are always leaf nodes in this hierarchy and are thus more likely to be instances of the concept/category in the non-leaf nodes.

In this study, these article pages are assumed to serve as instances for which concepts can be mined. Regardless of whether they are instances or concepts, the Category Labels contained in them should be concepts. An Article page can be linked to a number of Categories through Category Labels, which usually reflect the relations between instances and concepts. For example, the page “*Atlas Shrugged*” contains several Category Labels such as “*Category: 1957 novels*”, “*Category: Novels by Ayn Rand*”, and so on. But Category Labels cannot be used directly, because they are given by the Wikipedia page editors. These Category Labels contain useful information but

are not easy to use directly because the Categories are given by the composer/editor of the page, and there are no uniform guidelines on how the Category labels should be formed. In other words, the Category labels can be given arbitrarily. Again, due to the somewhat arbitrary choices of the Category information, there are also no rules to ensure that the hypernym relationship must be in the various levels of Category information. The Category Labels also tend to be long and noisy, and not abstract enough to use directly. For example, the instance "*Real Madrid C.F.*" is a famous Spanish football club. The corresponding Article page is associated with a number of Category labels such as "*Category: football clubs established in 1902*", "*Category: Spanish football clubs*", "*Category: Madrid football teams*", and so on. Thus, the direct use of Category Labels as a complete token for concept terms is not appropriate. Instead, as can be seen from this example, the term "*football clubs*", which appeared in all three labels, is a good candidate as the corresponding concept.

Based on this observation, in each instance page, a set of all Category Labels is further broken down into smaller component words. By collecting the statistical significance of all components words, only certain component words (unigrams or bigrams only) are selected as the concept term. More specifically, the Category Labels are split by stop words such as "in", "of", etc. Then, the frequencies of both the unigrams and bigrams of components are calculated and the component with the highest product of frequency and length will be considered to be the most general

concept term in all of the Categories associated with this instance. As in the collection of Leaf nodes, there are both Article pages and Category pages with the same title; the Category Labels of both should be merged before the lexical analysis is conducted. The algorithm for the extraction of concepts is displayed in Figure 12.

Among the steps in Figure 12, steps 1-3 are to collect the titles of Article pages, steps 4-12 are to collect unigrams and bigrams of Category Labels as candidate concepts, and steps 13-14 are to select one concept from the candidates for each instance.

```
1 For each selected Article page p,
2 |  Save the title name t in T;
3 End for (* end of extraction of all titles *)
4 For each title name t in T,
5 |  Save all the Category Labels c pointed by t to C_t (including duplicates);
6 |  For each Category Label c in C_t,
7 |  |  Split c by stop words from a given stop word list;
8 |  |  Save all split component words w into W_t (including duplicates);
9 |  End for (* extraction of component words *)
10 |  For each w in W_t,
11 |  |  Collect unigrams and bigrams of w with their frequencies into G_t;
12 |  End for (* collection of statistics of component terms *)
13 |  Select g from G_t with the highest product of frequency and length
   |  (*g is the concept for t *)
14 End for (* completed for one title *)
```

Figure 12 Algorithm of Concept Extraction from Category Labels in pseudocode

5.1.4. Simple Combination of Annotated Data

The above extracted semantic data from Wikipedia are used in concept mining individually. Simple combinations of them may lead to improvements. As a simple combination, it can start from the annotated information with the highest precision and lowest coverage, and then the one with middle precision and coverage will be applied to the uncovered part, and so on till the last kind of resource with lowest precision but covering most of the objective instances. The details and practical order of combination depend on the precisions and coverages of individual annotated semantic information. The experiment and evaluation part will give more details and performances of simple combinations as the baseline for comparison with the SVM method.

5.1.5. Applying Support Vector Machine (SVM)

The SVM method is used to take all the features from different kinds of annotated semantic information and statistical data, the binary classifier of which is based on each pair of instance pages and concept candidates to indicate whether their binding is correct or not. To use the SVM model, support vectors are defined for each concept candidate of a given instance; features are selected based on the semantic annotated data in the definition sentences and the category labels. The features for an SVM classification are defined and listed in Table 10. As a feature, “Source” indicates where the concept candidates are from, definitions, category labels or other annotated data. PoS tags state that given concept candidate is tagged as which kind of noun,

general nouns in plural/singular format or proper nouns in these two formats. The other four features are based on the observations and statistical information when all the annotated data are combined.

Features	Descriptions
Source	which resource is a current phrase or word from ({{Infobox}} Structures, definition sentences, or Category Labels) and frequency
POS tags	whether the POS tag of the last word is a noun (one of NNS, NN, NNPS, NNP) and the frequency of different tags
Disambiguation Information	whether a given instance page has a quote to indicate the domain information of a given instance, such as the title "*John Thompson (politician)*", which indicates the given instance "*John Thompson*" is a "*politician*"
Head word of Category	whether a given phrase or word is the head word of the Category Labels of the corresponding instance page
Unit as other candidates	whether a given phrase or word is the head word of other candidates of the corresponding instance page
Rank in candidate frequencies	counts all the frequencies of concept candidates as a concept of different instance pages

Table 10 Features from Wikipedia Resources for SVM

As the unigrams and bigrams as component words may be meaningless fragments, the head words and quoted information of Wikipedia article page titles are also used as features for the SVM model.

5.1.6. Experiments and Evaluation

The evaluation of the proposed methods is conducted on the English Wikipedia with the cut-off date of November 30th, 2006. The whole corpus contains about 1.1 million Wikipedia pages. Considering that the concepts that are obtained may differ according

to the resources involved, the evaluation is based on the result of instance and concept pairs, which can also be considered an evaluation of concept-instance relations. For example, the concept for the page "*Atlas Shrugged*" as an instance is {{Infobox *book*}} according to the {{Infobox}} Structure. But it is considered a "*novel*" when a definition sentence and Category information are used. Both "*book*" and "*novel*" are correct concepts for the instance "*Atlas Shrugged*".

As the association between concepts and instances are many-to-many relations, the evaluation is applied on about 700,000 Leaf pages including the titles of article pages and Category pages as instance names. A sampling method is used to randomly select instances to manually evaluate whether their associated concepts are correct. For each set of results according to different resources and methods, 400 samples are selected to limit the margin of error[9] to within 5%. The evaluation criteria for the proposed method are the most general precision and coverage in terms of the article pages.

$$Precision = \frac{\text{number of instances with correct concepts}}{\text{number of instances with acquired concepts}} \quad (5)$$

$$Coverage = \frac{\text{number of instances with correct concepts}}{\text{number of all instances}} \quad (6)$$

It should first be pointed out that the coverage is not the same as recall because it only indicates the resources used in the extraction. However, to give the reader a sense

[9] In *Wikipedia, The Free Encyclopedia*. Retrieved 03:46, May 5, 2009, from http://en.wikipedia.org/w/index.php?title=Margin_of_error&oldid=287987112

of balanced performance for both precision and coverage, a so-called F'-measure is introduced below.

$$F' - measure = \frac{2 \times Presision \times Coverage}{Precision + Coverage} \quad (7)$$

When using different Wikipedia information, the evaluation must be done respectively. For example, the concept for the page "*Atlas Shrugged*" is *book* according to the embedded annotated information in the {{Infobox *book*}} Structure. But "*Atlas Shrugged*" is considered a "*novel*" when a definition sentence and corresponding Category information are used. For the instance "*Atlas Shugged*", both "*book*" and "*novel*" are correct concepts.

	Infobox (INF)	Definition Sentences			Categories (CAT)	SVM (Top1)	SVM (Top3)
		BDS	NBD	ADS			
Precision	90.1%	76.7%	33.2%	73.2%	75.3%	85.5%	91.2%
Coverage	17%	77%	6.7%	83.7%	97.1%	100%	100%
F'-measure	28.6%	76.9%	11.2%	78.1%	84.8%	92.2%	95.4%

Table 11 Performances of different resources

Row 1 of Table 11 shows the performance when using the {{Infobox}} Structures, labelled as **INF** (INFobox), as the baseline. In INF, only 1,201 concepts are acquired for about 111,623 instances. For example, the concept "*company*" has 2,585 instances, such as "*Microsoft*" and "*Bank of China*". Among the extracted concepts, 90.1% are correct. However, the coverage is about 17% because only 17% of all Article pages contain the {{Infobox}} Structure. Thus, the F'-measure reaches only 28.6%.

For definition sentences, a POS tagger (Tsuruoka and Tsujii 2005) is employed to identify the noun phrases after the verbs. It tags general noun phrases as NN(S) and proper nouns as NNP(S). Our algorithm prefers to select NN(S) as concepts. Only when NN(S) is not present, will NNP(S) be taken as the concepts. As be-verb sentences definitely contain the *is-a* relation, the performance of be-verb sentences is separately evaluated in Table 11, labelled as **BDS** (Be-verb Definition Sentences). The evaluation of non-be-word sentences and all definition sentences are labelled as **NBD** (Non-be-verb Definitions) and **ADS** (All Definition Sentences), respectively. The precision of BDS is 76.7%, better than that of ADS by 3.5%, which supports the assumption that be-verbs are more likely than non-be-verbs to indicate definitions. The precision of the non-be-verb sentences is only around 33.2%, about 40% less than the be-verb sentences. But by adding non-be-verb sentences, the coverage is 6.7% greater than that of using be-verb sentences only. So the F'-measure of ADS is 1.2% better than that of BDS. ADS can cover 83.7% of all Article pages and gives a much improved coverage compared to that obtained when using the {{Infobox}} Structure. Errors in precision are caused by two kinds of problems. The first kind contains the corresponding concepts in the sentences, yet the extraction of NN and NNP cannot identify these concepts correctly. For example, in the sentence "*Vai Sikahema was an NFL running back who played for 8 seasons from 1986 to 1993.*", "*Running back*" is the correct concept. But the algorithm only extracted "*running*" as the concept. The

second kind does not contain the corresponding concepts. For example, in the sentence "*Denham railway station is on the Chiltern Line out of Marylebone towards High Wycombe*.", although "*Chiltern Line*" is a relevant noun phrase to "*Denham railway station*", it is not the right concept. In fact, the corresponding concept should be "*railway station*", which does not even appear after the be-verb. By a rough estimate, 62.5% of errors fall into the first kind, and 37.5% fall into the second kind.

The performance of using Category information is listed in Table16, labelled as **CAT**. The precision is between ADS and BDS. Nearly half of the mistakes that arise are due to place or facility names whose Category Labels are given by other hypernym place names that are also instances rather than concepts. The hypothesis that instance pages should be assigned to Categories that are concepts is obviously incorrect in these cases. For example, the instance "*Orwell, New York*" has two Categories: "*Oswego County, New York*" and "*Towns in New York*". Neither "*county* "nor "*towns*" is selected because the most frequently used phrase according to the gram-selection method is "*New York*". This problem is not caused by the selection algorithm. In place names, organization names, and other names where there is a natural hierarchical structure, instances can be linked to other instances of a higher level. By applying a simple pre-processing rule to address this problem, an improvement in precision of 7.5% is already reflected in Table 11. Categories cover 97.1% of Article pages. The uncovered part is likely caused by the substandard editing of pages on the part of the

editors. The F'-measure of CAT reaches 84.8%, a little better than the methods using ADS and the best among all of these three methods.

According to the coverages and precisions of different annotated semantic data listed in Table 11, the order of their simple combination should be {{Infobox}} structures, definition sentences, and categories. As there are two levels of definition sentences, the simple combination is conducted respectively. Figure 13 shows the evaluation of simple combination of {{Infobox}} structures, definition sentences, and categories, labeled as **INF+BDS+CAT** and **INF+ADS+CAT**, respectively. INF+BDS+CAT and INF+ADS+CAT methods are combined on the condition that if there is no presence of {{Infobox}} structure in corresponding pages, definition sentences are considered. If there is no appropriate NN or NNP in BDS/ADS, the corresponding category information will be used.

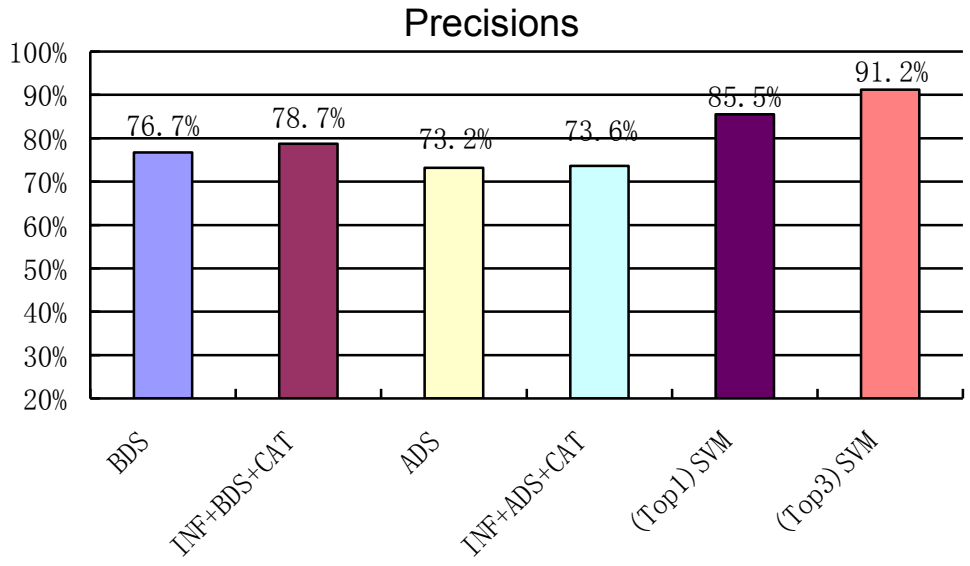

Figure 13 Performance Comparison of Simple Combined Annotated Data and SVM Method

Figure 22 indicates that the combination of INF+BDS+CAT gives an additional improvement of 2.0% to BDS. The improvement to ADS is only about 0.4%, this makes the combination more worthwhile for BDS. The fact that INF+BDS+CAT outperforms INF+ADS+CAT is because BDS information is more accurate. For example, BDS cannot find a concept for the page "*Domestic violence*" because its definition "*Domestic violence occurs when a family member, partner or ex-partner attempts to physically or psychologically dominate or harm the other.*" does not contain the be-verb. But ADS assigns "*family member*" as the concept which is not correct. However, as the page "*Domestic violence*" does not have an embedded {{Infobox}} structure, the combined method INF+BDS+CAT can identify "*violence*" as the result according to the category labels in the given article page.

Besides these simple combinations of resources, SVM is also applied to exploit more reasonable combinations of resources, and additional features may affect the result. Before carrying out the experiment of the SVM method as a further combination, there is a filter for concept candidates from definitions and Category Labels. This is to avoid having a test set that is overly large in scale, as the object of classification are pairs of instances and their concept candidates. For definitions, only candidates from BDS are considered to pair up with their corresponding instances as candidate pairs. For Category Labels, candidate unigrams and bigrams with the highest frequencies according to their instances are considered.

One hundred and six pairs of instances and their most appropriate concepts are selected as the training set. Among them, 99 are positive examples and the other 6 are negative ones. All other instance-concept candidate pairs are taken as the testing set. There are two evaluation criteria for SVM, as the SVM method gives a ranking of the scores of different concept candidates for the same concept. Corresponding to the methods using individual Wikipedia resources, the one with highest score is considered to be the most appropriate concept for each instance page, referred to as SVM(Top1) in Table 11. On the other hand, those ranked among the top three concept candidates in SVM, referred to as SVM(Top3) in Table 11), are also evaluated as a looser criterion for the evaluation of concepts, which gives a relatively reasonable evaluation to the SVM method itself. Table 11 demonstrates that if the concept candidate ranked number one in the SVM result is evaluated, the performance can reach 85.5% after the combination of resources and other added features are considered. If each top three ranking concept candidate for each instance is evaluated, the precision reaches 91.2%, which means that the most probable candidates are indeed ranked in the top positions.

To analyse which feature or features in the SVM model have the most impact, performances are compared after different features are removed respectively, with the exception of the features indicating that the sources are Categories, Definition sentences, or other sources.

	POS tags	Domain	Head word	Unit	Rank
Precision	78.1%	84.9%	83.1%	82.3%	76.5%
Deduction	7.4%	0.6%	2.4%	3.1%	9.0%

Table 12 Influence of different features in the SVM model

Table 12 shows the precisions that are obtained by removing different features in SVM, and the corresponding deductions from the overall precision of SVM (Top1). According to Table 12, the frequency ranking of concept candidates in has a greater influence on performance than adding information in POS tags to concept candidates. It is probably because the normalized frequency information is based on an analysis of all concept candidates. This is a more accurate approach than examining the frequency of certain POS tags based on individual concepts. Moreover, in the ranking list of concept candidates, a stop word list is applied to delete some noisy words such as "*birth*" and "*death*", and some names of places such as countries and states. The Domain information has the least influence on the SVM result. This is probably because the coverage of this domain information is very low in the instance pages. The percentage of instance pages with domain information in the title is only 4.0% of all Wikipedia article pages.

5.2. Attribute Extraction from Wikipedia

A concept is associated with a set of attributes, from which relations to other concepts are connected and can thus be derived. An ontology can then be built by organizing these concepts through their relations in a hierarchy according to their attributes. It is impractical and sometimes controversial to manually define an attribute set for every

concept, unless the set of concepts is a closed set. On the other hand, attributes (sometimes attribute values) as the descriptive information of concepts are more likely to occur near the concept terms in the context. However, a corpus-based approach suffers from the data sparseness problem, so semantic and knowledge rich corpora such as Wikipedia (Wiki) can be better choices to mine attributes for concepts. Wiki, as the largest online encyclopedia, provides definitions and descriptive information for terms indicating concepts and instances. The manually supplied tags, constructs, as well as internal and external links defined in Wikipedia pages make Wikipedia a better annotated resource for mining text and extracting information. Instances, which are extensions of concepts, usually occur more than their corresponding concepts in Wiki. In a real use, when distinguished from concepts according to associated attributes, instances can also be employed to assist in the extraction of concept attributes from Wiki.

In this section three extraction algorithms based on two kinds of information are proposed and compared. The two kinds of information are syntactic information in the context of given concept terms, and semantic information existing in the instances of concepts supplied by Wikipedia contributors. As for the three extraction algorithms, the first algorithm makes use of dependency relations in contexts, referred to as dependency analysis based attribute extraction (AE_{DA}). The second algorithm applies the FCA method to the context of given concepts, referred to as FCA-based attribute

extraction (AE_{FCA}). The third algorithm, employing instance information, is named instance-based attribute extraction (AE_{Ins}).

5.2.1. Dependency Analysis Based Attribute Extraction

The Dependency Analysis Based Attribute Extraction (AE_{DA}) method involves making use of the dependency relations existing in the descriptions of given concept terms to select attribute candidates of concepts, and then using WordNet to classify these candidates into their ancestors according to the hypernym relation. There are three steps for applying the AE_{DA} method to acquire attributes from Wiki. Figure 14 shows the main frame of AE_{DA}, which begins from the extraction of the concept description and ends at the attribute induction module, after acquiring the dependency relations between terms.

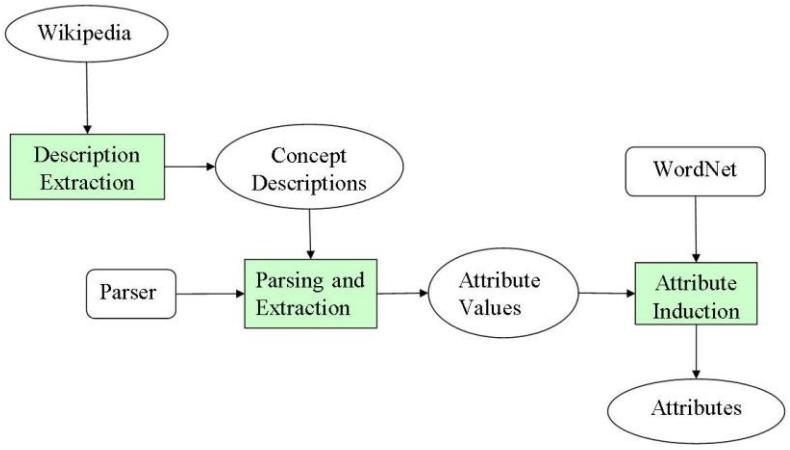

Figure 14 The Frame of Dependency Analysis Based Attribute Extraction

Before the whole process, the attributes that need to be acquired according to the dependency relations of concept terms should first be defined. Usually, attributes are considered to be the internal characteristics of concept terms, which can be used to

distinguish one concept term from others. In fact, the attributes of one concept or term as an object should contain more information than only the internal characteristics. Of course internal characteristics should be considered as attributes. Such attributes are usually represented by the nouns and adjectives in the context as modification words to given concept terms, referred to here as static attributes. Besides these inherent attributes, some properties will be shown through the actions of given concept terms to other ones and expressed by the subject-verb-object structures existing in the description sentences. These properties should also be considered as attributes of concept terms as objects, referred to as dynamic attributes. The roles and parameters in these dependency structures are displayed in Figure 15.

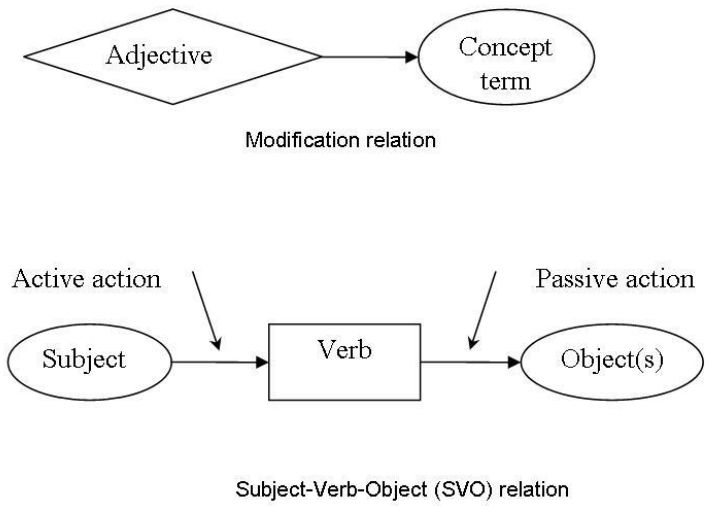

Figure 15 Parameters in dependency relations

There are three steps for applying the AE_{DA} method to acquire attributes from Wiki. The first step is to obtain a small domain corpus for targeted concepts in order to ensure the correlation of attribute candidates. For example, a small IT domain corpus

can be acquired by fixing a scope of Wikipedia pages through the method proposed in Chapter 3, and keeping the text parts of the top 1,000 pages most relevant to the given category page named “Information Technology” as the starting point. Thus, for a given concept *C*, the descriptive sentences containing *C* from the Wikipedia article page named *C* can be acquired by AE_{DA}.

The second step is to acquire attribute candidates with a close relation to *C*, and then rank these candidates. The dependency relations in which *C* participates can be extracted with the help of NLP tools. Two kinds of dependency relations are targeted in this method. The focus of this step is to extract the nouns and adjectives as modifications, and those in the subject-verb-object structures reflecting interactions of given concept terms to other ones. This step can be achieved by applying a full parser. The result of this step is a set of candidates, including attributes and attribute values that are mixed together. The candidates are then ranked according to their frequencies of co-occurrence with *C*.

The third step is designed for the purpose of integrating some mixed attributes and attribute values. The positions of these attributes and values in WordNet are located in and their one-level ancestors according to the hypernym relation in WordNet are acquired. The most commonly used ancestor synsets in the statistical information contained in WordNet are selected to remove the attribute values as instances. This is done because attribute values are more likely to be descendants than

ancestors of attributes in a hypernym relation. If more than one attribute candidate point to the same ancestor *F*, *F* is then considered to be the attribute of *C*, to indicate the whole attribute candidate set pointing to it. Otherwise, candidates of attributes that are not grouped by WordNet will themselves be considered as attributes to ensure the coverage.

5.2.2. FCA-Based Attribute Extraction

The approach of FCA-Based Attribute Extraction (AE_{FCA}) is in fact to apply the FCA method to Wikipedia article pages as a corpus. For the FCA method, the object set here is the given concept term list acquired from the IT domain, while the attribute set is the set of all content words in the context of these given concept terms.

The procedure of AE_{FCA} also contains three steps. The first step is similar to the first step of AE_{DA}. What is different is that sentences containing given concepts in all Wikipedia article pages are collected in order to obtain more reasonable statistical information. The second step is to acquire attribute candidates of concept terms. Content words such as nouns, verbs, and adjectives in the [-5, 5] window of a targeted concept term are considered to be attribute candidates of this concept term. They are ranked according to the frequency of their co-occurrence with targeted concept terms. The step of hypernym induction with WordNet is skipped to emphasize the effect of co-occurrence frequencies. Therefore, based on these three steps, the AE_{FCA} method can be considered some kind of variation on the AE_{DA} in extracting attributes for

concepts.

5.2.3. Instance-Based Attribute Extraction

5.2.3.1. The {{Infobox}} Structures in Wiki

An {{Infobox}} in Wikipedia is a consistently formatted table that is mostly present in articles with a common subject (concepts) to provide summary information and to help improve navigation to closely related articles (instances and attribute values) in that subject. In fact, an {{Infobox}} is a generalization of a taxobox (from taxonomy), which summarizes information for a subject or subjects.[10] An example displaying the structure and contents of a general {{Infobox}} structure in Wikipedia pages is shown in Figure 16.

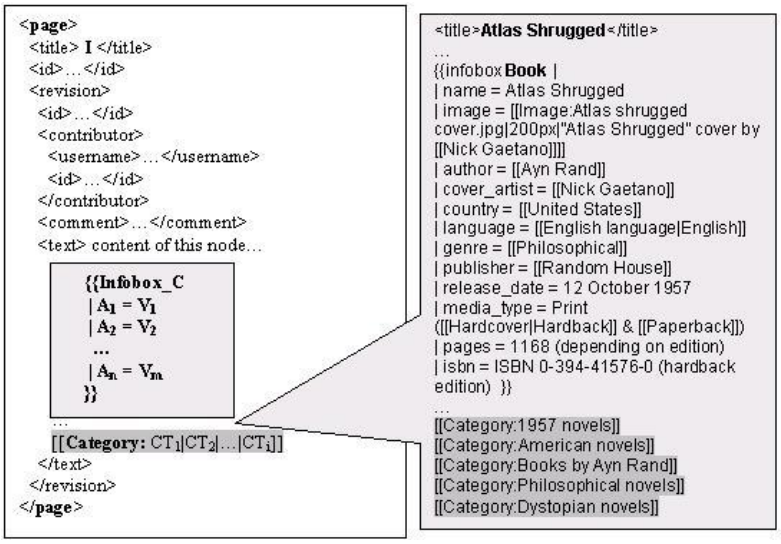

Figure 16 An Example of a Wikipedia Page with {{Infobox}} and Category Information

Figure 16 shows the syntax of an {{Infobox}} in a Wikipedia article page and an example of an {{Infobox}}. The left part of Figure 16 illustrates the syntax of a Wikipedia article page as an instance with an {{Infobox}} and Category List. The

[10] http://en.wikipedia.org/wiki/Wikipedia:Infobox

right part gives an example of the contents and labelled categories of an {{Infobox}}. It displays an article page as an instance titled [[Atlas Shrugged]], which is the name of a novel, and the related information according to a predefined {{Infobox}} syntax. The first line of an {{Infobox}} provides {book} as the concept to which the instance [[Atlas Shrugged]] belongs. The subsequent list defined using "=" are attributes and their values of {book} that are associated with the instance [[Atlas Shrugged]]. The terms or phrases on the right part are values of the corresponding attributes on the left-hand side. For example, in the entry *author=[[Ayn Rand]]*, author can be considered an attribute of {book}, denoted by $\{author\}_{book}$, and [[Ayn Rand]] is the value of $\{author\}_{book}$, which can also be considered an instance of the attribute $\{author\}_{book}$.

Consequently, an {{Infobox}} can be used for two purposes. First, it can be used to acquire attributes for a given concept. Second, it can be used to estimate an appropriate semantic type for each attribute of a concept. It should be pointed out that even though each {{infobox}} contains a list of attributes, a different {{infobox}} as an instance of the same concept may use a different list of attributes. Therefore, there is the issue of how to identify a common set of attributes that are considered most appropriate for a concept. In Wiki, an additional list of category information is contained in a page in the form of "*[[Category:*CT_1*|*CT_2*| ...|*CT_i*]]*", as shown in Figure 16, where CT_1 to CT_i are the total *i* number of categories that the page editors

consider to be related to the current page. This list of category information is used to classify articles in Wikipedia and serves as are used to classify articles in Wikipedia and serve as the table of contents for the whole Wikipedia structure. The categories of an article page should reflect the classifications to which the subject of the article belongs, or topics to which it is related (Lih 2004). In this section, Wikipedia category information is used to estimate the semantic type of the attributes.

5.2.3.2.Acquisition of Concept Attributes from {{Infobox}}

As {{Infobox}} structures in instance pages denote the corresponding concepts to which they belong, the first step is to collect all of the article pages as instances through the identification of {{Infobox}} structures in Wiki. As shown in Figure 16, the formats of a Wikipedia page and {{Infobox}} structure are relatively fixed. Thus, the corresponding information can be acquired by patterns. Each attribute *A* of different instances(*Is*) belonging to one concept *C* is collected into one set $S_A(C)$ with the help of two 2-level hash tables, one for concept and instance mapping named $H_I(C)$ and the other for instances and corresponding attributes with values, referred to as $H_A(C)$. Then, the attribute identification process can be achieved by the Instance-Based Attribute Extraction (IBAE) algorithm to obtain $S_A(C)$. $S_A(C)$ only contains a set of attributes whose number of appearances in $H_A(C)$ is bigger than a majority threshold value.

```
IBAE Algorithm
{  for each C in H_I(C)  {
        H(I)= H_I(C)->{C};
        Count_of_Ins = scalar(H(I));
        for each I in H(I) {
            for each A in H_A(I) {
                H_A(C)->{A}++;
            }endfor
        }endfor
    }endfor
    for each C in H_A(C) {
        Freq= H_A(C)->{A};
        if(Freq>=THRE*Count_of_Ins) {
            put A into S_A(C)
        }endif
    }endfor
}endalgorithm IBAE
```

Figure 17 Algorithm of instance-based attribute extraction in pseudocode

As shown in Figure 17, the IBAE algorithm can be divided into two parts. The first part involves counting the number of appearances of each attribute for *C*. The second part is the selection of qualified attributes according to the majority threshold value, named *THRE*, which is an experimentally determined algorithm parameter. In the IBAE algorithm, all counts of appearance are stored in a 2-level hash $H_A(C)$, with concepts as the first level of keys and attributes as the second level. The attributes associated with more than a threshold of instances for the same concept will be selected as qualified concept attributes. All of the qualified attributes are stored in the set $S_A(C)$.

5.2.3.3. Identifying Attribute Types

In the conclusion of their work, Almuhareb and Poesio (2004) mention that concepts can be described better if both attributes and attribute values are supplied. Generally

speaking, in ontology construction, the same attribute name can be used for concepts of an ontology in the same domain or concepts in ontologies of different domains. One way to identify a specific attribute for a concept is to identify the distinct semantic type of this attribute. For example, the concept {Airliner accident} has an associated $\{site\}_{Airliner\ accident}$ as an attribute to indicate the location of an accident, whereas for organizations such as {UKschool}, an associated attribute $\{site\}_{UKschool}$ can be the website address of this organization. This example shows that if the *domain of the attribute value range* (called the *attribute type*) is identified, it can further qualify the semantic type of the specific attribute.

Attribute values can be stated in different formats such as words, sentences, and tables. Identifying attribute types is not straightforward work because attribute values are quite arbitrary in their form of presentation. This section proposes to use the frequency of Wikipedia article names as instances to indicate the implicit inclination of attribute types. This approach is based on two considerations. First, descriptions of attribute value contain useful information but are usually not well formed. Therefore, the use of NLP tools such as parsers are not really suitable for this task. Second, most attribute values contain related Wikipedia article names using a predefined format, so it is easy to acquire these Wikipedia article names. In fact, these article names can be considered as named entities marked in descriptions of attribute value. Consequently, these Wikipedia article names are taken as key words in descriptions of attribute value

for analysing attribute types. For example, the attribute {developer}$_{software}$ of [[REXX]](a programming language) has a value "*Mike Cowlishaw & [[IBM]]*", the format of which indicates that there is a Wikipedia article entitled "IBM". Thus, the related article page name in attribute value is identified. If there is no Wikipedia article name marked in an attribute value description, the substrings of the attribute values will be used as key words to look for and match homonymic Wikipedia article names. Then, the category information labelled in the matched Wikipedia pages can be used to identify the type for each attribute of a concept. For example, attribute {developer}$_{software}$ of [[ICQ]] takes [[AOL]](American online) as a value and the same attribute {developer}$_{software}$ of another page [[Internet Explorer]] takes [[Microsoft]] as a value. Then, a 2-level hash table pointing Wikipedia article names to corresponding categories that can be employed to obtain categories of the Wikipedia pages named [[Mike Cowlishaw]], [[IBM]], [[AOL]], and [[Microsoft]], such as "Category:Computer Programmers", "Category:Software companies of the United States", and "Category:Companies listed on NASDAQ". The most frequently used categories will be selected as the attribute type of a given attribute. Similar to the IBAE algorithm, two 2-level hash tables are used in the Attribute Type Identification (ATI) algorithm. One is $H_V(A)$, which collects key words in attribute values and maps corresponding attributes to them. The other is the category hash, which maps article page names to gram lists from corresponding attribute values, referred to as $H_{CAT}(V)$.

```
ATI Algorithm
{  for each A in H_V(A) {
      MAX=0;
      for each V in H_CAT(V) {
         for each CAT in H_CAT(V) {
            Freq = H_CAT(A)->{CAT}++;
            If (Freq>= MAX){
              MAX=Freq;
            }endif
         }endfor
      }endfor
   }endfor
   for each A in H_CAT(A) {
      F = H_CAT(A)->{CAT};
      If (F==MAX) {
        put CAT into S_CAT(A);
      }endif
   }endfor
}endalgorithm ATI
```

Figure 18 Algorithm for identifying attribute types in pseudocode

The structure of algorithm ATI in Figure 18 also contains two parts that are similar to IBAE. The first part is to collect the candidate types of an attribute. The second is to select the most appropriate semantic type of an attribute. In the ATI algorithm, *A* is an attribute, *V* is a key word in attribute value, and *CAT* is a category label linking to one key word *V*. *Freq* records the number of article page names as instances linking to the same category. *MAX* records the maximum number of votes for each attribute. The first part of the ATI algorithm is to collect the categories with key word values directly linking to them and to record them with *Freq*. The candidates are considered to be possible attribute types and are stored in $H_{CAT}(A)$. In the second part, all of the qualified attribute types with the highest *Freq* will be stored in the result set, namely $S_{CAT}(A)$.

As Wikipedia is an open-content and collaboratively edited encyclopedia, some category labels are not formal enough to be considered as attribute types. For example, categories of dates are too specific, such as "category:1980" for the year 1980. The semantic type induced from this value should be DATE rather than a specific date value. In the experiment, the ATI algorithm was also integrated with some pre-processing and post-processing steps to handle these special cases, which will also be discussed in the experiment part.

5.2.4. Experiments and Discussions

The experiments are based on the XML file of English Wikipedia with the cut-off date of November 30th, 2006. It contains about 1.1 million article pages and function pages such as category pages. In the experiment on extracting attributes based on different kinds of information, the resources that are employed are all descriptive sentences containing target concepts in Wikipedia article pages. A set of core terms from the IT domain is used to mine their attributes. The core terms are the most frequently used IT concept terms as components of the top 1,000 article names most relevant to the IT domain, which has been described in Chapter 3. For example, the concept term "*network*" is a key term contained in Wikipedia article page titles such as "*Telecommunications Management Network*", "*Markov network*", and "*Artificial neural network*". The names of these Wikipedia pages are not directly used because they are long and are not qualified concept terms. After a rough manual filtering, the

core term set containing 653 concept terms is considered to be the target concept set. For the IBAE experiment, a total of 110,572 article pages with {{Infobox}} structures are used to extract data from their {{Infobox}} structures. In fact, the total number of {{Infobox}} structures is 111,408 because an article page can contain more than one {{Infobox}} structure. For example, "*Arnold Schwarzenegger*" is an instance of the concept *Actor* and also an instance of the concept *Governor*, which contains different attributes in two separate {{Infobox}} structures.

5.2.4.1. Evaluation of the AE_{DA} method

In the first step of the AE_{DA} method, around 6,000 relevant sentences were parsed by the parser MINIPAR (Berwick et al. 1991). This was done to extract static and dynamic attributes among the dependency relations existing in the context of given concepts after parsing. The result is made up of about 1,650 attributes, which translate to about 2.5 attributes for each given concept. As the information contained in a single Wikipedia page is limited, most candidate attributes occurred no more than twice. Therefore, the threshold of frequency is set to 1 and details of the performances of the static and dynamic attributes under this threshold are listed in Table 13.

	Static attributes	Dynamic attributes	Overall
Original Number	1,777	2,619	4,396
Hypernym	553	1,081	1,634
Precision	42.5%	32.7%	36.4%

Table 13 Details of Attributes acquired by the AE_{DA} method

As shown in Table 13, the absolute number of attributes is not large, even before

the use of WordNet. By using WordNet, about one third to one half of these attributes are mapped into their hypernyms. The dynamic attributes that are extracted have lower precision partly because it is more difficult to extract attributes through predicate relations. The results of the error analysis are listed in Table 14. There are mainly three sources of errors: 1) Parsing errors (Parsing in Table 14), 2) Hypernym induction errors (Hypernym in Table 14), and 3) Other errors are caused by the algorithm itself (Algorithm in Table 14). As for parsing mistakes, take the sentence fragment "*...part of the data is modified...*" as an example. The word "*modify*" is supposedly an attribute of the concept "*data*", which means that "*data*" has the capacity to be modified. However, because the parser can only identify binary relations, it wrongly identified "*part*", the modifier of "*data*", as the subject of the "*modify*" action according to the sentence fragment. Moreover, the parser cannot reflect all of the modifications to the given concepts. Only 492 out of 653 concepts can be assigned attributes by AE_{DA}. For the hypernym induction errors, take the term "*developer*" as an example. It is the attribute of the concept "*software*". "*Developer*" has more than one hypernym according to different semantic meanings. The most frequently used synset as the hypernym of "*developer*" is labeled "*photographic equipment*" instead of the correct "*creator*", which makes the attribute of "*software*" an error. Table 14 lists the percentages of different error sources based on the extraction results.

Error Sources	Parsing	Hypernym	Algorithm	Total
Static attributes	23.0%	17.9%	16.6%	57.5%
Dynamic attributes	26.5%	26.4%	17.2%	67.3%

Table 14 Error Analysis for the AE_{DA} method

Table 14 shows that more than half of the errors are caused by the parsing section and by the induction of hypernyms using WordNet. One reason for this is that the step of parsing limited the context words of concepts to candidates of attributes, which also reduced the effect of frequency. The use of WordNet to map attributes into their hypernyms did not much benefit the whole performance because it is not easy to select the most appropriate synset of a hypernym. Therefore, the experiment of the AE_{FCA} method is conducted to try to smooth out the problems caused by these two parts in AE_{DA}.

5.2.4.2. Evaluation of the AE_{FCA} method

The AE_{FCA} is applied to make use of a part-of-speech tagger (Tsuruoka and Tsujii 2005) instead of a parser to locate the attribute candidates for given concepts. Sentences containing given concept terms in all Wikipedia article pages are extracted in order to obtain more reasonable statistical information. Then, content words in the [-5, 5] window of a targeted concept term are considered as attribute candidates of this concept term. They are ranked according to the frequency of their co-occurrence with targeted concept terms. On the other hand, the step of hypernym induction with WordNet is skipped to emphasize the effect of co-occurrence frequencies. Figure 19 displays the performances of attribute candidates under different frequency thresholds

(i from 1 to 10). Under each threshold of frequency, 400 concept-attribute pairs are sampled for evaluation. As precision is measured on attributes, whereas recall is measured on concepts, they are not measured on the same objects. Therefore, a so called *f*-factor is taken as a trade-off to balance both measures, where each f_i is defined in a similar way as the commonly used *f*-factor as given in formula (5):

$$f_i = 2*r_i*p_i / (r_i+p_i) \quad (8)$$

where r_i and p_i are the concept recall and attribute precision of the *i*th increment of the threshold, respectively.

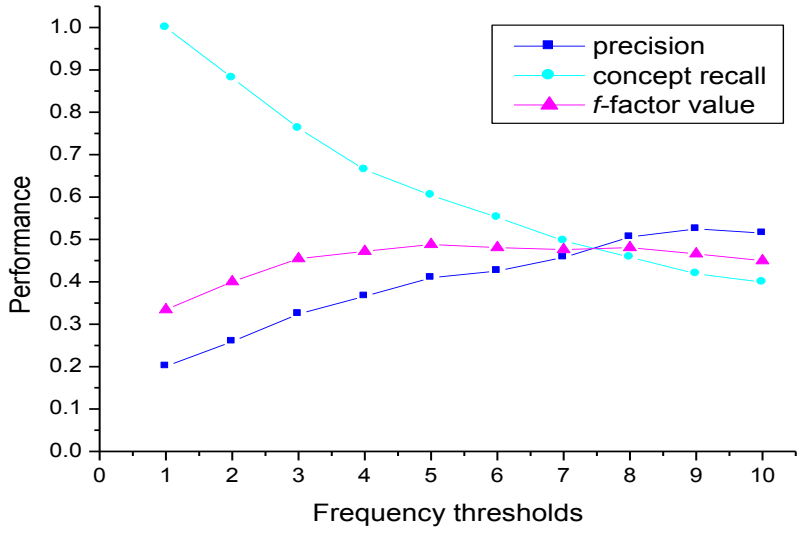

Figure 19 Trade-off performance of AE_{FCA}

As is shown in Figure 19, the attribute precision ranges from 20.1% to 52.7% when the threshold shifts from 1 to 9. The recall of concepts ranges from 100% with threshold of 1, to 39.9% with threshold of 10. The precision does not monotonically increase, which means that the quality of the attribute candidates is not a

single-variable function determined only by the co-occurrence frequency. In fact, an increase of frequency may lead to a loss of some qualified attributes. For example, "*value*" and "*size*" are two qualified attributes of the concept "*data*". But when the threshold changes from 9 to 10, these two attributes are filtered out. The *f*-factor reaches the peak point when the threshold is set to 5 and does not drop much when the threshold increases. The smallness of the drop is one benefit of the weakening downtrend of concept recall. At this point, the value of the *f*-factor is 48.8% with a precision of 41.0% and a concept recall of 60.4%. Although the precision is lower than in the AE_{DA} method at the point when the threshold equals to 1, the average performance is better than in the AE_{DA} method. The attribute-concept ratios (A/C) are also much higher than in the AE_{DA} method. Table 15 exhibits details of the ratios between concepts and attributes in AE_{FCA} according to different thresholds:

Threshold	1	2	3	4	5	6	7	8	9	10
A/C ratio	238.9	67.3	36.3	24.5	17.8	14.1	11.6	9.9	8.8	7.6

Table 15 Attribute/concept ratios under different thresholds of AE_{FCA}

From Table 15, it is obvious that the ratio decreases when the threshold increases. But even when the threshold reaches 10, the A/C ratio is still much higher than 2.5, the ratio of the AE_{DA} method. This means that more attributes can be extracted using the improved algorithm.

5.2.4.3.Evaluation of the AE_{Ins} method

The AE_{Ins} method involves starting from the {{Infobox}} structures in Wikipedia

pages. Experiments are conducted to evaluate the influence of *THRE*, which is an experiment-determined parameter ranging from 0.0 to 1.0, and its effects on the performance in terms of attribute precision and concept recall. The *f*-factor here is defined similar to the *f*-factor in AE_{FCA}. The same sampling method as in AE_{FCA} is also employed to estimate both the precision and recall in the evaluation. The performance trade-off is displayed in Figure 20.

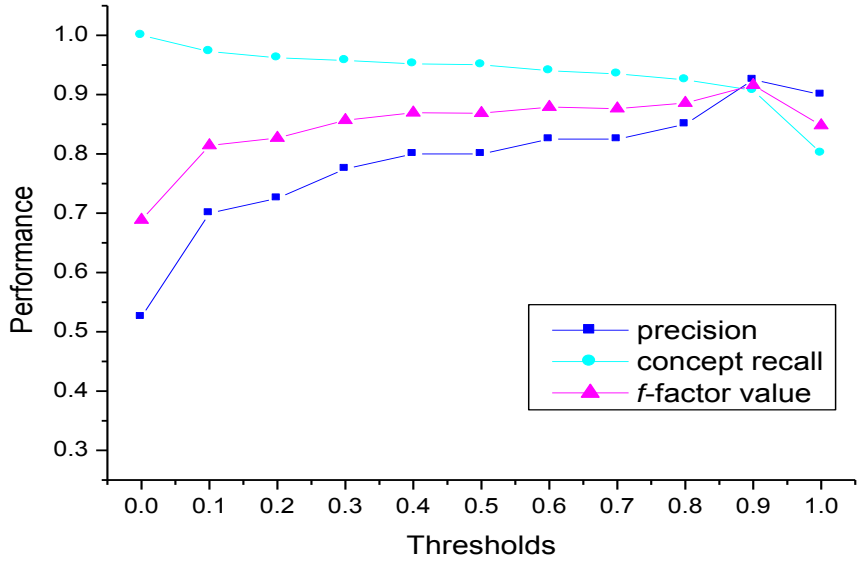

Figure 20 Trade-off performance of AE_{Ins} for selecting *THRE*

Figure 20 shows that the attribute precision ranges from 52.5% to 92.5% with an optimal point at around 0.9. On the other hand, the recall of concepts decreases from 100% to 80% when *THRE* reaches 1.0. For example, if *THRE* is set to1.0, 311 out of 1,109 concepts would have no attributes because there is no attribute that is shared by all instances for each of these concepts. For example, there are a total of 1,271 instance pages for the concept "*software*". When *THRE* is set to 1.0, this means that an

attribute can be considered to be qualified only if it appears in all of the instances. However, for some concepts such as "*software*", there is no such attribute under a threshold of 1.0. When *THRE* is reduced to 0.9, seven attributes common to 90% of the instances belonging to "*software*" are identified, including *name*, *developer*, *genre*, *operating system*, *license*, *latest released version*, and *website*. The highest value of the *f*-factor is reached when the threshold is set to 0.9. Thus, when *THRE* is at the balanced point of 0.9, the attribute-concept ratio is 10.7. At this point, attribute precision reaches 92.5% with a concept recall of 90.7%, and a trade-off *f*-factor of 91.6%. Table 16 supplies the attribute-concept ratios under different *THRE*s for AE_{Ins}.

THRE	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1
A/C ratio	45.6	15.2	13.8	13.0	12.3	12.1	11.5	11.1	10.7	9.3

Table 16 Attribute/concept ratios under different thresholds of AE_{Ins}

According to Table 16, the fluctuation of the attribute-concept ratio in the AE_{Ins} method is not as apparent as that in AE_{FCA}, which indicates that the manually supplied semantic information is more stable and reasonable in AE_{Ins}. However, most of the concepts derived from AE_{Ins} are in the general domain and only cover 172 concepts out of the 653 IT domain concepts. This means that the coverage of AE_{Ins} reaches only 26.3% for the given concept set in the IT domain.

5.2.4.4.Comparison and Discussion

After selecting appropriate thresholds for AE_{DA} and AE_{Ins}, the comparison of the AE_{DA}, AE_{FCA}, and AE_{Ins} methods is conducted under the threshold 5 for AE_{FCA} and

0.9 for AE_{Ins}. There are three factors that can reflect the quality of acquired attributes. The details of comparison are displayed in Table 17:

	Precision	Concept Recall	A/C ratio
AE_{DA}	36.4%	75.3%	2.5
AE_{FCA}	41.0%	60.4%	17.8
AE_{Ins}	92.5%	26.3%	10.7

Table 17 Comparisons of methods based on different kinds of resources

Table 17 states that the original AE_{DA} method covers most of the targeted concepts. However, it acquires the fewest attributes for concepts, which should also be considered to be a factor of integrated performance. The precision of the AE_{Ins} method is the best of all, because this method made the most of the manually annotated semantic-rich information in Wiki. Despite the high precision of the AE_{Ins} method, it can be applied only to those Wikipedia pages that contain {{Infobox}} structures. Thus, the method can cover no more than 17% of all article pages in Wikipedia and 26.3% of targeted concepts in the IT domain.

5.2.4.5. Evaluation of the Algorithm ATI

To evaluate the performance of attribute type identification, the optimal threshold value *THRE* of 0.9 is fixed. The evaluation of ATI examines the precision of acquired attribute types and how many concepts and attributes can be covered by ATI, referred to as the recall of concepts and attributes.

The initial evaluation of ATI uses 100 evenly distributed samples of <*concepts, attributes, attribute values*> by manual examination. The recall of typed concepts is

less than 50%, and attribute recall is also too low. A further analysis is then made to determine the reason for such a poor performance. As Wikipedia is a collaborative online encyclopedia, some editors list attributes with only names in the {{Infobox}} without supplying values, which gives rise to data sparseness. Also, the attribute values are not uniform in format, making it difficult to extract them even if they are present. In addition, some of the categories present in the Wikipedia pages are actual instance-level values rather than concept-level category names, as mentioned in section 5.2.3.3. Thus, the proposed method cannot acquire categories for all attributes, nor select the most appropriate attribute type. For example, the semantic type of $\{\text{clubs}\}_{\text{football player}}$ should be [football clubs], not instances of football clubs. However, most Wikipedia categories list the actual country names such as "*category: Spanish football clubs*" and "*category: Italian football clubs*". Therefore, the issue is to remove the instance information. In fact, simple pre-processing and post-processing of attribute values and categories can resolve the problem to some degree.

The pre-processing is done to eliminate the reference to named entities such as countries, cities, and so forth. An n-gram collection (n=1, 2 here) is applied to attribute values and categories. Unigrams and bigrams of these category labels are extracted as substitutes of categories, and those with the highest product of frequency and length will be considered to be a qualified attribute type. In fact, his approach involves using the components of a category instead of the category label itself. As a result, the

bigram [football clubs] will be extracted as an attribute type candidate.

The post-processing involves handling the errors caused by numeric values and hyperlinks. Some of the category labels in Wikipedia pages are actual attribute values rather than potential attribute types. For example, the attribute type of "*year 1200*" should be [year] rather than "Category:1200" containing the actual attribute value of year 1200. There are also cases where the category labels of attribute values, such as value 8,514,877 of $\{area\}_{country}$ for {Brazil} as an instance, are not defined as a Wikipedia article name. Some attributes are listed without given any attribute value, such as attribute $\{awards\}_{architect}$, which is empty for all instances of the concept {architect} in the Wikipedia version used here. According to the analysis, two simple induction rules are applied in post-processing. They are listed as follows:

- **R1:** If the text of an attribute contains only years or months or other date measurement words, its attribute type is labelled [DATE]; Otherwise, if they contain only numbers and delimiters, its attribute type is labelled [NUMBER]; All hyperlinks are labelled [LINKS].
- **R2:** If an attribute has no attribute values, its attribute type is labelled using the name of this attribute.

	AT Precision	Recall of Concepts	Recall of Attributes
ATI	16.7%	47.0%	14.0%
ATI_{Ngram}	28.2%	100%	67.0%
ATI_{Rules}	80.0%	100%	76.0%

Table 18 Comparison of Attribute Types Using Different ATIs

Table 18 shows the evaluation results of the original ATI (*ATI*), ATI+Preprocessing (ATI_{Ngram}), and ATI+Preprocessing+Post-processing (ATI_{Rules}). It can be seen that the original ATI covers less than half of the concepts. Also, less than one quarter of the attribute types are correctly identified. After adding the pre-processed part, coverage of the concepts and attributes reaches 100% and 67%, respectively. However, the precision is still no more than 30%, which means that category information alone is not enough. By applying two simple induction rules, the precision of ATI reaches 80% and the recall of attributes is also close to 80%. As the performance has improved quite significantly after applying n-grams and induction rules, a further analysis is carried out to look for reasonable explanations. Figure 21 shows the contributions of the original ATI, and the n-gram pre-processing and post-processing rules in identifying attribute types.

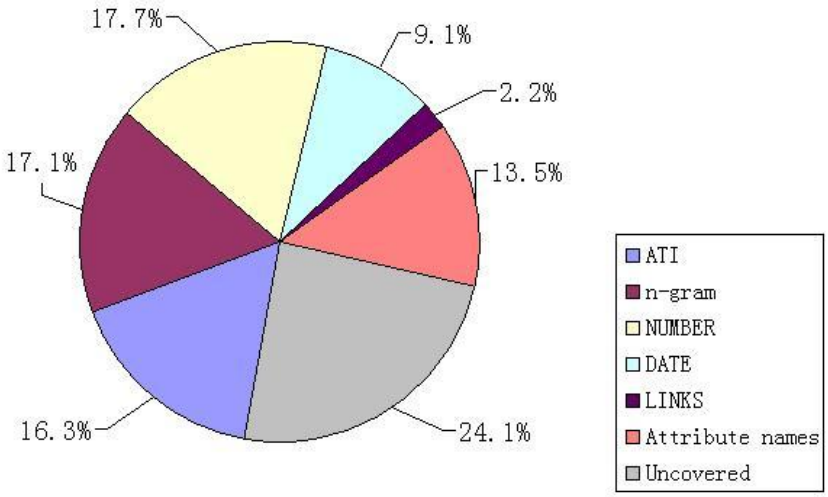

Figure 21 Distribution of Attribute Types

Figure 21 shows that pre-processing by using n-grams contributes an additional

17.1% to attribute type identification; and post-processing, including the use of NUMBER, DATE, LINKS, and attribute names, contributes to the identification of 42.5% of the attribute types including numbers, dates, links and attribute names. This explains why pre-processing and post-processing can significantly improve performance, so that many more attributes can be correctly categorized. However, about one quarter of attribute types still cannot be identified by the algorithm, which means that category information is not sufficient for identifying attribute types. The unvalued attributes also limit the precision of the algorithm. Other information in Wikipedia may be used in the future to improve the performance of ATI.

5.3. Chapter Summary

This chapter proposed a novel method of making use of the SVM model to extract concepts by combining annotated information from Wikipedia on instances for ontology construction. Several kinds of Wikipedia resources are used as the association between concepts and instances. The proposed method identified nearly 50,000 concepts for about 700,000 Wikipedia Article pages as instances, with a precision of 85.5%. This is a better level of coverage of concepts than has been achieved by existing work with a reasonable precision. The use of unigram and bigram statistics helps to eliminate instance information in conceptual descriptions, and the SVM method effectively combines different resources to identify the most appropriate concept terms for instances.

In addition, different methods making use of context information and instance information are explored to acquire attributes for concepts. Three factors are proposed to reflect the quality of attributes in different aspects. They are attribute precision, recall of concepts, and the ratio between attributes and concepts (how many attributes are acquired for each concept). The method based on the FCA model by extracting qualified context words as attributes covers 60% of given concepts and has the highest average A/C ratio, but has a relatively low level of precision with regard to attributes. The method based on instances covers fewer given concepts, but can attain a high level of precision of 92.5% and has a relatively stable and reasonable A/C ratio. Furthermore, it can suggest an attribute type for nearly 80% of acquired attributes, with a precision of 80%.

Chapter 6
Term Classification

According to the structure of an ontology, building an ontology with top-down methods needs different types of concept terms to serve as different roles and to be mapped in appropriate position of an ontology. Therefore, the terms representing different kinds of concepts should be classified first before further steps are taken. Part-of-speech (POS) tags usually carry some linguistic information about terms, so POS tagging can be seen as a kind of preliminary form of classification to distinguish the attributes belonging to given concepts and help map concept as nodes into the ontology, because features or attributes related to concepts under different POS types may be different. This chapter presents a simple approach to tagging domain terms for the convenience of ontology construction, referred to as *Term POS* (*TPOS*) *Tagging*. The proposed approach makes use of segmentation and tagging results from a general POS tagging software to predict tags for extracted domain-specific concept terms. No training is needed for this approach, and no context information.

Section 6.1 lays out the difference between term POS tagging and general POS tagging before introducing two possible schemes of term POS tagging. Section 6.2 discusses and evaluates the experiments of the candidate term POS tagging schemes. The experimental results show that the proposed approach achieves the best precision

of 95.41% for extracted terms and can be easily applied to different domains. Section 6.3 summarizes the whole chapter.

6.1. Term POS Tagging

Term POS tagging is different from general POS tagging tasks. In this study, it is assumed that a terminology extraction algorithm has already obtained the POS tags of individual words. For example, in the segmented and tagged sentence “计算机/n 图形/n 学/v 中/f ，/w 物体/n 常常/d 用/v 多边形/a 网格/n 来/f 表示/v 。/w” (in computer graphics, objects are usually represented as polygonal meshes), the term “多边形网格” (polygonal meshes) has been segmented into two individual words and tagged as “多边形/a” (polygonal /a) and “网格/n” (meshes /n). The terminology extraction algorithm would identify these two words “多边形/a” and “网格/n” as a single term in a specific domain. The proposed algorithm is to determine the POS of this single term “多边形网格” (polygonal meshes), thus the algorithm is referred to as TPOS tagging. It can be seen that general-purpose POS tagging and term POS tagging assign tags at different granularities. In principle, the context information of terms can help TPOS tagging and the individual POS tags may be good choices as classification features.

The proposed TPOS tagging algorithm consists of two modules. The first module is a terminology extraction pre-processing module. The second module carries out the TPOS tag assignment. In the terminology extraction pre-processing module, if the

result of the terminology extraction algorithm is a list of terms without POS tags, a general-purpose segmenter called ICTCLAS will be used to give POS tags to all individual words. ICTCLAS[11] has a precision of 97.58% on tagging general words in Chinese. Then, the output of this module is a list of terms, referred to as TermList, using algorithms such as the method described in the work of Ji et al. (2007b).

In this study, two simple schemes for the term POS tag assignment module are proposed. The first scheme is called the blind assignment scheme. In this scheme, a noun tag is simply assigned to every term in the TermList. This is done based on the assumption that most of the terms in a specific domain represent certain concepts that are most likely to be nouns. The result from this blind assignment scheme can be considered to be the baseline or the worst-case scenario. Even in the general domain, it is observed that nouns comprise the majority of Chinese words, making up more than 50% of all different POS tags (Wang 2006).

The second scheme is called a head-word-driven assignment scheme. Theoretically, the tag of the head word of one term is taken as the tag for the whole term. But here it simply takes the tag of the last word in a term. This is done based on the assumption that each term has a head word that in most cases is the last word in a term (Wang 2006). One additional experiment was done to verify this assumption. A manually annotated Chinese shallow Treebank in the general domain was used for the statistical work of Xu et al. (2005). There are nine different structures of Chinese

[11] Copyright © Institute of Computing Technology, Chinese Academy of Sciences.

phrases (Wu et al. 2003), but only three of them do not have their head words in the tail, which represent about 6.56% of all phrases. Following the earlier examples, the term "多边形/a 网格/n" (polygonal /a meshes /n) will be assigned the tag "/n" because the last word is labelled "/n".

There are many semanteme tags at the end of a term because terms are sometimes split into several segments and the last one is a semanteme. For example, "/ng" presents a single character postfix of a noun. But it would be improper if a term is tagged as "/ng". For example, the term "决策器" (decision-making machine) contains two segments as listed, with two components "决策/n" and "器/ng". It is obvious that "决策器/ng" is inappropriate. Thus, the head-word-driven assignment scheme also includes some rules to correct this kind of problem. As will be discussed in the experiment, the current result of TPOS tagging is appended, along with two simple induction rules that were applied in this algorithm.

6.2. Experiments and Discussions

The domain corpus used in the experiment contains 16 papers that appeared between 1998 and 2000, selected from different Chinese IT journals, with over 1,500,000 numbers of characters. The papers cover topics in the IT domain, such as electronics, software engineering, telecom, and wireless communication. The same corpus is used by the terminology extraction algorithm developed in the work of Ji et al. (2007b). In the domain of IT, two TermLists are used for the experiment. TermList1 is a manually

collected and verified term list from the selected corpus containing a total of 3,343 domain terms. **TermList1** is also referred to as the standard answer set to the corpus for the purpose of evaluating term extractions. **TermList2** is produced by running the terminology extraction algorithm according to the work of Van Rees (2003). TermList2 contains 2,660 items, of which 929 have been verified as qualified terms and 1,731 are not considered terms according to the standard answer in TermList1.

To verify the validity of the proposed method for different domains, a term list containing 366 legal terms obtained from Google searching results for "法律术语大全" (complete dictionary of legal terms) is selected for comparison, and named **TermList3**.

6.2.1. The Blind Assignment Scheme

The first experiment is designed to examine the proportion of nouns in TermList1 and TermList3, to validate the assumption of the blind assignment scheme. The first step in this experiment is to tag all of the 3,343 terms in TermList1 as nouns. The result shows that the precision of the blind assignment scheme is between 78.79% and 84.77%. The reason for the range is that there are about 200 terms in TermList1 that can be considered as being either nouns, gerunds, or even verbs without context as reference. For example, the term "局域网远程访问" ("remote access of local area network" or "remote access to local area network") and the term "极化" (polarization *or* polarize), can be considered to be either nouns if they are regarded as courses of

events or verbs if they refer to the actions for completing certain work. The specific type is dependent on the context, which is not provided without the use of a corpus. However, the result of the experiment does show that, in a specific domain, a much higher percentage of terms are nouns than other tags in general (Wang 2006). As to TermList3, the precision of the blind assignment is between 65.57% and 70.77% (19 mixed ones). TermList2 is the result of a terminology extraction algorithm and there are non-term items in the extraction result, so the blind assignment scheme is affected by these non-terms items and the precision is about 27.30%. The blue-coloured bars (the lighter colour) in Figure 22 show the result of the use of the blind assignment scheme on TermList1 and TermList3. This approach gives the two worst results, compared to our proposed approach to be discussed in Section 6.2.2

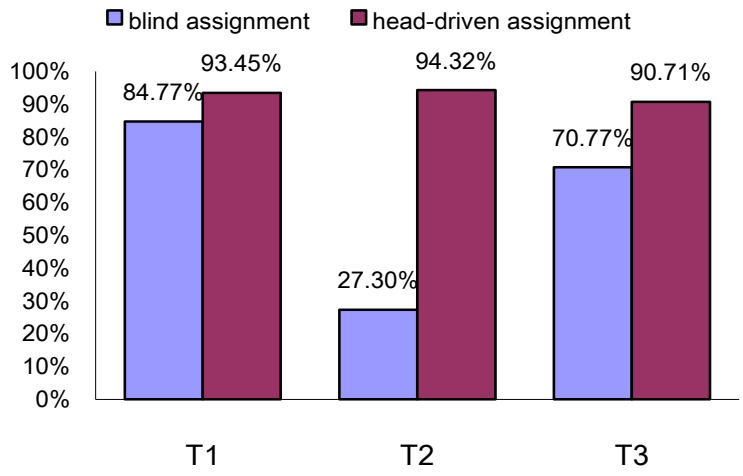

Figure 22 Performance of the Two Assignment Schemes on the Three Term Lists

6.2.2. The Head-Word-Driven Assignment Scheme

The experiment in this section is designed to validate the proposed head-word-driven assignment scheme. The same experiment is conducted on the three term lists respectively, and the results are as shown in Figure 22 in purple (the darker colour). The precision for assigning TPOS tags to TermList1 is 93.45%. Taking the result from a terminology extraction algorithm without regard to its potential for propagating errors, the precision of the head-word-driven assignment scheme for TermList2 is 94.32%. For TermList3, the precision of the POS tag assignment is 90.71%. Compared to the blind assignment scheme, the performance of this algorithm for all three term lists is reasonably good, with a precision of over 90%. It also delivers an improvement of 8.7% and 19.9% for TermList1 and TermList3, respectively, which is reasonably good, and without a heavy cost as compared to the blind assignment scheme. However, there are some abnormalities in these results. Supposedly, TermList1 is a hand-verified term list in the IT domain. Thus, its result should have less noise and it should therefore perform better than TermList2, which is not the case as shown in Figure 22.

6.2.3. Applying Induction Rules

By further analysing the errors, taking TermList1 for example, among these 3,343 terms, about 219 were given improper tags, such as the term “图形学” (Graphics). In this example, two individual words, “图形/n” and “学/v”, form a term. So the output was “图形学/v” for taking the tag of the last segment. This was a wrong tag because

the whole term was a noun. In fact, the error was caused by the general word POS tagging algorithm because, without context, the most likely tagging of "学", a semanteme, is as a verb. This kind of error in the tagging of semantemes appeared in the results of all three term lists, with 169 from TermList1, 29 from TermList2, and 12 from TermList3, respectively. This is a kind of error that can be corrected by applying some simple induction rules. For instance, for all semantemes with multiple tags (including nouns as in the example), the rule can be "*tagging terms with noun suffixes as nouns*". For example, terms "劳改/n 场/q" (reform-through-labour camp) and "计算机/n 图形/n 学/v" (computer graphics) were given different tags using the head-word-driven assignment scheme. They were assigned as: "劳改场/q" and "计算机图形学/v", which can be corrected by this rule. Another kind of mistake involves suffix tags, such as "/ng" (noun suffix) and "/vg"(verb suffix). For example, "知识/n 产权/n 庭/ng" (intellectual property tribunal) and "数据/n 集/vg" (data set) will be tagged as "知识产权庭/ng" and "数据集/vg", respectively, which is obviously wrong. Therefore, the simple rule of "*tagging terms with "/ng" and "/vg" to "/n"*" is applied. The influence of applying the two tuning induction rules on TPOS tagging is shown in Table 19.

Term Lists	Precision of tagging	Precision after adding induction rule	Improvement Percentage
TermList1	93.45%	97.03%	3.83%
TermList2	94.32%	95.41%	1.16%
TermList3	90.71%	93.99%	3.62%

Table 19 Influence of Induction Rules on Different Term Lists

It is obvious that making use of fine-tuning induction rules leads to much better results. In fact, the result for TermList1 reached 97.03%, which is quite close to POS tagging of general domain data. The abnormality also disappeared, as the performance of TermList1 had the best result. The improvement to TermList2 (1.16%) is not as obvious as that for TermList1 and TermList3, which are 3.83% and 3.62%, respectively. This, however, is reasonable, as TermList2 is produced directly from a terminology extraction algorithm using a corpus; thus, the results are noisier.

The result on TermList2 is further analysed to examine the influence of non-term items on this term list. The non-term items are items that are general words or items that cannot be considered as terminology according to the standard answer sheet. For example, neither of the terms “问题” (problem) and “模式训练是” (pattern training is) was considered to be a term because the former is a general word, and the latter should be considered a fragment rather than a word. In fact, in 2,660 items extracted by the algorithm as terminology, only 929 of them are indeed terminology (34.92%), and rest of them are not qualified domain-specific terms. The result of this analysis is listed in Table 20.

	Without Induction Rules		Induction Rules Applied	
	correct terms	precision	correct terms	precision
Terms (929)	879	94.62%	898	96.66%
Non-terms (1,731)	1,630	94.17%	1,640	94.74%
Total (2,660)	2,509	94.32%	2,538	95.41%

Table 20 Data Distribution Analysis on TermList2

The results show that 31 of the 929 correct terms were assigned improper POS tags using the proposed algorithm with the inductions rules; without these rules, 50 were wrongly assigned tags. That is, the precisions for correct data are comparable to those of TermList1 (93.45% and 97.03%, respectively). For the non-terms, 91 items and 101 items from 1,731 items were assigned improper tags with and without the induction rules, respectively. Even though the precisions for terms and non-terms without using the induction rules are almost the same (94.62% vs 94.17%), the improvement for the non-terms using the induction rules is much less impressive than that for the terms. This is the reason for the relatively less impressive performance of the induction rules for TermList2. It is interesting to know that, even though the performance of the terminology extraction algorithm is quite poor, with a precision of only around 35% (929 out of 2,666 terms), this does not have a great on the performance of the TPOS proposed in this paper. This is mainly because the items extracted are still legitimate words, compounds, or phrases, which are not necessarily domain specific.

6.2.4. Discussion

The algorithm proposed in this chapter uses a minimum of resources. No training process is needed for this approach, and even no context information. But the performance of the proposed algorithm is still quite good, and it can be used directly as preparatory work for the construction of a domain ontology because of its precision

rate of over 95%. Other POS tagging algorithms attain good performance in processing general words. For example, a k-nearest-neighbours strategy to identify possible POS tags for Chinese words can reach 90.25% for general word POS tagging (Sun et al. 1997). Another method based on the SVM method on an English corpus can reach 96.9% in POS tagging of known and unknown words (Nakagawa et al. 2001). These results show that the method proposed in this study is comparable in magnitude to these general POS tagging algorithms. Of course, one main reason for this is the difference in objective. The proposed method is for the POS tagging of domain-specific terms, which have much less ambiguity than the tagging of general text. Domain-specific terms are more likely to be nouns and there are some rules in the word-formation patterns, while the general POS tagging algorithms usually requires a training process in which large manually labelled corpora would be involved. The results also show that this simple method can be applied to data in different domains.

6.3. Chapter Summary

In this chapter, a simple but effective method for assigning POS tags to domain-specific terms was presented. This is preliminary work on the classification of terms. This approach needs no training process, and not even context information. Yet a relatively good result is obtained. The method itself is not domain dependent; thus, it is applicable to different domains. The results show that, in certain applications, a simple method may be more effective under similar circumstances.

Chapter 7
Conclusion and Future Works

Considering the purpose of ontology construction, the main objective of this dissertation was to mine concepts and their attributes with the least amount of human intervention. An ontology is constructed using the FCA method based on different kinds of corpus and the experiment results indicate that to achieve the objective of this dissertation, several problems needed to be resolved: For the bottom-up ontology construction method, how to acquire a domain corpus as a resource for the automatic acquisition of concepts; how to acquire concepts and their attributes from instance-rich resources; For the top-down construction of ontology, how to classify domain terms to distinguish them to further map them in to an ontology.

For the first problem, a classification tree was acquired from traversing the category and article pages in Wikipedia using the proposed CT-BFS algorithm. Category labels were considered as edges between Wikipedia pages, and all of the traversed Wikipedia pages were ranked according to both the in-edges and out-edges linking to them. By taking the text part of the top 20,000 Wikipedia pages as a corpus, the precision of the contents belonging to the IT domain reached 92.5%, while for the biology domain it reached 86.5%, which are reliable precisions. The corpora sizes reached 98M and 101M, respectively, which are considerably large sizes for the

extraction and mining of domain information.

For the second problem, there are two aspects. To mine concepts from instance rich corpus, Wikipedia, a method was proposed that explores the annotated information in Wikipedia and combines them by making use of the SVM model to extract concepts based on corresponding instances. The proposed method identified nearly 50,000 concepts for about 700,000 Wikipedia Article pages as instances. It did so with a precision of 85.5%, and covered far more concepts than existing work with a reasonable precision. Another aspect of this problem can be solved by extracting attributes based on both context and instance information in Wikipedia. The relations existing in the contexts of both concept terms and instances were employed to verify the quality of attributes. Three factors were proposed to reflect the quality of attributes in different aspects. They were attribute precision, recall of concepts, and the ratio between attributes and concepts (how many attributes are acquired for each concept). The method based on the FCA model by extracting qualified context words as attributes covers 60% of given concepts and has the highest average A/C ratio, but has a relatively low level of precision with regard to attributes. The method based on instances covers fewer given concepts but can attain a high level of precision of 92.5% and has a relatively stable and reasonable A/C ratio. In addition, it can suggest an attribute type for nearly 80% of acquired attributes, with a precision of 80%.

Then, for the last problem, a simple but effective method for assigning POS tags

to domain-specific concept terms was presented. The method was devised because POS tags contain simple semantic information on classifications for domain terms indicating different concepts. No training process is required for this method, and not even context information. Furthermore, the method itself is not domain dependent; thus, it is applicable to different domains. The precisions on the IT domain and legal domain were 97% and 94%, respectively–comparable even to the precision of the POS tagging of general words. With such precision, the concept terms can be distinguished in the process of building an ontology. In addition, the result indicates that in certain applications, a simple method may be more effective under similar circumstances.

In summary, this dissertation has made the following five major contributions:

It compared general corpus and domain specific corpus for building a domain ontology with the FCA method. Different kinds of context words have been sufficiently discussed and analysed in the process of attribute set selection.

It exploited Wikipedia to acquire a domain-specific corpus by applying a CT-BFS algorithm, which is domain independent and only needs a page to be selected as the start for the traversal.

It mined a large number of concepts with sound precision from Wikipedia by making use of annotated instance information and statistical information in Wiki, and filled them as features into an SVM model to acquire concepts.

It acquires attributes with reasonable precision for concepts by proposing two methods to make use of context information and instance information in Wiki, respectively.

It classified terms by giving them different POS tags for the further mapping of concepts to the ontology. This also provided an effective solution for the POS tagging of Chinese terms in different granularities.

There are also limitations in the work proposed in this dissertation. With regard to the exploitation of corpora, most of the consideration is based on the titles of article pages and category pages. Not enough emphasis has been laid on the contents and internal links of article pages when considering whether or not they are domain relevant. For the classification of terms based on POS tagging, although the fact that no corpus and training are needed is an advantage, statistical information on context and tags has also been neglected. For the selection and acquisition of attributes, information on the context and instances of a given concept term has been taken into account individually, and each kind of information has its weakness in precision and coverage. For concept mining, not enough features have been explored to apply the SVM model.

Therefore, possible improvements can be classified into three major directions for future study:

To explore and employ more resources in Wiki, such as the internal links in

article pages, the redirect information for different names pointing to the same article, and the disambiguation pages that can be mined, which can be of benefit to corpus selection, attribute acquisition, and concept mining.

To make more use of statistical information about the contexts of concepts, such as statistics on co-occurrence, POS tags, and relative positions, which can be of benefit for term classification, attribute acquisition, and concept mining.

To seek appropriate combinations of existing information and resources, such as information on context and instances mixed with concepts, which can be of benefit in the acquisition of attributes and concepts.

Appendix 1

QA 76.	Computer Science
QA 76.625.	Internet Programming
QA 76.73.	Programming Languages
QA 76.73.J38.	Java
QA 76.73.J39.	JavaScript
QA 76.76.	Computer Software (special topics)
QA 76.76.H94.	Hypertext. HTML
QA 76.76.H94.	RSS
QA 76.76.O63.	Operating Systems. Unix
QA 76.9.	Computer Science(other topics)
QA 76.9.D3.	Databases
QA 76.9.W43.	Web Databases
TK.	Electrical Engineering
TK 5105.	Telecommunications: Data Transmission Systems
TK 5105.565.	CGI
TK 5105.73.	Electronic Mail Systems
TK 5105.875.	Special Networks and Systems
TK 5105.875.I57.	Internet
TK 5105.875.U83.	Usenet
TK 5105.882.	Browsers
TK 5105.8835.	Internet domain names
TK 5105.884.	Search engines
TK 5105.886.	Internet Relay Chat
TK 5105.888.	World Wide Web
TK 5105.8882.	Wikis
TK 5105.8884.	Weblogs
ZA.	Information Resources (general)
ZA 4080.	Digital Libraries
ZA 4201.	Internet
ZA 4226.	World Wide Web
ZA 4480.	Electronic Discussion Groups
ZA 4550.	Video Recordings

Appendix 2

TK1-9971	Electrical engineering. Electronics.
TK301-399	Electric meters
TK452-454.4	Electric circuits. Electric networks
TK1001-1841	Powerplants. Central stations
TK2000-2891	Dynamoelectric machinery and auxiliaries Generators, Motors, Transformers
TK4125-4399	Electric lighting
TK4601-4661	Electric heating
TK5101-6720	Telecommunication Telegraphy, telephone, radio, radar, television
TK 5105.	Data Transmission Systems
TK 5105.565.	CGI
TK 5105.73.	Electronic Mail Systems
TK 5105.875.	Special Networks and Systems
TK 5105.875.I57.	Internet
TK 5105.875.U83.	Usenet
TK 5105.882.	Browsers
TK 5105.8835.	Internet domain names
TK 5105.884.	Search engines
TK 5105.886.	Internet Relay Chat
TK 5105.888.	World Wide Web
TK 5105.8882.	Wikis
TK 5105.8884.	Weblogs
TK7800-8360	Electronics
TK7885-7895	Computer engineering. Computer hardware
TK8300-8360	Photoelectronic devices (General)

Appendix 3

Terms		
ASCII	CPU	编程(programming)
操作系统(operating system)	存储器(storage)	存取(storing)
大型机(mainframe computer)	单板机(Single Board Computer)	电脑(computer)
电子商务(e-business)	电子邮件(email)	调制解调器(modem)
读取(reading)	服务器(server)	工作站(workstation)
光标(cursor)	硅谷(Silicon Valley)	缓存(cache)
回车(return)	寄存器(register)	计算机(computer)
计算机辅助(computer aided)	计算机化(computerization)	计算中心(computing centre)
监视器(monitor)	兼容机(compatible machine)	键盘(keyboard)
解码(decoding)	空格(space)	联机(online)
模式识别(pattern recognition)	内存(memory)	屏幕(screen)
软硬件(hardware and software)	数字计算机(digital computer)	图标(icon)
微处理机(microcomputer)	微处理器(microprocessor)	微电脑(microcomputer)
微机(microcomputer)	微型机(microcomputer)	硬磁盘(hard disk)
硬盘(hard disk)	中央处理器(CPU)	终端(terminal)
终端机(terminal)	主板(mainboard)	字节(byte)
总线(bus)		

Bibliography

Almuhareb, A. and Poesio, M., 2004. *Attribute-Based and Value-Based Clustering: An Evaluation*. In the Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP), Barcelona, Spain.

Almuhareb, A. and Poesio, M., 2005. *Concept Learning and Categorization from the Web*. In the Proceedings of the 27th Annual Cognitive Science Conference, Stresa, Italy, July 21-23.

Baker, C.F., Fillmore, C.J., and Lowe, J.B., 1998. *The Berkeley FrameNet Project*. In the Proceedings of COLING/ACL 98, pp. 86-90.

Bellomi, F. and Bonato, R., 2005. *Network Analysis for Wikipedia*. In the Proceedings of Wikimania.

Berwick, R.C., Abney, S.P., and Tenny, C., 1991. *Principle-Based Parsing: Computation and Psycholinguistics*. Kluwer Academic Publishers.

Brants, T., 2000. *TnT: A Statistical Part-of-Speech Tagger*. In the Proceedings of ANLP-2000, the 6th Applied Natural Language Processing Conference, pp. 224-231.

Bruner, J.S., Goodnow, J.J., and Austin, G.A., 1967. *A Study of Thinking*. New York: Science Editions.

Chandrasekaran, B., Josephson, J.R., and Benjamins, V.R., 1999. *What Are Ontologies, and Why Do We Need Them?* IEEE Intelligent Systems, Vol. 14,

Issue 1, pp. 20-26.

Chen, Y.R., Lu, Q., Li, W.J., Li, W.Y., Ji, L.N., and Cui, G.Y., 2007. *Automatic Construction of a Chinese Core Ontology from an English-Chinese Term Bank*. In the Proceedings of the ISWC2007 Workshop OntoLex07 - From Text to Knowledge: The Lexicon/Ontology Interface, Busan, Korea, pp. 78-87.

Chernov, S., Iofciu, T., Nejdl, W., and Zhou, X., 2006. *Extracting Semantic Relationships between Wikipedia Categories*. In the Proceedings of SemiWikipedia 2006 at the ESWC.

Cimiano, P., Staab, S., and Tane, J., 2003. *Automatic Acquisition of Taxonomies from Text: FCA Meets NLP*. In the Proceedings of the ATEM-2003, the PKDD/ECML'03 International Workshop on Adaptive Text Extraction and Mining. Cavtat-Dubrovnik, Croatia, Sep. 22, pp. 10-17.

Cimiano, P., Handschuh, S., and Staab, S., 2004. *Towards the Self Annotating Web*. In the Proceedings of the 13th International Conference on the World Wide Web, New York, U.S.A., May 17-20, pp. 462-471.

Denoyer, L. and Gallinari, P., 2006. *The Wikipedia XML Corpus*. ACM SIGIR Forum, Vol. 40, No. 1, June.

Dowty, D., Wall, R., and Peters, S., 1980. *Introduction to Montague Semantics*. Springer.

Gelfand, B., Wulfekuhler, M., and Punch, W.F., 1998. *Automated Concept Extraction*

From Plain Text. In the Proceedings of the AAAI-98 Workshop on Learning for Text Categorization, Technical Report WS-98-05, pp. 13-17.

Fleischman, M., Hovy, E., and Echihabi, A., 2003. *Offline strategies for online question answering: Answering questions before they are asked*. In the Proceedings of the Association for Computational Linguistics (ACL2003), 2003, pp. 1-7.

Gregorowicz, A. and Kramer, M.A., 2006. *Mining a Large-Scale Term-Concept Network from Wikipedia*, Technical Report #06-1028, The MITRE Corp., October.

Gruber, T.R., 1995. Toward Principles for the Design of Ontologies Used for Knowledge Sharing. In the Proceedings of the International Journal of Human and Computer Studies, 43(5/6), pp. 907-928.

Haav, H.M., 2003. *An Application of Inductive Concept Analysis to Construction of Domain-specific Ontologies*. In the Proceedings of the Pre-conferenceVLDB2003, the 29th International Conference on Very Large Data Bases, Los Altos, September 9-12, pp. 63-67.

He, Y., Sui, Z.F., Duan, H.M., Yu, S.W., 2006. *Term Mining Combining Term Component Bank*. Computer Engineeing and Applications. Vol. 42, No. 33.

Heylighen, F., 1995. *Ontology, introduction*. Referencing the page in Principia Cybernetica Web, http://pespmc1.vub.ac.be/ONTOLI.html.

Ji, L.N., Lu, Q., Li, W.J., and Chen, Y.R., 2007a. Automatic Construction of a Core Lexicon for Specific Domain. In the Proceedings of the Sixth International Conference on Advanced Language Processing and Web Information (ALPIT 2007), Luoyang, China, Aug. 22-24.

Ji, L.N., Sum, M., Lu, Q., Li, W.J., and Chen, Y.R., 2007b. *Chinese Terminology Extraction Using Window-Based Contextual Information*. CICLing 2007, LNCS 4394, the 8th Conference on Intelligent Text Processing and Computational Linguistics, Mexico City, Mexico, February 18-24, pp. 62-74.

Jiang, G.Q., Ogasawarab, K., Endoha, A., and Sakuraia, T., 2003. *Context-based ontology building support for clinical domains using formal concept analysis*. International Journal of Medical Informatics, 2003, pp. 71-81.

Khan, L. and Luo, F., 2002. *Ontology Construction for Information Selection*. In the Proceedings of the International Conference on Tools with Artificial Intelligence, Washington, U.S.A., Nov. 4-6.

Li, S.J., Lu, Q., and Li, W.J., 2005. Experiments of Ontology Construction with Formal Concept Analysis. In the Proceedings of OntoLex2005, the 4th workshop of Ontologies and Lexical Resources, Jeju Island, South Korea, October 15, pp. 67-75.

Lih, A., 2004. *Wikipedia as Participatory Journalism. Reliable Sources? Metrics for evaluating collaborative media as a news resource*. Symposium. In the

Proceedings of the International Symposium on Online Journalism.

Liu, B., Chin, C.W., and Ng, H.T., 2003. *Mining Topic-Specific Concepts and Definitions on the Web*. In the Proceedings of WWW 2003, Budapest, Hungary, May 23-24.

Maedche, A. and Staab, S., 2001. *Ontology Learning for the Semantic Web*, IEEE Intelligent Systems: Special Issue on the Semantic Web, Vol. 16, pp. 72-79.

Margolis, E. and Laurence, S., 2007. The Ontology of Concepts-Abstract Objects or Mental Representations? In Nous Electronic Resource, Vol. 41, No. 4, pp. 561-593. Blackwell Publishing.

McCarthy, J., 1980. *Circumscription – A Form of Non-Monotonic Reasoning*, In Artificial Intelligence, Vol. 5, No. 13, pp. 27-39.

Mikheev, A., 1996. *Unsupervised Learning of Word-Category Guessing Rules*. In the Proceedings of ACL'96, the 34th meeting of the Association for Computational Linguistics, Santa Cruz, June.

Mikheev, A., 1997. *Automatic Rule Induction for Un-known Word Guessing*. Computational Linguistics, Vol. 23, No. 3 , pp. 405-423.

Miller, G.A., Beckwith, R., Fellbaum, C., Gross, D., and Miller, K., 1990. *Introduction to WordNet: An Online Lexical Database*. International Journal of Lexicography, 3: pp. 235-312.

Missikoff, M., Navigli, R., and Velardi, P., 2002. *Integrated Approach to Web*

Ontology Learning and Engineering. In IEEE Computer, Vol. 35, Issue 11, pp. 60-63.

Nakagawa, T., Kudoh, T., and Matsumoto, Y., 2001. *Unknown Word Guessing and Part-of-Speech Tagging Using Support Vector Machines*. In the Proceedings of NLPPRS2001, the 6th Natural Language Processing Pacific Rim Symposium, Tokyo, Japan, November 27-30, pp. 325-331.

Navigli, R. and Velardi, P., 2004. *Learning Domain Ontologies from Document Warehouses and Dedicated Web Sites*, Computational Linguistics, MIT Press.

Niles, I. and Pease, A., 2001. *Towards a Standard Upper Ontology*. In the Proceedings of the 2nd International Conference on Formal Ontology in Information Systems (FOIS-2001), Ogunquit, Maine, U.S.A., October 17-19.

Obitko, M., Snasel, V., and Smid, J., 2004. *Ontology Design with Formal Concept Analysis*. In the Proceedings of the CLA 2004 Workshop: Concept Lattices and Their Applications Workshop, Czech Republic, September, pp. 111-119.

Orphanos, G. and Christodoulakis, D., 1999. *POS Disambiguation and Unknown Word Guessing with Decision Trees*. In the Proceedings of EACL'99, the 9th Conference of the European Chapter of the Association for Computational Linguistics, Bergen, June 8 - 12, pp. 134-141.

Pasca, M. and Van Durme, B., 2007. *What you seek is what you get: Extraction of class attributes from query logs*. In the Proceedings of IJCA-07, the 20th

International Joint Conference on Artificial Intelligence, Hyderabad, India, January, pp. 2832-2837.

Poesio, M. and Almuhareb, A., 2005. *Identifying Concept Attributes Using a Classifier*. In the Proceedings of the ACL-SIGLEX 2005 Workshop on Deep Lexical Acquisition.

Samuelsson, C., 1993. *Morphological Tagging Based Entirely on Bayesian Inference*. In the Proceedings of NCCL1993, the 9th Nordic Conference on Computational Linguistics, Stockholm, Sweden, June 3-5.

Sun, M.S., Shen, D.Y., and Huang, C.N., 1997. *CSeg& Tag1.0: a practical word segmenter and POS tagger for Chinese texts*. In the Proceedings of the 5th Conference on Applied Natural Language Processing.

Smith, B. and Welty, C. A., 2001. *Ontology: Towards a new synthesis*. Formal Ontology in Information Systems (FOIS'01), Ogunquit, Maine, U.S.A., Oct. 17-19.

Strube, M. and Ponzetto, S.P., 2006. *WikiRelate! Computing Semantic Relatedness Using Wikipedia*. In the Proceedings of the AAAI.

Stumme, G. and Maedche, A., 2001. *FCA-Merge: bottom-up merging of ontologies*. In the Proceedings of IJCAI'01, the 7th International Conference on Artificial Intelligence, Seattle, WA, USA, pp. 225-230.

Sumida, A., Torisawa, K., and Shinzato, K., 2006. *Concept-Instance Relation*

Extraction from Simple Noun Sequence Using a Full-Text Search Engine. In the Proceedings of the ISWC 2006 workshop on Web Content Mining with Human Language Technologies (WebConMine), Athens, Greece.

Toutanova, K., Klein, D., Manning, C., and Singer, Y., 2003. *Feature-Rich Part-of-Speech Tagging with a Cyclic Dependency Network*. In the Proceedings of HLT/NAACL2003, the Human Language Technology conference / North American chapter of the Association for Computational Linguistics annual meeting, Edmonton, Canada, May 27-June 1.

Tseng, H., Jurafsky, D., and Manning, C., 2005. *Morphological Features Help POS Tagging of Unknown Words across Language Varieties*. In the Proceedings of the 4th SIGHAN Workshop on Chinese Language Processing.

Tsuruoka, Y. and Tsujii, J., 2005. *Bidirectional Inference with the Easiest-First Strategy for Tagging Sequence Data*. In the Proceedings of the Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing (HLT/EMNLP), Vancouver, October, pp. 467-474.

Uschold, M. and Gruninger, M., 1996. *Ontologies: Principles, methods and applications*. Knowledge Engineering Review, Vol. 11, No. 2.

Van Rees, R., 2003. *Clarity in the Usage of the Terms Ontology, Taxonomy and Classification*. In the Proceedings of the 2003 CIB w78 conference in Auckland, New Zealand.

Voss, J., 2005. *Measuring Wikipedia*. In the Proceedings of the 10th International Conference of the ISSI.

Völkel, M., Krötzsch, M., Vrandecic, D., Haller, H., and Studer, R., 2006. *Semantic Wikipedia*. In the Proceedings of the 15th International Conference on the World Wide Web.

Wang, H. *Statistical studies on Chinese vocabulary* (汉语词汇统计研究). http://www.huayuqiao.org/articles/wanghui/wanghui06.doc. The date of publication is unknown from the online source. Last checked: 2010-01-03.

Wu, Y.F., Chang, B.B., and Zhan, W.D., 2003. *Building a Chinese-English Bilingual Phrase Database*, Terminology Standardization and Information Technology, Vol. 4, pp. 41-45.

Xu, R.F., Lu, Q., Li, Y., and Li, W.Y., 2005. *The Design and Construction of the PolyU Shallow Tree-bank*. International Journal of Computational Linguistics and Chinese Language Processing, Vol. 10, No. 3.

Yoshinaga, N. and Torisawa, K., 2007. *Open-Domain Attribute-Value Acquisition from Semi-Structured Texts*. In the Proceedings of the OntoLex07 - From Text to Knowledge: The Lexicon/Ontology Interface, Busan, South Korea, November 11.

Zhou, L., 2007. *Ontology Learning: State of the Art and Open Issues*, Information Technology and Management, Vol. 8, No. 3, pp. 241-252.