2007

Knowledge-based methods for automatic extraction of domain-specific ontologies

Janardhana R. Punuru
Louisiana State University and Agricultural and Mechanical College

KNOWLEDGE-BASED METHODS FOR AUTOMATIC EXTRACTION OF DOMAIN-SPECIFIC ONTOLOGIES

A Dissertation

Submitted to the Graduate Faculty of the
Louisiana State University and
Agricultural and Mechanical College
in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy

in

The Department of Computer Science

by
Janardhana R. Punuru
B.Tech., Sri Krishna Devaraya University, 1998
M.S., Louisiana State University, 2002
May, 2007

Dedication

To my wife Sandhya Reddy, my parents Tulasamma and Nagi Reddy, and my brother Konda Reddy and his family.

Acknowledgments

Let me first express deepest appreciation and gratitude to my major adviser Dr. Jianhua Chen, for her inspiration, encouragement, patience, mentorship, and guidance. I am grateful for her assistance and motivation throughout this research. Without her assistance this dissertation would not be possible.

I also very much appreciate the advice and helpful comments of the advisory committee, Dr. Sukhamay Kundu, Dr. Donald H. Kraft, Dr. Guoli Ding, and Dr. Edward F. Watson. Thanks are due to the Department of Computer Science for the financial support received in the form of assistantship that allowed me to focus on the development of this research.

I would like to thank Dr. Ralph W. Pike professor in Chemical Engineering for his encouragement and moral support throughout my studies.

Finally, I would like to acknowledge the support and encouragement of relatives and friends who helped me directly or indirectly during the course of this research.

Table of Contents

List of Tables

List of Figures

Abstract

Semantic web technology aims at developing methodologies for representing large amount of knowledge in web accessible form. The semantics of knowledge should be easy to interpret and understand by computer programs, so that sharing and utilizing knowledge across the Web would be possible. Domain specific ontologies form the basis for knowledge representation in the semantic web. Research on automated development of ontologies from texts has become increasingly important because manual construction of ontologies is labor intensive and costly, and, at the same time, large amount of texts for individual domains is already available in electronic form. However, automatic extraction of domain specific ontologies is challenging due to the unstructured nature of texts and inherent semantic ambiguities in natural language. Moreover, the large size of texts to be processed renders full-fledged natural language processing methods infeasible.

In this dissertation, we develop a set of knowledge-based techniques for automatic extraction of ontological components (concepts, taxonomic and non-taxonomic relations) from domain texts. The proposed methods combine information retrieval metrics, lexical knowledge-base(like WordNet), machine learning techniques, heuristics, and statistical approaches to meet the challenge of the task. These methods are domain-independent and automatic approaches.

For extraction of concepts, the proposed WNSCA+{PE, POP} method utilizes the lexical knowledge base WordNet to improve precision and recall over the traditional information retrieval metrics. A WordNet-based approach, the compound term heuristic, and a supervised learning approach are developed for taxonomy extraction. We also developed a weighted word-sense disambiguation method for use with the WordNet-based approach. An unsupervised approach using log-likelihood ratios is proposed for extracting non-taxonomic relations. Further more, a supervised approach is investigated to learn the semantic constraints for identifying relations from prepositional phrases. The proposed methods are validated by experiments with the *Electronic Voting* and the *Tender Offers, Mergers, and Acquisitions* domain corpus. Experimental results and comparisons with some existing approaches clearly indicate the superiority of our methods.

In summary, a good combination of information retrieval, lexical knowledge base, statistics and machine learning methods in this study has led to the techniques efficient and effective for extracting ontological components automatically.

Chapter 1

Introduction

Ontologies are widely used in Artificial Intelligence, Knowledge Engineering, Knowledge Management, Natural Language Processing, Information Retrieval, and Intelligent Information Integration fields. The importance of ontologies has re-emerged with the proposal of semantic web [Berners-Lee et al., 2001] by Tim Berners Lee. Originally, ontologies are used in closed domains such as molecular biology, bioinformatics, etc to assist in knowledge management, knowledge engineering, question answering, information extraction, and text summarization systems.

1.1 A Brief History of Ontologies

Initially, the term ontology originated from philosophy when the ancient people concerned about the difficulties encountered when they tried to find the essence of things through the changes. In philosophy, ontology means systematic explanation of being. The term ontology means the *philosophy of being* was coined in 17^{th} century. Here Onto means *being* and logos means *treatise*. At the end of of 20^{th} century, ontologies have emerged as a research area in computer science. The definition of ontology has changed over the years. Various definitions are presented in the literature for ontology. The following seven distinct definitions are collected and analyzed in [Gómes-Pérez et al., 2004] and [Guarino and Giaretta, 1995].

1. Ontology as Philosophical discipline.
2. Ontology as in informal conceptual system.
3. Ontology as a formal semantic account.
4. Ontology as a specification of conceptualization system.
5. Ontology as a representation of a conceptual system characterized by specific formal properties and only by its specific purposes.
6. Ontology as the vocabulary used by logical theory.
7. Ontology as a specification of logical theory.

1.2 Ontology and Its Usefulness

Even though various definitions exists for ontology in the literature, two of the notions motivating the ontology extraction task are as follows. According to Webster's dictionary, ontology is a particular theory of the nature of being or the kind of existent. In [Gruber, 1993], ontology is defined as an explicit specification of a conceptualization. From these two definitions, conceptualization consists of the existents and their characteristics. Ontology is interpreted as the formal representation of the conceptualization. As an analogy, one can describe the ontology of a domain as the relational schema of a database. Relational schema represents both the entities(concepts) and the dependency relations between the entities whereas the ontology consists of concepts and semantic relations between the concepts.

Anatomy of Ontologies In general, ontology of a domain consists of four major components listed below.

- **Concepts:** Concepts of a domain are an abstract or concrete entities derived from specific instances or occurrences.
- **Attributes:** Attributes are characteristics of the concepts which may or may not be concepts by themselves.
- **Taxonomy:** Taxonomy provides hierarchical relations between the concepts.
- **Non-taxonomic Relations:** Non-taxonomic relations specify non-hierarchical semantic relationships between the concepts.

Along with the above four components, ontologies may also consist of instances for each of the concepts, and inference rules of domain. This dissertation emphasizes concepts, taxonomic relations, and non-taxonomic relations only. Detailed discussion of ontological components and existing techniques to extract each of the components are presented in chapter 2.

1.2.1 Examples of Ontologies

A wide variety of communities developed either general purpose or domain specific ontologies for various domains. Among various ontologies, CYC [Lenant, 1990] and WordNet [Miller, 1990] are the two most popular ontologies. The CYC project started in early 90's with the aim of formal representation of the human knowledge, which includes facts, rules of thumb, and heuristics for reasoning about the objects and events of every day life. WordNet is a lexical reference system which consists of synonym sets of English nouns, verbs, adjectives, adverbs as lexical concepts and semantic relations between the concepts. Both CYC and WordNet are general purpose ontologies. Along with CYC and WordNet, there exists a large number of domain specific ontologies also. Some of the examples are Gene Ontology for genomics, Sequence Ontology for biological sequences, Plant Ontology for plant structures, and Unified Medical Language System(UMLS) for biomedicine and health.

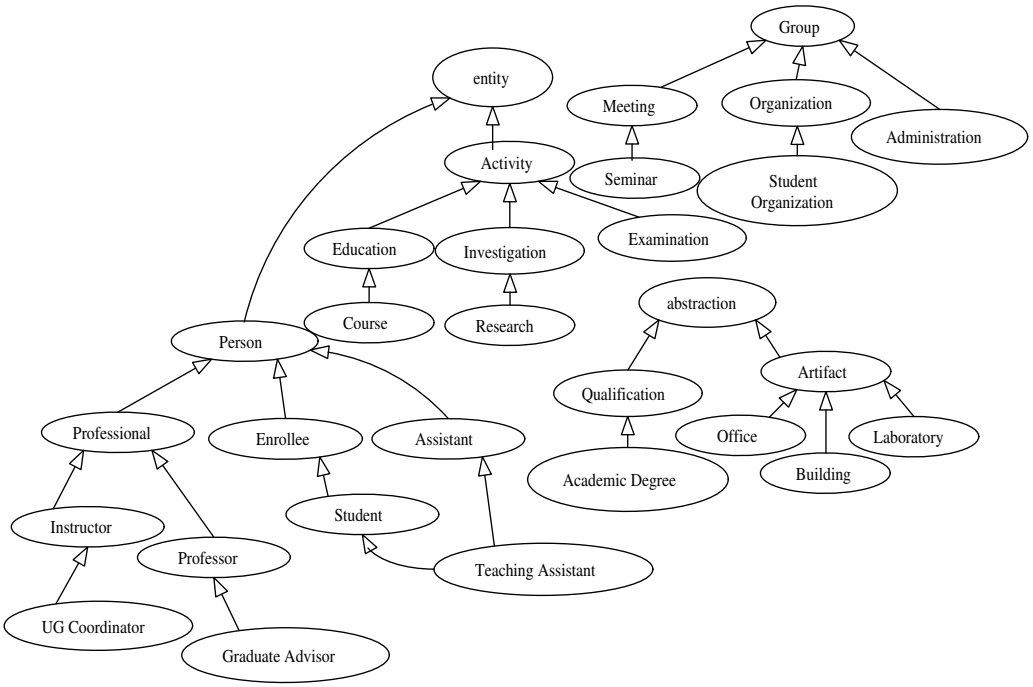

Figure 1.1: Partial Ontology of Computer Science Department

To further illustrate, a portion of the ontology for computer science Department is shown in Figure 1.1. The example ontology in Figure 1.1 indicates some of the major concepts and their hierarchical relations only. In general, the ontology includes non-taxonomic relations as listed below.

`Student→take→ Course`
`Professor→teach→Course`
`Professor→consistsOf→ Office`

1.2.2 Usefulness of Ontologies

Ontologies possess a wide variety of applications in knowledge management, information retrieval, information extraction, question answering systems, and artificial intelligence. In particular, ontologies are useful to share the common understanding of the domain between agents, to enable the reuse of knowledge, to make domain assumptions explicit, and to analyze the domain knowledge. As mentioned before, ontologies are the basis for semantic web. To make the semantic web dream into reality, annotation of web pages with ontological information is necessary.

A brief description on the role of ontologies in semantic web is as follows. One of the applications of semantic web is replacement of key word based web search with the knowledge level querying. That is, at present search technologies retrieve web pages arranged with efficient page ranking algorithms consisting of key words of the user query. In this scenario, the user has to read all the web pages retrieved to

find the answers to the user queries. Whereas in semantic web, each of the websites is annotated with ontologies. Hence the whole web consists of agglomerations of domain-specific ontologies. In semantic web, the user query is analyzed at knowledge level and will be answered by performing logical inferencing using ontologies.

Considering the above notion of semantic web, various components are involved in realizing the semantic web applications. Few of them are as follows:

1. Languages for representation of ontologies.
2. Web scalable algorithms for logical inferencing.
3. Acceptability of communities(either users or businesses) for change.
4. Creation of domain specific ontologies.

As part of the language standards, various meta languages such as XML, RDF, and OWL etc are developed for encoding ontologies of the domain. Several algorithms or techniques for merging or querying ontologies are developed and/or in research based on Description Logic. The most difficult issue is to make users accept the semantic web technology. The fourth component is creation of ontologies. It is required to represent websites or domain texts in terms of ontologies using one of the ontology languages. This dissertation concentrates on methods for automatic extraction of ontologies.

1.3 Development of Ontologies

Even though there exists a wide variety of applications of ontologies, as of now ontologies for various domains are developed manually. Some of the issues involved in the design and development of ontologies are the requirement of expert knowledge of the domain, extensive group discussions in understanding the view point of the domain, and incremental modifications to the ontology. For example, as mentioned in [Sabou et al., 2005], building the initial ontology for ^{my}Grid project, in bioinformatics domain, took two months for an ontology expert with four years of experience in building the description logic based biomedical ontologies. In general, the construction of ontologies require the steps similar to steps involved in software development life cycle. But as in software development there are no such standards(or methods) established for the development of ontologies. Because the lack of standards in ontology development, manual construction of ontologies is costly both in time and labor.

In computing literature, various approaches or guidelines are presented for manual construction of ontologies. Several tools such as Ontolingua [Farquhar et al., 1997], OilEd [Bechhofer et al., 2001], Protege [Protege, 2001], and OntoEdit [Sure et al., 2002] are developed for the construction and management of ontologies. Most prominent of these are Protege and OntoEdit. The main objective of these tools is to assist the domain expert in the construction of ontologies.

To reduce the effort in design and development of ontologies, this dissertation research develops a set of methods for automatic extraction of each of the ontological components from domain texts. The methods presented here are domain independent and extract each of the components automatically. The presented methods make use of text processing techniques and knowledge acquisition methods. Techniques depend on text processing and knowledge acquisition methods must address the following constraints.

1.3.1 Constraints in Text Processing

Natural language texts are not only unstructured but also ambiguous in word meanings and usage. Because of the unstructuredness and ambiguousness, it is difficult to perform semantic analysis of natural language texts. Methods for ontology extraction methods must address the following issues with respect to text processing.

- Unstructured text.
- Ambiguity in English text.
- Lack of closed domain of lexical categories.
- Noisy text.
- Requirement of very large texts.
- Lack of standards in text processing.

Unstructured Text Even though the natural language processing research has its origins from the early 50's, because of the unstructuredness of the text, there exists no fixed schemata for interpreting the natural language statements. Hence it is difficult to convert unstructured texts into a structured representation which is required for computer processing systems.

Ambiguity in English Text In addition to the unstructuredness, natural language text is also ambiguous. The meaning of a word varies based on the context in which it occurs. In natural language processing, context in which a word can occur is defined as the "sense" of the word. In English language, most of the nouns consists of more than one sense. For example, the word "form" has sixteen different senses. A word not only consists of multiple senses but also appears in multiple parts of speech. For example, the word "like" can occur in eight distinct parts of speech [Chris, 2006] as shown below.

verb	"Fruit flies *like* a banana."
noun	"We may never see its *like* again."
adjective	"People of *like* tastes agree."
adverb	"The rate is more *like* 12 percent."
preposition	"Time flies *like* an arrow."

conjunction	"They acted *like* they were scared."
interjection	"*Like*, man, that was far out."
verbal auxiliary	"So loud I *like* to fell out of bed."

Lack of Closed Domain of Lexical Categories The possible lists of pronouns, prepositions, conjunctions, and interjections are fixed. These four parts of speech are called as functional or closed categories. Elements of the remaining parts of speech such as nouns, verbs, adjectives, and adverbs are not fixed. These are called lexical or open categories. New words are added to the English dictionaries from other languages or some other sources. Variations of the nouns are used as adjectives, verbs, and adverbs. Because of the lack of closed sets for lexical categories, it is very difficult to identify the validity of the extracted terms automatically.

Noisy Text When large amounts of text is collected for processing, there exists a very high possibility for the presence of noise in the collected text. It is difficult to identify and filter such noisy text without knowing the content of the whole text. For example, texts may contain analogies and metaphors which are not relevant to the domain text.

Requirement of Large Texts Most of the existing techniques for text analysis are based on the repetition of information or contexts. Hence, to make use of the such techniques, the input text should be very large. Further more, if the text is not large enough, the coincidental occurrences may dominate the text. It may lead to the extraction of incorrect semantic information from texts.

Lack of Standards in Text Processing In general, text documents written by various authors convey different perspectives of the domain. Hence all the documents may not represent the same view of the domain. Even all documents may belong to a single domain, some of them may support and others may oppose a view point. It is very difficult to identify the collections of text which represent the single view of the domain. No standards are established in identifying the perspective (Example: supports or opposes an idea) of a document.

1.3.2 Constraints in Knowledge Acquisition

There exists various issues for automatic knowledge extraction from the English text. Some of them are described below.

Lack of Standards in Knowledge Representation Because of the lack of standards in ontology design, different research communities use different root-level hierarchies for their ontology designs. Root-level hierarchy vary with respect to the domain. Various top-level hierarchies for different ontology projects are described in [Noy and Hafner, 1997]. Hence, it is difficult to choose a fixed hierarchy as the root-level hierarchy for all the ontologies. For example, In WordNet, there is no fixed

root-level concept. Even the authors claimed that nobody agrees if any such thing as *entity* is mentioned as the root concept. Nouns in the WordNet are classified into 25 unique classes at the root-level. These 25 classes are called as unique beginners. Description of WordNet and its unique beginners is presented in Appendix B. Recently, IEEE formed a work group named IEEE SUMO(Standard Upper Merged Ontology) to standardize the root-level hierarchy [Schoening, 2003].

Lack of Fully Automatic Methods for Knowledge Acquisition Because of the unstructuredness and ambiguity in texts, no fixed procedures exist for the analysis of text. Many of the existing techniques for knowledge acquisition from texts are domain dependent and/or based on supervised learning methods. Supervised learning methods require large amounts of training data for each of the domains. Since the ontology represents the knowledge of the whole domain, it is difficult to have large amounts of the data for training to build the ontology and additional data for ontology extraction. Ideally, the techniques for ontology extraction should be domain independent and should not rely on large amounts of training data.

Lack of Techniques for Coverage of Whole Texts Most of the current approaches in the literature are developed with the aim of building or extending thesaurus from the texts. Existing approaches are based on the word frequencies, co-occurrence statistics, and syntactic-patterns. These approaches cover only terms or sentences which satisfy above constraints. The remaining text is ignored. But most of the ignored text also contains useful knowledge about the domain. It is desirable to develop the techniques which covers the most of text possible.

1.4 Problems and Approaches

Because of the several constraints mentioned above, automatic ontology extraction is a difficult task. To reduce the effort in the construction of ontologies, a set of methods are developed to extract each of the ontological components. In this dissertation, the problem of automatic acquisition of ontologies has been divided into three different tasks listed below.

1. Acquisition of Concepts
2. Taxonomy Extraction
3. Non-taxonomic Relations Extraction

To solve each of the tasks listed, we proposed a set of techniques based on contextual heuristics, lexico-syntactic patterns, information retrieval metrics, lexical knowledge bases, machine learning techniques, and statistical based methods. Among the three tasks, output of task 1 is used by tasks 2 and 3. Taxonomic relations and non-taxonomic relations are extracted for the concepts obtained in task 1. To increase the modularity and to avoid error carry over, acquisition of ontological components

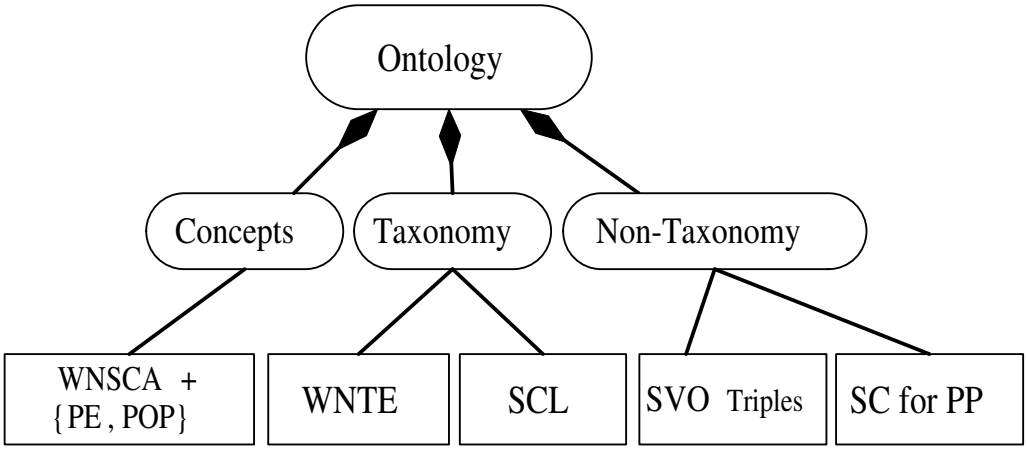

Figure 1.2: Ontology Extraction Components

problem is tackled by solving each of the tasks independently. The set of methods developed for extraction of ontological components is shown in the Figure 1.2.

Acquisition of Concepts As shown in the Figure 1.2, the WNSCA+{PE, POP} method is developed to extract the concepts. The WNSCA+{PE, POP} utilizes information retrieval metrics such as raw frequency counting and tf.idf along with senses information extracted from the lexical knowledge base.

Taxonomy Extraction To find the taxonomic relations between the concepts, we proposed three different methods. The initial method extracts the hierarchical relations from knowledge base by finding contextual senses of the concepts. Another method we proposed finds the taxonomy relations using compound term heuristic. Finally, The other method investigated is identification of semantic classes of the concepts using naive Bayes classifier. The knowledge based approach for taxonomy extraction is named as WNTE and semantic class identification method is named as SCL in the Figure 1.2.

Non-Taxonomic Relations Extraction A statistical method based on log-likelihood ratios is developed for finding non-taxonomic relations. In addition, a supervised learning algorithm, using semantic classes of the concepts, is developed to label the semantic relations between the concepts based on their occurrence in prepositional phrases. In Figure 1.2, log-likelihood ratios method is named as the SVO Triples method and the learning method for finding relations from prepositional phrases is named as SC for PP.

The methods listed above utilize lexical annotation tools such as tagger and chunker, WordNet, sense disambiguation, term heuristics, naive Bayes classifier and C4.5 machine learning algorithms, and information retrieval metrics namely raw frequency counting and tf.idf. All the methods are developed and experimented with *Electronic*

Voting domain and *Tender Offers, Mergers, and Acquisitions* domain texts. Accuracy results for each of the methods are reported.

The remainder of the dissertation is organized as follows. Chapter 2 explains each of the sub tasks and provides a detailed discussion on the existing approaches and their shortcomings. The proposed methods for concept extraction task and their experimental results are presented in chapter 3. Chapter 4 presents the methods developed for finding taxonomic relations between the concepts extracted. Similarly, chapter 5 describes the methods proposed for finding non-taxonomic relations between the concepts extracted and the experimental results. Chapter 6 provides a detailed discussion of each of the methods developed. Finally, conclusions and future directions are presented in chapter 7.

1.5 Summary

Ontology of a domain consists of concepts, taxonomic relations and non-taxonomic relations between the concepts. Ontologies consisting of above units are widely used in information retrieval, artificial intelligence, and intelligent information integration tasks. The importance of ontologies has re-emerged with the proposal of semantic web. Even though ontologies posses a variety of applications, as of now ontologies are developed manually. But manual construction of ontologies is costly both in time and labor. To reduce the effort in manual construction of ontologies, this dissertation research presents a set of techniques for extracting each of the ontological components.

The presented techniques combine information retrieval metrics, lexical knowledge base, machine learning techniques, statistical methods, and term heuristics. For extraction of concepts, senses information extracted from lexical knowledge base is combined with raw frequency counting and tf.idf metrics. A knowledge based approach which utilizes contextual senses of the concepts, a compound term heuristic, and the naive Bayes classifier are developed for finding taxonomic relations. To identify non-taxonomic relations among the concepts, a statistical method based on log-likelihood ratios is developed. In addition, a supervised approach for learning semantic constraints for labeling the relations between concepts appearing in prepositional phrases is developed.

The next chapter reviews the literature of methods that have been proposed for extraction of each of the components of the ontology.

Chapter 2

Literature Review

As mentioned in the previous chapter, with the advent of semantic web, many ontology engineering projects are incepted. But most of the projects are still in at their infancy. Some of those are Text-to-Onto [Maedche and Volz, 2001] and Hasti [Shamsfad and Barforoush, 2003]. Text-to-Onto is a part of the KAON(KArlsruhe ONtology) Tools for ontology management. It supports semi-automatic creation of ontologies using text mining algorithms. Currently, the tool includes concept extraction and concept association extraction algorithms. Text-to-Onto is embedded in in the ontology editing tool OntoEdit [Sure et al., 2002]. OntoEdit allows to browse and edit the existing ontological concepts. Text-to-Onto extracts the conceptual structures using the term frequencies from text. Concept hierarchy is extracted using the hierarchical clustering algorithms and non-taxonomic relations are extracted using association rule mining algorithm. Text-to-Onto tool requires user's verification at each stage of the ontology extraction process. For example, concepts extracted using frequency counting need to be verified before finding the relations between the concepts. Also, it requires manual labeling of internal nodes of the hierarchical clusters to find the taxonomic relations.

Similar to Text-to-Onto, Hasti is another tool developed extracting ontologies. Hasti is developed for processing Persian texts. Hasti operates in both cooperative and unsupervised modes. In cooperative mode, the user decides the selection or rejection at each stage of the process. For example, the user has to select the concepts from the candidate ones. In the unsupervised mode the system automatically selects each of the components of the ontology. In an overview, Hasti, initially, accepts a few top-level concepts, taxonomic and non-taxonomic relations as kernel elements, and extends initial seeds by adding more concepts. These kernel elements are linguistically motivated concepts like object, action, property, and etc. In Hasti, to extract the candidate concepts, a set of rules are defined to identify the structural sentences. A set of sentences matching one of the rules are considered as candidates. From the candidate sentences, candidate concepts are extracted by identifying nouns using predefined structures. To find the taxonomic and non-taxonomic relations, both hierarchical and non-hierarchical clustering algorithms are used. In addition to clustering algorithms, Hasti uses predefined semantic templates to extract the knowledge from

the candidate sentences. Hasti is an ongoing project and also does not report any new methods on identification of relations.

Along the lines of Text-to-Onto and Hasti, several other organizations have started various projects for ontology extraction such as ASIUM [Faure and Nedellec, 1998] and FFCA [Quan et al., 2004]. ASIUM learns semantic relations by clustering the nouns based on their occurrence with the verbs. In ASIUM each of the clusters of nouns is presented to the user for labeling. FFCA incorporates fuzzy logic into formal concept analysis for learning ontologies. In FFCA, concepts are extracted based on fuzzy membership value associated with each context. Conceptual relations between the concepts are obtained using fuzzy conceptual clustering algorithms.

Even though there is a lack of much work on extraction of full scale ontologies, considerable research has been done in the extraction of individual components of the ontologies. The individual components of the ontologies are useful in building either terminological knowledge bases or dynamic thesaurus. Most of these works are closely related for ontology extraction. The following sections describe the existing approaches in the acquisition of each of the components of the ontology.

2.1 Concept Extraction

Even though there exists no formal definition for concept, it is commonly agreed that any domain specific term which describes a part of the domain is called a concept. Keywords occurred in the domain texts are often considered as the domain specific terms. The keywords of a domain are often used in various text processing applications such as indexing, text summarization, and automatic abstract generation etc. Because of the various applications of the keywords, keyword extraction has been investigated extensively. With the perception of the above definition for the concept, most of the keywords of the domain are considered as valid concepts of the domain. Though there exists an extensive literature on concept extraction, most of the techniques are rely on large training data [Turney, 2000], domain dependent [Paice and Jones, 1993], or semi-automatic [Jacquemin, 1996]. Supervised learning methods for extraction of concepts rely on large amounts of training data to identify the patterns to extract new terms. Domain dependent techniques assume domain specific patterns to be defined beforehand. Predefined patterns are applied to texts to extract terms. In semi-automatic approaches, a set of reference terms for bootstrapping need to be provided, and extracted concepts are validated by the domain experts. It is desirable to develop domain independent techniques for conceptual term extraction because we generally can not the existence of domain specific patterns beforehand.

In this section, we discuss three different existing techniques which are closely related to our proposed approach. These techniques are based on the contextual patterns, syntactic patterns, and co-occurrence heuristics.

In [Paice and Jones, 1993], Paice and Jones presented a technique for extraction of conceptual terms using contextual patterns. In this technique, the authors define, manually, a collection of patterns using various stylistic sentence structures, and the conceptual roles are associated with them for each domain. Terms extracted from

sentences satisfying the patterns are considered as the concepts of the domain. For example, for a given context pattern

PEST is a ? pest of SPECIES

the example sentence which satisfy the pattern is

`A.lolli is a common pest of ryegrass`.

From the preceding sentence, `A.lolli` which matches the concept *PEST*, and `ryegrass` which matches the concept *SPECIES* are extracted as valid terms.

The advantage of this techniques is that it doesn't require expensive natural language processing techniques like parts of speech tagging or parsing. But the presented technique finds the instances of the concepts rather than concepts themselves. In the above sentence, `A. lolli` and `ryegrass` are instances of the concepts *PEST* and *SPECIES* respectively. Since the context patterns used in the above approach are domain specific, one needs to define such patterns for each new domain. Hence it lacks the portability to new domains. Also terms which does not appear in any of the pre-specified patterns will not be retrieved. Though it is not verified explicitly, the presented technique might have a low recall.

Another technique explored for extraction of domain specific terms is presented by Jacquemin [Jacquemin, 1996]. Jacquemin's technique relies on a set of reference terms of the domain for bootstrapping. The occurrence of reference terms in the text are tagged using FASTR(FAst Syntactic Term Recognizer) * partial parser. Using the part of speech(PoS) tags of reference terms appearing in text, syntactic patterns are developed automatically. Additional patterns are derived using variations of PoS tags of reference terms. A set of domain specific terms are extracted using the new patterns. New terms, which may not exist in domain text, are generated by applying coordination, insertion, and permutation strategies.

The advantage of this approach is that it allows to extract domain relevant terms which are not present in the domain text. But the validity of the new terms needs to be verified manually. The main constraint of this technique is that a set of reference terms need to be provided as input for each new domain to start with. Also, the extracted terms need to be verified manually.

Gelfand et al.'s technique [Gelfand et al., 1998] for term extraction utilizes lexical knowledge base. In this technique, initially, a set of words are collected as base list. For each word in the base list, the system extracts hypernyms and hyponyms for each of the senses from WordNet†. New words extracted from the WordNet are added to the base list. Hyeprnyms and hyponyms for each word are extracted for a predefined threshold of height and depth in WordNet. A directed graph is drawn in which each node represents a word and each edge represents hypernym/hyponym relation between the words. From the graph, nodes with smaller degree of incidence are removed from the graph indicating that the corresponding terms are not domain

*FASTR is developed by Jacquemin.

†Detailed discussion on WordNet is presented in the Appendix B.

specific. Words representing the remaining terms are considered as the concepts of the domain.

The major drawback of Gelfand et al.'s method is that it assumes all of the words collected from domain text are defined in the WordNet. From our observation, it is clear that large number of words, especially compound terms, are not defined in the WordNet. Further more, a given word's hypernym(s) or hyponym(s) varies depending on context of the word. Since no explicit sense disambiguation technique is applied to identify the senses for words in the base list, it is possible that irrelevant words of the domain get added to the base list. With reference to the experiments in [Vossen, 2001], using WordNet hierarchy without considering WordNet senses does not improve the performance of the conceptualization system.

Along with the above methods, there exists other techniques for keywords extraction based on term frequency, tf.idf metrics [Salton and Buckley, 1988] [Tomokiyo and Hurst, 2003], naive Bayes classifier [Frank et al., 1999], and genetic algorithms [Turney, 2000]. The main idea behind frequency based techniques is that terms which occur most frequently are considered as the relevant terms of the domain. Whereas tf.idf measures relevancy a term to a given document in a collection of documents. In this dissertation, we developed a new technique for extracting conceptual terms from domain texts. The proposed technique takes the advantage of raw frequency counting and tf.idf metrics for extracting an initial cut of relevant terms. The initial cut of terms obtained are then refined using lexical knowledge base. Additional relevant terms are added using highly relevant terms from the refined terms. Details of the proposed technique for concept extraction are presented in chapter 3.

2.2 Taxonomy Extraction

For automatic construction of ontologies, we need techniques for automatic extraction of both hierarchical and non-hierarchical relations. Existing methods on relation extraction are presented in two separate sections as hierarchical relations extraction and non-hierarchical relations extraction.

In the literature, hierarchical relations among the concepts are also called taxonomic relations or simply taxonomy. Existing techniques for finding taxonomic relations can be classified as pattern based, clustering based approaches, and combination of both. In pattern based approaches, the user defines a set of predefined lexico-syntactic patterns. Domain text is verified against the patterns to obtain the instances of taxonomic relations. In clustering based approaches, hierarchical clustering algorithms are used for finding the taxonomic relations between the concepts. And heuristics are used for labeling the internal nodes in the clusters. In the combined approaches, internal nodes are labeled using the instances extracted using lexico-syntactic patterns.

One of the early works for finding taxonomic relations based on lexico-syntactic patterns is presented by Hearst [Hearst, 1992]. Hearst's patterns and their corresponding hyponym relations are shown in Table 2.1.

Table 2.1: Hearst's Patterns for Taxonomy

S.No	Syntactic Pattern	Hyponym Relation ($\forall NP_i\ 1 \leq i \leq n$)
1	NP_0 such as $\{NP_1, NP_2, ...$,(and | or)$\}$ NP_n	hyponym(NP_i, NP_0)
2	such NP_0 as $\{NP_i,^*$ (or | and) $\}$ NP_n	hyponym(NP_i, NP_0)
3	NP_1 $\{, NP_i\}^*$ $\{,\}$ (or | and) other $NP_n + 1$	hyponym(NP_i, NP_{n+1})
4	NP_0 $\{,\}$ (including | especially) $\{NP_i, \}^*$ {or | and } NP_n	hyponym(NP_i, NP_0)

Hearst's procedure to identify the hyponym relations is as follows. Extract the sentences which satisfy any of the patterns listed in Table 2.1. For each sentence, identify the noun phrases which satisfy corresponding NP in the pattern. Label the relation among the noun phrases using the corresponding hyponym relation of the pattern. For example, the sentence,

`The bow lute, such as the Bambara ndang, is plucked and has an individual curved neck for each string.`,

satisfies the pattern 1 in Table 2.1. Here, NP_0 corresponds to `bow lute` and NP_n corresponds to `Bambara ndang`. Hence, the homonym relation extracted is

hyponym("Bambara ndang", "bow lute").

It is quite intuitive that the authors mention such sentences as illustrations of the unknown terms meaning identification. Further more, Hearst presented a simple heuristic to extract the instances of additional relations as follows. Select a set of pairs of terms which satisfy the target semantic relation for bootstrapping. Extract the sentences which consist of pairs of terms. From the extracted sentences, identify the commonalities and hypothesize the common structures that yield patterns of target relation. Even though the above heuristic seems to work, Hearst mentioned that they didn't get much success in extracting the meronym(i.e. part-whole) relations.

Taxonomic knowledge acquisition technique presented in [Iwanska et al., 1999] is similar to Hearst's approach. [Iwanska et al., 1999] presents mainly two lexico-syntactic patterns for taxonomy extraction. One of the patterns is same as pattern 2 in Table 2.1. Another pattern consists of if a pair of terms connected by the verb *like*. But, according to [Iwanska et al., 1999], large number of pairs extracted with the *like* pattern are spurious. Simple heuristic rules are proposed to reduce the spurious relations and to identify the concept boundaries. Detailed experimentation of the patterns is performed on the *Time Magazine* corpus.

A framework for acquisition of hypernym links among multi-word terms using single-word candidates is presented in [Morin and Jacquemin, 2003]. This system is

built on the previous work described in [Hearst, 1992]. It provides a classifier for the purpose of discovering new lexico-syntactic patterns through corpus exploration for the given semantic relation. As a whole, the system is a combination of *Promothee*, a tool for structuring the relationships among the single-word terms, *ACABIT* [Daille, 2003], a tool for acquisition of multi-word terms, and *FASTR* [Jacquemin, 1996], a tool for term variant recognition of the candidate terms. Finally, the system inherits the relations between the single word terms to the corresponding multi-word variants. The *Promothee* system extracts lexico-syntactic patterns for the given semantic relation using a set of terms which satisfy the relation, In summary, *Promothee* collects sentences from the corpus and determine the patterns of the sentences in which the above seed terms are present. Additional sentences satisfying the patterns are extracted to find more instances of the semantic relation. The *Promothee* system is experimented with three (hypernym, merge, produce) relations. Similar to techniques, the *Promothee* also extracts relations between the terms which occur in the same sentence only. To find the relations between the terms across different sentences, the system tries to identify the variations of the terms for which the relations are already determined, then the same relation assigns to the variants. For example, if the relation between *fruit* and *apple* is known then the relation between multi-word variants *fruit juice* and *apple juice* is also labeled as same. The FASTR extracts multi-word terms using syntactic, morpho-syntactic, ans semantic categories of variations. For each of the categories, various rules are defined to identify the multi-word terms.

Semantic relations among the multi-word terms, with reference to semantic relations among their constituent words, are labeled if the following three constraints are satisfied.

Semantic Constraint Two multi-word terms w_1w_2 and $w_1^{'}w_2^{'}$ are semantic variants of each other if the following three constraints are satisfied.

1. Some type of semantic relation **S** holds between w_1 and $w_1^{'}$ and/or between w_2 and $w_2^{'}$.

2. w_1 and $w_1^{'}$ are head words and w_2 and $w_2^{'}$ are arguments with similar thematic roles.

3. w_1w_2 and $w_1^{'}w_2^{'}$ share the same type **S** of semantic relation.

The above technique for finding semantic relations among multi-word terms also provides the opportunity to cluster the semantically related words. This technique provides the opportunity to increase the recall in terms of the number of relations extracted and also the coverage of the terms. It is able to find the relations among multi-word terms which occur in two different sentences. But to label such relations, the relation between their constituents should be known by some other means. The expert's intervention is required to validate the patterns identified by the *Promothee* using the seeds. Extraction of the seeds from the knowledge base for the given semantic relation also requires human involvement. Taxonomic relations among the

terms which does not follow the preselected patterns are not retrieved using the above mentioned system.

Even though the above patterns retrieve valid taxonomic relations, these patterns extract hyponym relations between the concepts which occur only in the predefined patterns. According to the results presented in the corresponding works, the number of hyponym relations extracted comparing the size of the corpus is very small. These approaches may have high precision because most of the extracted relations are valid but produce a low recall because of the occurrence of few such patterns in domain text. Further more, corpus used in these experiments does not belong to a fixed domain. The disadvantage of the pattern based approaches is that these approaches find pairs of nouns which hold taxonomic relations rather find the relation between the given concepts. Hence pattern based approaches may be suitable for extending thesaurus but not for ontology acquisition.

To build a hypernym-labeled tree from text, Caraballo [Caraballo, 1999] presented a technique based on cosine similarities using bottom-up clustering. The input to the technique is a set of nouns which are separated by conjunctions or appeared as appositives. Using the frequency of occurrence of each word along with the other words as the criteria, similarity of the words is determined by the cosine metric. Two nouns which are highly similar are grouped by giving them the common parent. The process is repeated until a single parent is found for all the nouns. Similarities among the internal nodes are determined using the weighted measure of the similarities of their leaves. Labels for the internal nodes are determined using their leaves and the Hearst's patterns. Each leaf maintains a vector of hypernyms extracted using the patterns. For each internal node of the tree, we construct a vector of hypernyms using the hypernyms of the children. Internal nodes are labeled with the hypernym which has maximum count. For each internal node, the author suggested assigning the best, second-best, and third-best hypernyms based on their occurrence count. Also, Caraballo suggested a simple heuristic to reduce the size of the tree by eliminating the unlabeled internal nodes.

Though this technique is quite straightforward and simple, it also depends on the Hearst's patterns for labeling the hypernyms. Due to this, as the author mentioned, large number of nodes are unlabeled. Another constraint is it considers only terms with single word which occurs in the specific contexts. These words may describe only a subset of the domain. This method is also experimented on domain independent corpus.

Snow et al [Snow et al., 2004] proposed a supervised learning technique using dependency paths as features to find the syntactic patterns for hypernym relation extraction from text. The dependency paths are generated using parse trees. Training set for this approach is pairs of terms (W_i, W_j) which occur in a sentence. The pair of terms are classified as valid hyponym/hypernyms if both of them are in hypernym relation according to the WordNet with the most frequent sense. The patterns for hypernym relation are discovered from the dependency paths in parse trees which occur in at least five unique hypernym/hyponym pairs in the corpus. This technique also restricts the hypernym relations between the terms in the same sentence only.

Similar to the above techniques for taxonomic relation extraction, in [Cederberg and Widdows, 2003], the authors used the Latent Semantic Analysis(LSA) [Yates and Neto, 1999] to eliminate the invalid hyponym/hypernym pairs, and used the coordination information to improve the recall. Other related works for the taxonomy extraction are [Kashyap et al., 2004], [Ryu and Choi, 2004], and [Fotzo and Gallinari, 2004]. But the techniques presented in [Kashyap et al., 2004] and [Ryu and Choi, 2004] are specific to the medical domain text, and thus domain specific compositional terms can be exploited. In [Fotzo and Gallinari, 2004], hierarchies are found using document collection subsumption rule i.e. Hyponym relation between W_1 and W_2 is based on the relative frequencies that the number of documents contains both W_1 and W_2 versus number of documents in which W_2 alone is present.

Even though there exists an extensive collection of literature on taxonomy extraction, none of the presented techniques assume that all documents belongs to a single domain. As mentioned before, existing methods extract the taxonomic relations from text by identifying instance of the patterns or nouns occurring in pre-specified positions. To the best of our knowledge, none of the methods find the taxonomic relations between the given set of concepts using the text in which they occurred. In our research, we developed a technique for retrieving the taxonomic relations from WordNet by identifying the sense in which the terms are occurred. We also proposed a simple heuristic to find the taxonomic relations of the compound terms. In addition, we developed a supervised learning technique for finding semantic classes of the concepts. Detailed discussion of our proposed methods is presented in chapter 4.

2.3 Non-Taxonomic Relations Extraction

Another major component of ontologies is non-hierarchical relations between the concepts. Extraction of non-hierarchical relations is the least tackled problem in ontology learning tasks. In general, identification of non-hierarchical relations involves finding the candidate pairs of concepts such that the their constituents are semantically related and identification of the label for the semantic relationship. For example, for the concept pair (*company*, *product*), the relationship label can be *sell*, *manufacture*, or *consume*.

With above notion of non-hierarchical relations, existing works on extracting relations from texts can be classified into three categories listed as follows.

1. Approaches for finding relations between named entities.

2. Approaches for extraction of concept pairs which hold the given relationship label.

3. Approaches for identification of relationships between the concepts in a given set of concepts.

One of the important problems in information extraction research is finding the relations between the named entities. Here relations or relationship labels are identified among a fixed set of entities such as *person*, *organization*, *location*, and etc. Some

of the works for finding relations between such entities are presented in [Hasegawa et al., 2004], [Stevenson, 2004], [Yangarber et al., 2000], [Riloff, 1996], [Zelenko et al., 2000], and [Agichtein and Gravano, 2000]. All the above listed works use supervised or unsupervised learning, or contextual patterns based approaches for finding the relations between the entities. Initially, these works use named entity tagging for identification of instances of the entities in plain texts and try to find the contextual patterns to label the relationships between the entities. In these approaches entities are fixed irrespective of domain text considered. Since concepts of the ontology varies with respect to the domain text considered, techniques for finding the relationships between named entities may not be suitable for finding the ontological relations.

The second category of the approaches for relationship extraction task are techniques for finding the concept pairs such that their constituents holds the pre-specified semantic relationship. Some of the existing works which follow the above mentioned approach are [Berland and Charniak, 1999], [Girju et al., 2003], [Girju and Moldovan, 2002], and [Turney, 2006]. Among the existing works listed above, [Berland and Charniak, 1999] and [Girju et al., 2003] finds the noun pairs which hold *part-whole* semantic relation. [Girju and Moldovan, 2002] presents the patterns for identification of concept pairs which hold *cause-effect* relationship. [Girju and Moldovan, 2002]'s technique learns the semantic patterns for a given semantic relation. Learned patterns are used to find concept pairs which hold the same relationship.

Charniak et al presented a pattern based technique to extract the parts of the components from large corpora. To extract the patterns for *part-whole* relation, the authors used the pair("basement", "building") which hold the specified semantic relation and extracted all the sentences which consists of the pair. From these sentences, a set of patterns are extracted. After the manual evaluation, the number of patterns extracted are reduced to two. To extract additional pairs, for a given word, all the sentences which satisfy any of the two selected patterns are extracted. From each sentence, the noun phrase which is in the part position is extracted. All the extracted parts are ordered by the likelihood that they are true parts according to the sigdiff metric. The metric is based on the idea that for a given whole W and part P, how far apart can we be sure the distributions $P(W|P)$ and $P(W)$ at the given significance level, say .05 or .01. The authors tested the above technique for six different part words for each of the whole words. After the human evaluation by six different subjects, the authors claim that the presented technique results in 55% accuracy for the top 50 words as ranked by the system. As the author mentioned, this technique relies on very large corpus (100,000,000 words). This technique requires to provide the terms which satisfy the given relation to identify the patterns. To reduce the extraction of invalid parts, for example, "driveability" is strongly correlated with car, the author tried weed out the most of the qualities by removing the words with suffixes "ness", "ing", and "ity".

Similar to the above work, [Girju et al., 2003] described a technique to learn semantic constraints for finding the *part-whole* relations. Though the authors didn't mention explicitly, this technique is an extension of the work in [Berland and Charniak, 1999]. Here the authors able to extract three patterns shown in Table 2.2 by

Table 2.2: Lexical Patterns for Part-Whole Relations

S.No	Lexical Pattern
1.	NP_1 of NP_2
2.	NP_1's NP_2
3.	NP_1 Verb NP_2

analyzing the TREC-9 corpus. Among the extracted patterns, patterns 2 and 3 may also indicate *Possession* relation. For example, it Kate has a cat.

To identify valid *part-whole* pairs from sentences which satisfy one of the patterns in Table 2.2, the authors proposed a supervised learning technique using C4.5 decision tree algorithm [Quinlan, 1993] for learning semantic constraints. The attributes representing each candidate pair are WordNet class and sense number of the part and whole terms. Each noun pair is classified as whether its constituents hold a valid *part-whole* relation or not. The authors extracted 34,609 sentences as positive examples and 46,971 as negative examples. For both *Part NP* and *whole NP* in each example, the authors assigned semantic (WordNet class) class and sense number manually. From these examples, the authors filtered out ambiguous instances by assigning more specific WordNet classes. Specialization process is repeated until the ambiguity in the input data is resolved. The C4.5 algorithm is applied to unambiguous examples to learn the semantic class pairs which indicate valid *part-whole* relation. The rules learned using C4.5 algorithm are considered as the constraints to be satisfied for any two NPs to satisfy the *part-whole* relation. Here, the authors put enormous effort in assigning the class and sense number for each pair of the NPs in the sentences manually. The learned semantic constraints can be used to filter some of the irrelevant noun pairs. To apply the rules learned for a noun pair, it is required to find, for each noun, the taxonomy path from noun to a top class in the WordNet. These rules are not useful for noun pairs whose constituents are not listed in the WordNet. Even for a noun pair whose both of the nouns are present in the WordNet it is required to identify their sense correctly to able use the learned rules. A supervised learning technique for finding the semantic classes is proposed in this dissertation. Detailed discussion on the proposed approach for semantic class labeling is discussed in chapter 4.

In [Girju and Moldovan, 2002], a semi-automatic technique for extraction of lexico-syntactic patterns for *cause-effect* relation is presented. This technique also relies on large corpus and the WordNet. The algorithm primarily consists of two steps. In the first step, the algorithm selects a set of pairs of noun phrases which hold the *cause-effect* relation from the WordNet. Extract the sentences which consists of the selected noun phrases and are of the of the form $< NP1$ *verb*$|$*verb-expression* $NP2 >$ from corpus. Filter the nouns such that each of the nouns corresponds to $NP2$ has to be one of the *human action*, *phenomenon*, *state*, *psychological feature*, and *event* WordNet classes. The nouns corresponds to $NP1$ must be subclasses of *causal agent*. The *Verb/Verb-expression* must have few number of senses and highly frequent. The causal relationship extracted using the above patterns is validated and

assigned a rank (between 1 and 4) to indicate its strength. A simple algorithm based on the WordNet classes, frequency, and ambiguity of the verbs is proposed to rank the each relationship. Using the causation verbs extracted from the above approach, 50 sentences for each verb thus around 3000 sentences extracted and the 1321 sentences of them are in the $< NP1$ *verb* $NP2 >$ pattern. From these sentences, the system extracted 230 relations as valid with one of the four ranks. This approach is quite general, domain independent, and is not dependent on the hand coded patterns. But this technique requires valid instances of the causal relationship and also makes use of the external lexical knowledge base(i.e.WordNet). The number of relations extracted from the large corpus (3GB of news articles) is very few (230 relations).

We believe the presented techniques might be useful in extending the thesaurus or lexical knowledge bases. But these techniques might not be suitable for learning ontology relations between the concepts because of the following reasons. One is extracted ontology concepts may not present in the patterns identified. The other is nouns present in sentences which satisfy the patterns might not have been considered as valid concepts of the domain. Further more, using these techniques, considering the amount of input text processed, only a very few pairs are identified.

The last category of approaches for identification of semantic relations is finding the candidate concept pairs and labeling the relationship between their constituents. Techniques in this category need to identify the existence of a relationship between concepts in a concept pair and also to label the relationship appropriately. In [Kavalec et al., 2004], the authors presented a simple heuristic based on the conditional probability to label the relations between the concepts using verbs. The relation labeling technique is based on the hypothesis that predicate of a semantic relation can be characterized by the *verbs* frequently occurring in the neighborhood of pairs of concepts associated with it. Each triple, pair of concepts and the verb nearby (C_1, C_2, V), is treated as a transaction. For a given transaction, if its frequency of occurrence is greater than the expected frequency then the verb(V) is considered as a candidate to label the relation between the concepts. All the verbs which occur above the expected frequency along with the concepts are considered as candidates for relations among the concepts. The triple (C_1, C_2, V) is valid, if and only if both the concepts C_1, C_2 occur within n (experimentally $n = 8$) words from V. In the experiments, TAP knowledge base[‡] is used to identify the classes of the named entities. TAP consists of a large repository of lexical entries such as proper names of places, companies, people and the like. The portability of TAP knowledge base to other domains is a question. The technique for automatic identification of the concept for a given instance is a research question. Another major drawback of this approach is the it is not able to identify the direction of the relationship($C_1 \rightarrow C_2$ or $C_1 \leftarrow C_2$). Also, it is not able to label the relations between the concepts whose lexical entries are connected by prepositions or conjunctions. Further more, this technique does not address the issue of finding the relations among the concepts which does not occur in the same sentence. As the author mentioned the results are not impressive due to the following

[‡]www.tap.stanford.edu

reasons: richness and relevance of the concept taxonomy, richness and relevance of the lexicon, style of the underlying text, performances of the PoS tagger.

Another related work comes under third category for semantic relation extraction is presented in [Ciaramita et al., 2005]. It is based on the χ^2 test. The technique works as follows. Each occurrence of instances of the concepts in the domain text are replaced with the corresponding concepts. From the modified text, select the sentences which satisfy the pre-specified patterns. The dependency patterns are extracted for each of the sentences. For a given pair of concepts and a dependency pattern, if the occurrence of concepts as fillers of the pattern is greater than the expected frequency then the relation between the concepts is labeled is name of the dependency pattern. The χ^2 test at 95% confidence interval is used to test the hypothesis. As mentioned before, the key issue is identification of the concept for a given instance. The authors didn't explain about the technique used to replace the instances with their corresponding concepts. This technique also finds the relations between concepts in the same sentence only. This work is specific to the molecular biology domain. Portability to other domains is a question because structural patterns of terms varies with the domain.

Along with the above techniques, other techniques in the literature for finding the relations between the concepts are [Schutz and Buitelaar, 2005] and [Faure and Nedellec, 1998]. Similar to [Ciaramita et al., 2005], [Schutz and Buitelaar, 2005]'s work also extracts concept pairs present in dependency relations and use the χ^2 test to verify the statistical significance on the togetherness of the concepts. [Faure and Nedellec, 1998]'s work learns semantic relations from sentences occurring in pre-specified patterns. Nouns occurring in the pre-specified positions(subject or object) for a given verb are clustered. Each of the clusters are manually labeled with the representative concept. The verb with which a cluster is formed is considered as the label for relationship between the concepts. The main constraint of this method is it requires manual labeling of clusters with concept names in finding the relationships.

Among the three different categories of methods presented, the third category of methods are more essential for finding the non-hierarchical relations between concepts. Considering the various constraints mentioned above with the existing approaches, we have developed two different methods for finding non-taxonomic relations. One is by identification of the verbs associated with the existing concepts and using log-likelihood ratios. The other approach is a supervised learning algorithm. The supervised learning algorithm learns semantic constraints for labeling relations between the concepts using prepositional phrases occurring in texts. Detailed discussion on these approaches is presented in chapter 5.

2.4 Summary

Because of the various constraints in natural language processing and knowledge acquisition, extraction of domain-specific ontologies is a difficult task. Even though several projects are started for automatic construction of ontologies, most of them are still in at their infancy. Some of those are Text-to-Onto and Hasti. At this

point, both Text-to-Onto and Hasti find conceptual terms using term frequencies and concepts association using clustering algorithms. Even though there exists a lack of research on the automatic extraction of full-scale ontologies, considerable attention has been focused on the extraction of its individual components. The components of the ontologies are useful for either constructing or extending terminological knowledge bases and dynamic thesaurus.

Existing techniques for extraction of concepts are based on contextual patterns, semi-automatic, or supervised learning approaches. Contextual pattern based approaches require patterns to be defined manually. Domain text is verified against the patterns to find the occurrence of domain-specific terms. The main constraint of this approach is that it requires manual effort to define the patterns for each of the domains. Because of this difficulty , these techniques many not be portable to new domains. Semi-automatic approaches [Jacquemin, 1996] require either seeds to be provided for bootstrapping or extracted terms need to be verified manually. These approaches can extract terms which are not exist in the text by permutation of the words in the already extracted terms. But The obtained terms need to be verified for their correctness. In supervised learning approaches, a large number of conceptual terms with their features need to be provided for learning constraints to extract the relevant terms. Because of these constraints in the existing approaches, this dissertation presents a technique for automatic extraction of the conceptual terms using raw frequency counting, tf.idf metric, and senses information extracted from the WordNet.

In applied natural language processing, finding taxonomic relations from text is widely investigated. Existing techniques for taxonomy extraction can be classified into pattern based, clustering based, and combination of both. Most of the pattern based approaches rely on the Hearst's patterns listed in Table 2.1. In pattern based approaches, domain text is verified against the pre-specified patterns to find the instances of taxonomic relations. Since only a few sentences satisfy the pre-specified patterns, recall of these methods will be poor. In addition, pattern based approaches find taxonomic relations between nouns occurred in pre-specified patterns rather than between the concepts already identified. In clustering based approaches, hierarchical clustering algorithms are used for finding taxonomic relations between the concepts. The main difficulty with the clustering based approaches is finding labels for the internal nodes. Some of the techniques are developed for labeling internal nodes using results of the pattern based approaches. Even these approaches able identify labels for only a few internal nodes. Considering difficulties in the existing approaches, WordNet based approach using sense disambiguation, compound term heuristic, and naive Bayes approach for semantic classification of concepts are developed for finding taxonomic relations.

One of the least tackled tasks in automatic ontology learning is identification of non-taxonomic relations. Existing methods for finding non-taxonomic relations can be classified into three categories as finding relations between the named entities, finding concept pairs which appear in the pre-specified relation, and the other is finding relations with labels between the concepts in a given set of concepts. The first category techniques find instances of the relations between the `person`, `organization`, and `location` entities. Second category techniques find concept pairs which are in

pre-specified relation namely *part-whole*, *cause-effect*, etc. The last category of approaches find relations between the concepts based on the dependency patterns and statistical techniques. Main constraint of these approaches is that semantic relations between the concepts occurred in prepositional phrases and appositives are not considered. Because of the constraints with the existing approaches, in this dissertation, SVO Triples method is developed to find the semantic relations between concepts occurred as subject and object(s). And a supervised approach for learning semantic constraints to find the relations between the concepts occurred in prepositional phrases is investigated.

In summary, our research develops a set of methods for automatic extraction of the concepts, taxonomic relations, and non-taxonomic relations. These methods can be combined to form an integrated system for automatic acquisition of ontologies.

Chapter 3

Concept Extraction

As mentioned in chapter 2, even though there exist various methods for extraction of concepts, most of the techniques rely on large training data, domain dependent, or semi-automatic. Domain dependent techniques rely on domain specific patterns to be defined manually before applying them to texts. In semi-automatic approaches, a set of reference terms for bootstrapping are needed or extracted concepts have to be validated by domain experts. In this chapter, we present several domain independent techniques to extract the concepts in fully automatic way. The proposed techniques use word sense information to improve precision and use multi-word term compositions to improve recall. These techniques are applied on top of the frequency based methods such as raw frequency counting and tf.idf metrics.

In automatic extraction of concepts, the validity and relevancy of the terms must be addressed. Domain texts contain multi-word terms in addition to the single word terms. Though most of the single word terms can be validated by verifying in the dictionaries, multi-word terms are most likely absent in dictionaries. For example, though "electronic voting machine" is a valid term, it is not listed in dictionaries. All valid terms extracted from texts are not relevant to the domain. For example, even though "electronic voting machine", and "independent identity" are valid terms, only "electronic voting machine" is relevant to the electronic voting domain. Existing approaches have difficulty in handling multi-word terms and in achieving good precision and recall. Our method starts using frequency based techniques such as raw frequency counting or tf.idf weighting methods. To improve the recall and precision, several heuristics based on the senses and compound structure of the terms are developed. Senses of the terms are verified to filter out the irrelevant terms. Individual words in the multi-word terms are used to find out additional relevant terms. Combining all these heuristics, we have developed two different methods, namely Word Count Approach and WordNet sense count approach(WNSCA). Word Count Approach considers only the composition of terms whereas WNSCA uses senses of the terms along with the composition of terms.

3.1 Text Preprocessing

To identify the relevant terms from text, as part of the preprocessing step, the whole text is part of speech tagged. Noun phrase boundaries are determined using noun phrase chunker. Each of the identified noun phrases is refined by eliminating stop words and performing morphological analysis. Preprocessing steps are briefly described as follows and illustrated in Table 3.1.

Part of Speech Tagging and Noun Phrase Identification From the exploration of various part of speech(PoS) tagging tools, we considered using the Brill's Rule-based part of speech tagger [Brill, 1992]. Brill's tagger is a simple rule based general purpose tagger. It learns the rules from the tagged Wall Street Journal Corpus and Brown Corpus. The tagger has several advantages: its performance is comparable to stochastic taggers, it is portable to a different tag-set, and it stores less information. The Brill's Tagger tags the text using university of Pennsylvania(UPenn) tree bank tag-set. The tagger assigns a unique part of speech tag for each word.

Once the PoS tagging of text is completed, it is necessary to determine the phrase boundaries to identify the valid noun phrases in the text. BaseNP Chunker [Ramshaw and Marcus, 1995] is used to mark all the noun phrases. BaseNP Chunker takes PoS tagged English text as input and inserts brackets marking the contained BaseNP structures. BaseNP Chunker uses a fixed set of 500 BaseNP rules trained on Wall Street Journal corpus. The sequence of words enclosed in the brackets is considered as a valid term. To enhance the count of occurrence of relevant terms, stop word elimination and morphological analysis are performed as part of the preprocessing of text.

Stop Word Elimination Words which occur too frequent and do not contain any domain relevant information are called stop words. For example, *the*, *is*, *do*, and *such* etc are stop words. Since the stop words dominate the content words and obstruct the concept extraction, stop word removal is necessary. A list of 319 words from Porter's stop word list* is used as a reference stop word list. Each word in the noun phrases is verified against the stop word list to remove the stop words from noun phrases.

Morphological Analysis The appearance of a phrase in the text varies based on the context in which it occurs. To count all variations of a phrase as the occurrence of same phrase, morphological analysis is applied to the noun phrases. For example, singular and plural forms of a noun phrase should be treated as the same phrase. English grammar rules defined in [Hodges et al., 1997] are used for converting plural form of a noun to its singular form. If a noun phrase is in plural form (identified using PoS), its root form is identified by considering the word endings. Suppose the plural form of a phrase ends with "ices" then remove the last three characters and add "x" at the end of the phrase. For example, append"ices" → append"ix".
If a word has a possessive ending ('s, or '), the word is truncated by removing the

*www.dcs.gla.ac.uk/idom/ir_resources/linguistic_utils

Table 3.1: Illustration of Text Preprocessing

Step Name	Output
Raw Text	Many local election officials and voting machine companies are fighting paper trails, in part because they will create more work and will raise difficult questions if the paper and electronic tallies do not match.
⇓	
POS Tagging	Many/JJ local/JJ election/NN officials/NNS and/CC voting/NN machine/NN companies/NNS are/VBP fighting/VBG paper/NN trails,/NN in/IN part/NN because/IN they/PRP will/MD create/VB more/JJR work/NN and/CC will/MD raise/VB difficult/JJ questions/NNS if/IN the/DT paper/NN and/CC electronic/JJ tallies/NNS do/VBP not/RB match./JJ
⇓	
NP Chunking	[Many/JJ local/JJ election/NN officials/NNS] and/CC [voting/NN machine/NN companies/NNS] are/VBP fighting/VBG [paper/NN trails,/NN] in/IN [part/NN] because/IN [they/PRP] will/MD create/VB [more/JJR work/NN] and/CC will/MD raise/VB [difficult/JJ questions/NNS] if/IN [the/DT paper/NN] and/CC [electronic/JJ tallies/NNS] do/VBP not/RB [match./JJ]
⇓	
Stopword Elimination	local/JJ election/NN officials/NNS, voting/NN machine/NN companies/NNS , paper/NN trails,/NN, part/NN, work/NN, difficult/JJ questions/NNS, paper/NN, electronic/JJ tallies/NNS, match./JJ
⇓	
Morphological Analysis	local election official, voting machine company, paper trail, part, work, difficult question, paper, electronic tally

possessive characters. Each of the preprocessing steps described are illustrated with an example sentence in Table 3.1.

Name Phrase Elimination Although the occurrence of proper names is typically not frequent, these names are not useful for conceptualization of the domain. So, it is useful to eliminate such name phrases. There exists a named entity recognition(NER)[†] research dedicated to the recognition of name phrases. For our purposes, the following simple heuristic is used. A phrase whose first letter is a capital letter, and is not present in the start of a sentence is considered as the name phrase. Though this heuristic looks simple, it is able to extract most of the named entities.

3.2 Raw Frequency Counting and tf.idf Metric

The preprocessing steps produce a set of noun phrases along with their frequencies. Theses noun phrases are considered as the valid terms of the domain. However, not all valid terms are relevant to the underlying domain. To obtain the initial cut for relevant terms, two different metrics are employed. One is using raw frequency counting and the other is using tf.idf metric.

Raw Frequency Counting Method Valid terms which possess high frequency of occurrence in the domain texts are considered as the relevant terms of the domain.

tf.idf Metric Method In information retrieval, tf.idf weighting scheme is often used to determine the relevancy of a term to a document in a given corpus. One way of representing the tf.idf scheme is shown in the equation 3.1. In equation 3.1, $T_{i,j}$ refers to term i in document j, D is a collection of documents, and D_i is the collection of documents in which term i occurred.

$$tf.idf(T_{i,j}) = f(T_{i,j}) * log\frac{|D|}{|D_i|} \tag{3.1}$$

For concept extraction, the input text considered consists of domain documents. Also, the preprocessing techniques, mentioned in section 3.1, are employed to eliminate the general terms. We believe the terms which occur in all the domain documents are also candidate concepts. To determine the candidacy of a term to be a concept, we have modified the tf.idf as shown in the equation 3.2. The modified tf.idf is developed with the notion that terms which occur more often in domain relevant documents and very rare in general texts are considered as the relevant terms. In equation 3.2, D and G are collections of domain documents and general texts respectively, D_i and G_i are documents in which term T_i occurs in the collections D and G respectively, and $f_D(T_i)$ is the frequency of term T_i in the domain collection D. For each noun phrase extracted from D, tf.idf values are computed using equation 3.2. Terms with high tf.idf values are selected to obtain the initial cut of the relevant terms.

[†]NER is the part of 1995 MUC-6 task

$$tf.idf(T_i) = f_D(T_i) * log\frac{|D| + |G|}{|D_i| + |G_i|} \tag{3.2}$$

Our proposed heuristics listed in the following sections use the initial cut of relevant terms to extract additional relevant terms.

3.3 Word Count Approach

The Word Count Approach considers a phrase as relevant if some or all of the words in the phrase occur in the initial cut of terms words list. It proceeds as follows. A base list is created, which consists of individual words appearing in the initial cut of terms(say top 10% of the terms). Apply one of the following two heuristics, using the base list, to extract the relevant terms.

1. **All Words Heuristic** For each of the extracted noun phrases, identify the individual words of the phrase. If all of its words are in the base list then the phrase is considered as a concept of the domain.

2. **Any Word Heuristic** For each of the extracted noun phrases, identify the individual words of the phrase. If at least one of its words is in the base list then the phrase is considered as a concept of the domain.

It is quite obvious that All Words Heuristic may have high precision and Any Word Heuristic may have high recall. More details on these results are presented in the evaluation section(§3.5§). These heuristics possess the advantage of being simple and easy to implement. They do not assume any prior knowledge about the domain text. At the same time, these heuristics do not use any external knowledge bases like WordNet or dictionaries. Since the terms considered appear at least once in the texts, the retrieved terms are indeed valid terms.

3.4 WNSCA+{PE, POP}

WordNet is a lexical knowledge base developed at Princeton university. In WordNet, nouns, adjectives, adverbs, and verbs are organized with semantic information for each of the senses. Detailed discussion on WordNet is presented in the Appendix B. WordNet sense count approach(WNSCA) makes use of the WordNet sense information to obtain the initial cut of relevant terms.

WNSCA WNSCA is based on the intuitive idea that terms with fewer senses are more specific to the domain than the high sense count terms. Therefore one could use the number of senses(sense count) of a term as a filter to eliminate irrelevant terms of the domain. This will help to improve the precision. Our observation of the sense counts for various terms indicates that on the average each word has 3 to 4 senses.

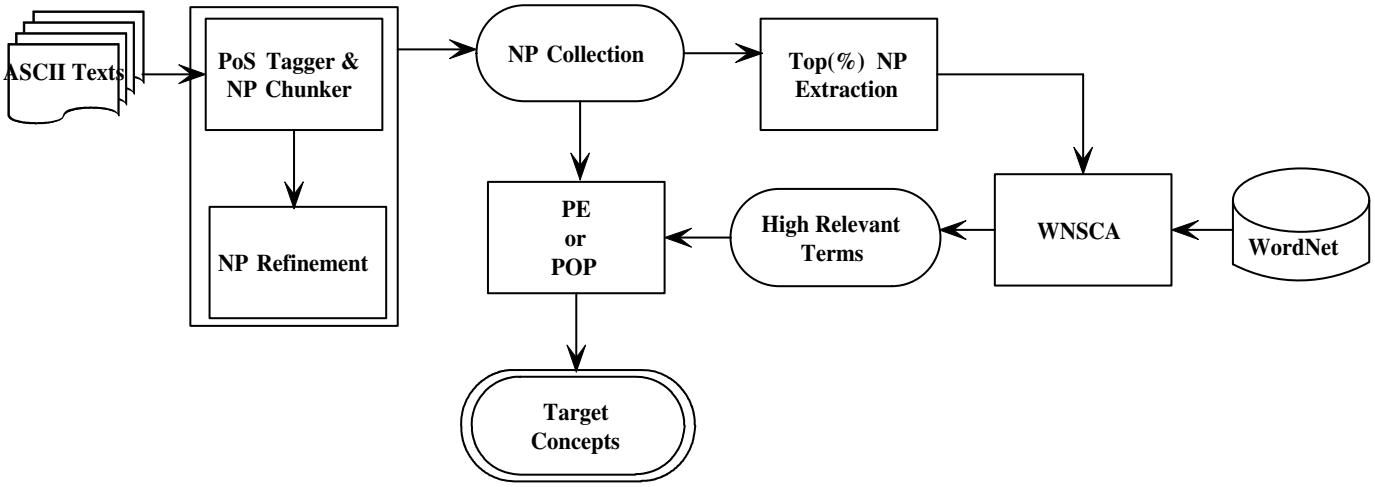

Figure 3.1: General Framework of WNSCA method

Formally, WNSCA can be described as follows. Let $C = \{T_1, T_2, .., T_n\}$ be the set of terms collected using frequency based techniques, $SC(T_i)$ be the sense count i.e. the number of senses in which the term T_i is defined in the WordNet, and SCT be the sense count threshold. The value for SCT[‡] is selected experimentally as 3 . For each T_i in C, if $SC(T_i) \leq SCT$ then the term T_i is considered as a concept of the domain.

Though the WordNet is a very large database of words, it may not contain all terms. For example, even though the terms "electronic voting machine" and "election official" are relevant to "electronic voting" domain, WordNet does not contain those terms. Hence the sense count for terms($SC(T_i)$) is determined using the following sense count for terms heuristic.

Sense Count for Terms ($SC(T_i)$) is determined by finding the rightmost sub phrase of the term (T_i) present in the WordNet. For a given term(T_i), determine the sense count of the rightmost longest sub phrase from WordNet. Since the term T_i consists of additional modifiers to the sub phrase, the term T_i should be more specific than the sub phrase. So, the sense count of T_i must be less than that of the sub phrase. Here we assume each additional modifier in T_i plays an equal role in determining the specificity. Hence the $SC(T_i)$ is defined as the sense count of the sub phrase divided by the number of additional words in T_i plus 1(for sub phrase). Formally, let $T_i = W_{i1}W_{i2}..W_{in}$ and WN be the collection of terms defined in the WordNet. If $W_{ij}..W_{in} \in WN$ and $W_{ij-1}W_{ij}..W_{in} \notin WN$ then $SC(T_i) = \frac{SC(W_{ij}..W_{in})}{(j-1)+1}$.

For example, the term "minority vote suppression" is not in the WordNet. Even the term "vote suppression" is not present in the WordNet. The word "suppression" has 4 senses. From the above rule, $SC($"minority vote suppression"$) = 4/(2+1) \approx 1$.

[‡]Experiments are conducted with SCT as 4 and 5 also.

Table 3.2: Example Concepts Retrieved by WNSCA+{PE, POP}

WNSCA	WNSCA+PE	WNSCA+POP
voter	election	vote
electronic voting	machine	paper record
provisional ballot	company	paper
county	voter-verified paper trail	poll
manufacturer	year election	provisional ballots
voting	close election	election officials
poll worker	fraud	paper trails
problem	hand recount	2000 election mess
electronic voting machine	local election official	voter confidence
polls	recent election	voting right activist

Since WNSCA tries to filter out the irrelevant terms from the initial cut, it increases the precision of the terms. Along with the irrelevant terms, WNSCA may also eliminate some of the relevant terms because of their high sense count. Hence recall of WNSCA will be poor compared to the frequency based methods. For example, since $SC(\text{“vote”}) = 5$, the term “vote” gets eliminated from the initial cut even if it is retrieved by frequency based methods. To improve the recall without affecting the precision, the following Phrase Ending and Part Of The Phrase heuristics are presented.

WNSCA+PE In the WNSCA+Phrase Ending(PE) method, initially, we take the resultant terms of WNSCA to build the base list of words. For each of the WNSCA terms, head words are extracted to form the base list. For each of the terms extracted from text, head word is verified with the terms in the base list. If its head word is in the base list then the term is considered as a concept of the domain.

Suppose the term “provisional ballot” is retrieved as a concept using WNSCA, PE method retrieves all the terms which end with the word “ballot” also as relevant terms. Some of the such terms would be “meaningless ballot”, “paper ballot”, and “butterfly ballot”.

WNSCA+POP The WNSCA+Part Of The Phrase(POP) method extracts the individual words from terms produced by WNSCA and forms a base word list. For each of the terms in texts, if it contains at least one word from the base list then the term is considered as a concept of the domain.

If the term “provisional ballot” is retrieved using WNSCA then the terms containing words “provisional”, “ballot” or both are retrieved as conceptual terms using WNSCA+POP. Some of those terms would be “ballot box”, “ballot integrity”, and “provisional balloting”, in addition to the terms retrieved by WNSCA+PE method.

The general framework for the extraction of concepts using WNSCA and its extensions is shown in Figure 3.1. To illustrate WNSCA+{PE, POP}, Table 3.2 shows

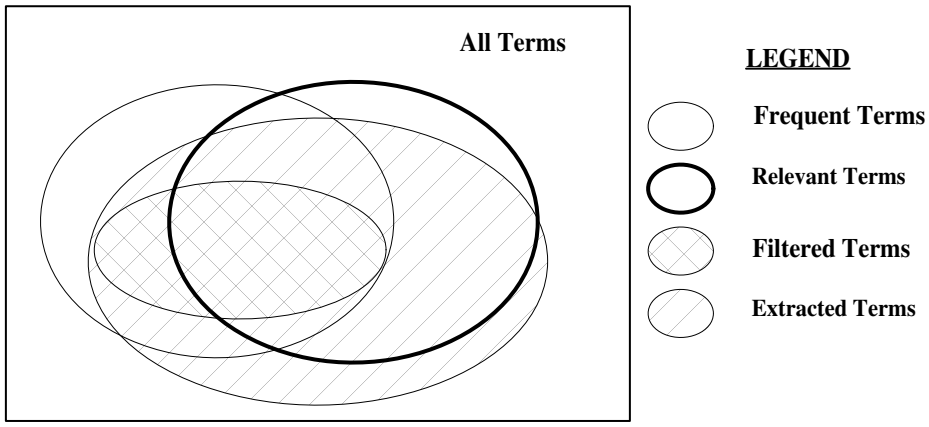

Figure 3.2: Venn Diagram of WNSCA+{PE, POP}

some of the concepts retrieved using each of the methods. In Table 3.2, the first column shows the top 10 concepts retrieved using WNSCA. The second column shows the top 10 new concepts retrieved by WNSCA+PE but not by WNSCA. The top 10 new concepts retrieved by WNSCA+POP, but not by either WNSCA or WNSCA+PE, are shown in the 3^{rd} column.

Figure 3.2 shows the Venn diagram of WNSCA and its variations(PE or POP). In Figure 3.2, filtered terms are the ones obtained with the application of WNSCA. Extracted terms are the ones obtained with the application of POP or PE heuristics on WNSCA resultants. Precision and recall results of the application PE and POP methods for WNSCA resultants are shown in the following section.

Though WNSCA requires an external general purpose knowledge base(WordNet), it does not assume any knowledge about the domain. So WNSCA+{PE, POP} is a domain independent approach. It is also a fully automatic method. The main idea behind WNSCA+{PE, POP} is to extract highly relevant terms of the domain as an initial cut and use individual words of the terms in this set to identify additional relevant terms. WNSCA uses WordNet sense information to obtain the initial cut.

3.5 Evaluation and Results

Results of the of Word Count Approach and WNSCA+{PE, POP} applied on top of raw frequency counting technique are presented in this section. Several experiments are conducted using the *Electronic Voting* domain text. Description of the *Electronic Voting* domain text is presented in the Appendix C. After preprocessing, the program has extracted a total of 768 distinct noun phrases. All these phrases are considered as valid terms. Each of the noun phrases is classified manually either as relevant or irrelevant. It resulted 329 terms as relevant and 439 as irrelevant.

Precision and recall measurements at various frequency thresholds are shown in Table 3.3. In Table 3.3, the "criteria" column specifies the percentage of frequent

Table 3.3: Precision and Recall of Raw Frequency Counting.

Criteria	R	RR	Precision	Recall	F-measure
Top 10%	77	44	57.1	13.3	21.6
Top 25%	193	104	53.9	31.6	39.8
Top 50%	385	203	52.7	61.7	56.9
Top 75%	577	268	46.4	81.4	59.1

Table 3.4: Precision and Recall of Word Count Approaches.

Method	R	RR	Precision	Recall	F-measure
Top 10%					
All Words	122	88	72.1	26.7	38.9
Any Word	329	244	74.2	74.2	74.2
Top 25%					
All Words	266	163	61.3	49.5	54.8
Any Word	462	281	60.8	85.4	71.03

terms considered, the second column("R" i.e. Retrieved) indicates the number of terms retrieved using the specified criteria, the third column("RR i.e. Relevant & Retrieved) column lists the number of relevant phrases are retrieved. Precision column indicates the % of terms retrieved are relevant i.e. Precision $= \frac{RR}{R} * 100$. Recall specifies the % of relevant terms are retrieved i.e. Recall $= \frac{RR}{Relevant} * 100$. Finally, F-measure column gives the combined measure of evenly weighted precision and recall. F-measure is computed as shown in the equation 3.3. For all the experiments in this section, value of *Relevant* is set as 329 after the human evaluation of each of the extracted terms.

$$F - measure = \frac{2 * Precision * Recall}{Precision + Recall} \tag{3.3}$$

From the results of Table 3.3, it is clear that terms with high frequency are relevant to the domain. As expected, in the raw frequency counting method, as the % of phrases considered increases, the precision decreases and the recall increases.

3.5.1 Word Count Approach Evaluation

Table 3.4 shows the results of All Words and Any Word heuristics discussed on the section 3.3. Top 10% and Top 25% labels indicate the terms used for creating the base list. In Table 3.4, first column indicates the name of the method applied and rest of the columns are same as the corresponding columns in the Table 3.3.

Observing Table 3.4, it is clear that Any Word heuristic has better recall compared to All Words heuristic. But surprisingly, Any Word heuristic did not penalize the precision. This is probably because most of the words in the high frequent terms indeed

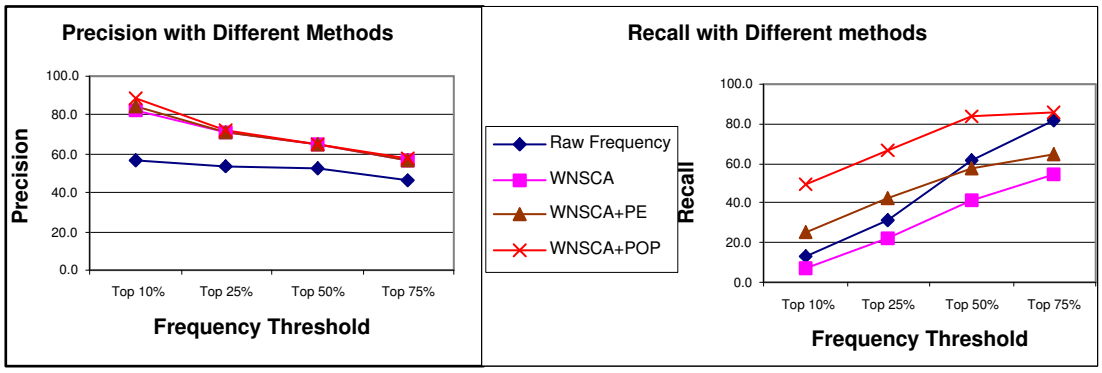

Figure 3.3: Precision and Recall Comparison of WNSCA{+PE, POP}

occurred in relevant terms of the electronic voting domain texts. Experimentation with different domain texts is required to further conclude this observation.

3.5.2 WNSCA+{PE, POP} Evaluation

WNSCA is evaluated with various frequency thresholds (10%, 25%, 50%, and 75%) and with different sense count thresholds i.e. $SCT = 3$, 4, and 5. As the SCT rises from 3 to 5, the precision decreased and recall improved. Since WNSCA is aimed at resulting higher precision, SCT is set to 3. Using WNSCA also, as the frequency threshold increases the precision decreased and recall improved. Precision and recall results of WNSCA at 10% and 25% thresholds with $SCT = 3$ are shown in the Table 3.5. From the initial rows in Tables 3.3 and 3.5, it is clear that among 77 terms retrieved using raw frequency counting, WNSCA has eliminated $48(77 - 29)$ terms as irrelevant. Among the 29 terms retrieved using WNSCA, only $4(29 - 25)$ are irrelevant. It is clear that WNSCA eliminates most of the irrelevant terms. Even though WNSCA has high precision, it filters out some of the relevant terms also. For example, term "vote" is not retrieved using WNSCA because $SCT(\text{"vote"}) = 5$. In this experiment, among the top 10% frequent terms, $19(44 - 25)$ relevant terms are eliminated by WNSCA. Most of these 19 terms and other relevant terms are extracted using WNSCA+PE and WNSCA+POP heuristics. Results of WNSCA+PE and WNSCA+POP are also shown in the Table 3.5. Recall is improved with the application of PE and POP to the resultants of WNSCA. Observing Table 3.5, it is clear that recall is tripled at 10% threshold and doubled at 25% threshold for WNSCA+PE without reduction in precision.

The WNSCA+POP heuristic is applied to extract additional relevant terms. Results of WNSCA+POP indicate that it has approximately two times the recall of WNSCA+PE. Surprisingly, precision is also increased by a small fraction at 10% threshold. This indicates that domain specific individual words occurred more as modifiers rather than the head words. From Figure 3.3, it is clear that PE and POP

Table 3.5: Precision and Recall of WNSCA($SCT \leq 3$).

Method	R	RR	Precision	Recall	F-measure
Top 10%					
WNSCA	29	25	86.2	7.6	13.9
WNSCA+PE	104	90	86.5	27.4	41.6
WNSCA+POP	190	171	90.0	52.0	65.9
Top 25%					
WNSCA	107	79	73.8	24.0	36.9
WNSCA+PE	208	153	73.6	46.5	56.9
WNSCA+POP	321	233	72.6	70.8	71.7

improve recall without reducing precision. But as the frequency threshold increases to 50% or 75% to construct the base list, though the recall improves, precision decreases with the application of PE or POP. This is because as the threshold rises, modifiers or head words of the irrelevant terms also get added to the base list.

Experimental results presented here suggest that using high frequency terms as the base list for WNSCA+{PE, POP} produce most of the relevant terms. Also, it appears that WNSCA+POP is more useful than WNSCA+PE because WNSCA+POP has higher recall than WNSCA+PE with no reduction in precision. These evaluations suggest that use of WordNet is highly beneficial in improving precision. applying PE or POP heuristics produce better results for the concept extraction task.

3.5.3 WNSCA Evaluation with tf.idf

As an alternative approach to obtain the initial cut of the terms, we experimented the input data with tf.idf metric. Terms with high tf.idf value are considered as the members of initial cut. WNSCA is applied to the initial cut. PE and POP are applied to the resultants of WNSCA to extract concepts of the domain. Precision and recall results of the WNSCA+{PE, POP} with tf.idf metric for *Electronic Voting* domain text are shown in the Table 3.6. From the observation of Table 3.6, it is clear that the WNSCA method produces higher precision compared to the tf.idf metric. When we apply PE or POP heuristics to the resultants of WNSCA, large portion of the added terms are indeed relevant. Hence both recall and precision are improved for WNSCA+PE and WNSCA+POP compared to WNSCA.

WNSCA+{PE, POP} Evaluation for TNM Data To further confirm the accuracy of WNSCA, PE, and POP heuristics, we experimented our approaches on *Tender Offers, Mergers, and Acquisitions(TNM)* corpus. Description of TNM corpus is presented in the Appendix C. After preprocessing of the corpus, total number of terms obtained are 183,348. Because of the difficulty in manual classification for such a large term set, only the top 10% (i.e. 18,334) of the terms are considered for experimentation. It took approximately 30 hours to manually label 18, 334 terms as

Table 3.6: Precision and Recall of WNSCA with tf.idf.

Method	R	RR	Precision	Recall	F-measure
Top 10%					
tf.idf	77	56	72.7	17.0	27.6
WNSCA	48	38	79.2	11.6	20.2
WNSCA+PE	110	91	82.7	27.6	41.4
WNSCA+POP	329	206	86.8	54.4	66.8
Top 25%					
tf.idf	193	108	55.9	32.8	41.3
WNSCA	108	67	62.0	20.4	30.7
WNSCA+PE	195	134	68.7	40.7	51.1
WNSCA+POP	317	229	72.2	69.6	70.9

relevant or not. Among the 18,334 terms, 3388 are labeled as relevant. That is only 18.5% of the top 10% terms are relevant to the domain.

For TNM data, top 1% of the terms are used as the initial cut. The WNSCA is applied to the initial cut, then PE and POP are applied to the top 10% terms for generating additional relevant terms. Performance of each of the methods is measured using manually labeled terms as the gold standard. Precision and recall measurements are shown in the Table 3.7.

From the observation of the results in Table 3.7, it is clear that WNSCA has removed the greater percentage of the irrelevant terms. Hence, the precision for WNSCA increased by 15.4% over tf.idf. Using PE and POP heuristics most of the relevant terms are added to the candidate concepts. But at the same time large number irrelevant terms also get added. This resulted in reduction in the precision greatly. This probably due to most of the relevant terms are compound terms and the constituents of compound terms are irrelevant. In addition, we have verified our proposed methods with only the top 10% of the domain terms. Since only 18.5% of the top 10% terms are relevant to the domain, PE and POP methods are not able to extract much of the additional relevant terms from domain texts.

3.6 Summary

One of the major components of ontologies is concepts. In this chapter, we considered the relevant terms occurred in domain texts as the candidates for the concepts. To extract the relevant terms, WNSCA+{PE, POP} method is proposed. The WNSCA method uses information retrieval metrics such as raw frequency counting and tf.idf metrics to obtain the initial cut of relevant terms. To filter out irrelevant terms from the initial cut, The WNSCA uses WordNet sense information. To identify the concepts which are filtered out due to frequency thresholds, the WNSCA+{PE, POP} uses Phrase Ending and Part of Phrase heuristics. These heuristics also produced high accuracy performance.

Table 3.7: tf.idf Results with TNM Corpus(Top 1% Terms)

Criteria	R	RR	Precision	Recall	F-measure
tf.idf	1834	695	37.8	20.5	26.6
WNSCA	1047	557	53.2	16.4	25.1
WNSCA+10%PE	7869	2091	26.6	61.7	37.2
WNSCA+10%POP	11232	2742	24.4	80.9	37.5

The WNSCA+{PE, POP} is experimented with *Electronic Voting* domain and *Tender Offers, Mergers, and Acquisitions* domain texts. Experimental results indicate that the WNSCA+{PE, POP} results conceptual terms with higher precision and recall compared to the raw frequency counting and tf.idf methods. From the observation of the results, it is clear that WordNet sense information is effective in removing the irrelevant terms from the extracted terms and PE and POP heuristics are helpful for adding additional relevant terms.

This chapter presented the WNSCA+{PE, POP} method for extraction of concepts and their evaluations. The subsequent chapters present the methods for finding taxonomic relations and non-taxonomic relations between the concepts.

Chapter 4

Taxonomy Extraction

Various heuristics and their evaluations for extracting conceptual terms are presented in the previous chapter. In this chapter, we present the techniques for finding taxonomic relations between the concepts extracted. As discussed in the literature, the existing approaches for taxonomy extraction are based on either lexico-syntactic patterns or hierarchical clustering methods. It is clear that lexico-syntactic pattern based approaches result in low recall and clustering approaches pose the difficulty in labeling the internal nodes. Also, pattern based approaches find nouns which are in taxonomic relation rather than determining the taxonomic relation for given concepts. To address the problems with the existing approaches, in this chapter we propose three different techniques for finding the taxonomic relations. 1. Using lexical knowledge base, 2. Using terms compound structure, and 3. Learning-based approach.

In lexical-knowledge based technique, contextual senses of the concepts are determined to extract the taxonomic relations of the given concepts. In compound structure based approach, taxonomic relations are extracted from the compositions of terms. In learning-based approach, concepts of the domain are classified into one of the root classes(Unique beginners) defined in the WordNet.

4.1 Using WordNet for Taxonomy Extraction

A WordNet based method for taxonomy extraction for concepts resulted using WN-SCA+{PE, POP} is presented in the Figure 4.1. As presented in Appendix B, WordNet consists of semantic information for lexical categories. More specifically, in WordNet, Hypernym and hyponym are defined for each noun. Since the concepts are domain-specific noun phrases, we can extract taxonomic knowledge for domain concepts from the WordNet. But WordNet is a general purpose knowledge base, which provides semantic information for all possible semantical senses in which a noun can occur. For example, WordNet provides a hypernym for "plant" as "building complex" or as "organism" based on sense 1 or 2 considered respectively. Hence, to make use of the WordNet for extracting taxonomy one needs to identify the domain context of nouns and this necessitates word sense disambiguation. Experiments with two existing word sense disambiguation methods show the limitations of them. We developed a

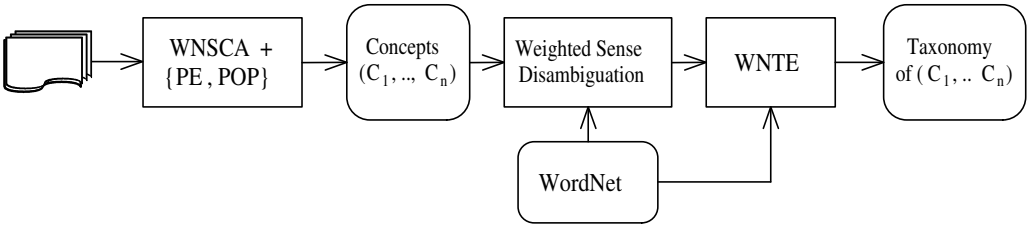

Figure 4.1: WordNet Based Taxonomy Extraction

new method for finding the sense index using their frequency of occurrence in general contexts. The following subsection describes the existing methods and proposed approach for sense disambiguation along with their evaluations with SensEval-3 [Palmer, 2004] data. The final part of this sub-section provides the evaluation on taxonomy extraction using WordNet.

4.1.1 Sense Disambiguation

It is known that a given word consists of various meanings or senses when it is considered alone. The meaning of a word varies widely with the context it is used. For example, the word *plant* consists four different meanings as per WordNet.

- plant, works, industrial plant – (buildings for carrying on industrial labor; "they built a large plant to manufacture automobiles")
- plant, flora, plant life – (a living organism lacking the power of locomotion)
- plant – (something planted secretly for discovery by another; "the police used a plant to trick the thieves"; "he claimed that the evidence against him was a plant")
- plant – (an actor situated in the audience whose acting is rehearsed but seems spontaneous to the audience)

The task of sense disambiguation is to identify which sense of a given word is used in a given context. In natural language processing research, various approaches are developed for sense disambiguation. The existing approaches are broadly classified into supervised, unsupervised, and dictionary based approaches [Manning and Schutze, 1999]. Supervised and unsupervised approaches model the sense disambiguation problem as a classification problem. And these approaches identify the sense of the word among its most popular senses rather than from all possible senses found in the dictionaries. In addition, these approaches require large amount of training data as an evidence to classify the sense of the word. The main idea of the dictionary based approaches is that sense of a target word is identified by matching the lexicon description for a sense of target word with the lexicon descriptions of the contextual words surrounding the target word. Since WordNet is utilized to extract

semantic relations for concepts, we have investigated WordNet based approaches for sense disambiguation. In this dissertation, Lesk's approach and Adapted Lesk algorithm are the two dictionary based approaches explored for sense disambiguation task. Moreover, weighted sense disambiguation is another approach proposed in this dissertation. A brief description of each of these approaches is presented in the following paragraphs.

Lesk Algorithm One of the early approaches for dictionary based sense disambiguation is proposed by Lesk [Lesk, 1986]. Lesk's algorithm disambiguates words based on their definitions in the dictionaries. Words in the definition of each sense for the target word is compared with context words' definitions. For the target word, sense is assigned such that whose definition shares largest number of words with words in the definitions of context words. For example, in the phrase *Time flies like an arrow* , words in each sense definition of *Time* compared with sense definitions of *fly* and *arrow*. Lesk's approach is implemented as part of this research and is evaluated using SensEval-3 [Palmer, 2004] data. Description and how to obtain the SensEval-3 data are presented in Appendix C. Lesk's algorithm relies on the definitions found in the traditional dictionaries to disambiguate the words. Another dictionary based approach explored in this research is Adapted Lesk algorithm. Adapted Lesk algorithm is an extension of Lesk's approach and it uses WordNet as dictionary.

Adapted Lesk Algorithm Adapted lesk algorithm [Banerjee and Pedersen, 2002] is similar to Lesk's algorithm except that Adapted Lesk algorithm uses WordNet information such as definitions of synonyms, hypernyms, hyponyms of words in the context window. In this algorithm, for each target word, a fixed set of words present in the left and right side of the target word are considered as the context window. To find the sense of the target word, all combinations of sense assignments of context words is evaluated. A combination score is computed for each candidate combination. The target word is assigned the sense which gives highest combination score.

The combination score is computed by finding the overlap between context words glosses. While computing the combination score, instead of counting the number of matchings of words in the glosses, longest consecutive sequence of word matches are counted. Each match of words contributes a score equal to square of the words in the sequence. The combination scores are computed for all possible combinations and the one with the highest score is the winner.

This approach gives the senses of the context words as a side effect. As mentioned in [Banerjee and Pedersen, 2002], the procedure is experimented with senseval-2 [Mihalcea, 2001] data and reported that method produced 32% accuracy. The main drawback of the adapted lesk algorithm is that its computational complexity. For a given word, it compares all combinations of senses of target word and context words. Suppose the context window of target word is N and each word has m senses on the average then the number of combination scores need to be computed are N^m. Adapted lesk algorithm is reimplemented as part of this research and tested on senseval-3 [Palmer, 2004] corpus. It resulted 43% average accuracy. Experimental

evaluation of adapted lesk algorithm with senseval-3 data is shown toward the end of section 4.1.1.

Weighted Sense Disambiguation(WSD) Similar to Lesk's approach and adapted lesk algorithm, weighted sense disambiguation approach is also a dictionary based approach. It uses WordNet for sense disambiguation. Along with the glosses of context words, actual words occurred in semantic relations such as hypernym, hyponym, and meronym are used as the available information for sense disambiguation. In addition, WSD assigns weights to each of the sense indexes based on the frequencies of occurrence in the semantically tagged corpora. Count of overlaps for each sense with the context words is multiplied with the assigned weights to compute the score for a given sense. Sense index with the highest score is considered as the sense of the target word.

WSD approach for sense disambiguation can be described formally as follows: Let the target word W appears in one of 1, 2, .., n senses. Sense indexes for W are arranged based on the frequency of use in semantically tagged corpora. Here, sense 1 has highest number of occurrences and n has least number of occurrences for W in the tagged corpora. This indicates that, in a given context of W, it is more likely that sense 1 is used than other senses. Hence weight $Wt(i)$ the for sense index i is computed based on its relative frequency of occurrence as shown in the equation 4.1.

$$Wt(i) = \frac{n+1-i}{\frac{n(n+1)}{2}} \tag{4.1}$$

Another score computed for each of the sense indexes is based on the overlap between the context words information and the given sense index information available in the WordNet. Let C be the set of words collected from WordNet for the given context words. Here, C contains hypernyms, hyponyms, meronyms, and words in the glosses for each of the context words. Let WS_i be the set of words collected from the WordNet for sense i of the target word W and $WS = \bigcup_{i=1}^{n} WS_i$. From C and WS_i, the overlap score is computed as shown in equation 4.2 for each of the senses. Sense index $SI(W)$ for W is computed using equation 4.3.

$$OS(i) = \frac{|WS_i \cap C|}{|WS \cap C|} \tag{4.2}$$

$$SI(W) = \arg \max_{1 \leq i \leq n} Wt(i) * OS(i) \tag{4.3}$$

If more than one sense index has the same $Wt(i) * OS(i)$ value then one with the smaller index is considered as the target sense of W. Experimentation of WSD with the Senseval-3 data is presented in the following section.

Evaluation of Sense Disambiguation Methods Along with the Lesk's approach and Adapted Lesk algorithm, WSD method is evaluated with Senseval-3 data. Accuracy of the algorithms is compared with the baseline approach where the first sense

Table 4.1: Evaluation of Sense Disambiguation Methods

Algorithm	Nouns	Adjectives	Adverbs	Verbs	Total
Wall Street Journal Article 1					
Baseline	70.3	**57.2**	90.9	**50.2**	**59.8**
Lesk Algorithm	43.1	46.3	**100.0**	30.1	38.3
Adapted Lesk Algorithm	41.5	48.4	**100.0**	20.7	33.7
Weighted Sense Disambiguation	**71.4**	50.5	**100.0**	47.7	58.3
Wall Street Journal Article 2					
Baseline	59.8	64.9	**100.0**	54.9	60.1
Lesk Algorithm	47.6	41.5	0.0	32.1	42.3
Adapted Lesk Algorithm	53.6	56.0	**100.0**	34.9	49.9
Weighted Sense Disambiguation	**61.1**	**68.0**	**100.0**	56.1	**61.9**
Brown Corpus Excerpt					
Baseline	**68.4**	**68.9**	**100.0**	**48.7**	**61.0**
Lesk Algorithm	50.1	57.6	66.6	30.8	43.8
Adapted Lesk Algorithm	52.3	67.0	**100.0**	28.1	45.4
Weighted Sense Disambiguation	67.3	68.2	**100.0**	47.7	60.0

Table 4.2: Average Accuracies of Sense Disambiguation Algorithms

Algorithm	Nouns	Adjectives	Adverbs	Verbs	Total
Baseline	66.1	**63.6**	**100.0**	**51.3**	**60.3**
Lesk Algorithm	46.9	48.4	55.3	31.0	41.4
Adapted Lesk Algorithm	49.1	57.1	**100.0**	27.9	43.0
Weighted Sense Disambiguation	**66.6**	62.2	**100.0**	50.5	60.0

is assigned as the target sense for all the words. Senseval-3 test data consists of approximately 5000 words of running texts from two Wall Street Journal articles and an excerpt of the Brown Corpus [Navigli and Velardi, 2005]. A total of 2212 words were manually annotated by a number of linguistic experts, with a reported agreement of 72.5%. This low percent itself demonstrates the inherent difficulty of the task. Since these data sets are made available for public, we have experimented our algorithms with this data. In the test data, words whose sense was marked as unknown in the answer key(U) and words that could not be retrieved in WordNet are identified with sense -1 (wnsn=-1). For evaluation, words with sense -1 are ignored. Accuracy of sense disambiguation for each of the algorithms is shown in Table 4.1. Table 4.1 shows the % of words are classified correctly by four different sense disambiguation algorithms in four different part of speech tags. Columns 1, 2, 3, and 4 show % of words classified correctly in nouns, adjectives, adverbs, and verbs respectively. Last column shows the % of accuracy combining all four part of speech tags words. In Table 4.1, rows 1, 5, and 10 indicate baseline accuracy in three documents listed. Rows 2, 6, and 11 show accuracies of Lesk algorithm. Rows 3, 7, and 12 show the accuracies of the Adapted Lesk algorithm. Finally, rows 4, 8, and 13 show the accuracies of WSD algorithm. From the observation of these results, it is clear that it is very difficult beat the baseline accuracy. Only Weighted sense disambiguation algorithm able to match with baseline accuracy. All the other algorithm performed very poorly compared to base line approach. Table 4.2 shows the average accuracies combining all three articles. Similar to Table 4.1, columns in Table 4.2 shows the average accuracies in the corresponding part of speech tags.

From these results, it is clear that Weighted Sense Disambiguation algorithm performs sense disambiguation comparably to the baseline accuracy. Weighted Sense disambiguation approach has two advantages over Lesk and Adapted Lesk algorithms. One is its performance. WSD has higher accuracy than Lesk and Adapted Lesk algorithms for all four part of speech tags and in all three documents. The other is computation time for disambiguation of senses for each of the words is less. In Lesk and Adapted Lesk algorithms, number of vectors of context words to be verified for each sense of the target depends on the number of context words considered and number of senses for each of the context words. Hence the number of comparisons for each sense is going to be exponential on the number of senses of each of the context words. Where as in Weighted sense disambiguation approach, number of context vectors going to be verified is only one for each sense. Because of these

Table 4.3: Evaluation of the WordNet Based Approach

Method	Total Edges	Correct Edges	Accuracy
Top10P	259	226	87.3
WNSCA10P	180	159	88.3
WNSCA10P+PE	187	166	88.7
WNSCA10P+POP	393	356	90.5
TOP25P	583	499	85.6
WNSCA25P	397	359	90.4
WNSCA25P+PE	408	370	90.6
WNSCA25P+POP	636	591	92.9

two advantages, WSD algorithm is used to identify the senses of the concepts for extracting taxonomies from WordNet.

4.1.2 Evaluation of Taxonomy Extraction Using WSD

Taxonomic relations are extracted from the WordNet by identifying senses of the concepts using WSD. The works as follows. For each of the concepts present in the WordNet, its sense is identified based on the domain texts with WSD. Once the sense of a concept is identified, all words in the hypernym path for the concept defined in the WordNet are considered as the concepts in taxonomic relation. For illustration, partial taxonomic relations of the electronic voting domain extracted from WordNet are shown in the Figure 4.2.

For *electronic voting domain* texts, concepts extracted using WNSCA+{PE, POP} methods are used for finding taxonomic relations. To identify the context words for sense disambiguation, instead of selecting a different set of context words for each concept, we selected all mono-sense concepts as context words. To find the senses of concepts using WSD method, each of the senses WordNet data is compared with each of the mono-sense concepts WordNet data. Using already extracted concepts as context words eliminate the need to extract a separate set of context words for each concept individually. Since concepts obtained from texts are domain specific and representative of the domain texts, all the mono-sense concepts are good candidates as context words for disambiguation.

Evaluation of taxonomic relations extraction from WordNet is performed based on how many edges in the taxonomy path are correct. Since it is difficult to list all possible taxonomic relations among the concepts, it is difficult to compute the recall on taxonomic relations of the domain concepts. Accuracy of the obtained taxonomic relations is computed based on how many of the obtained relations are correct. Evaluation of taxonomy results, with the above metric for measuring the accuracy, are shown in the in the Table 4.3.

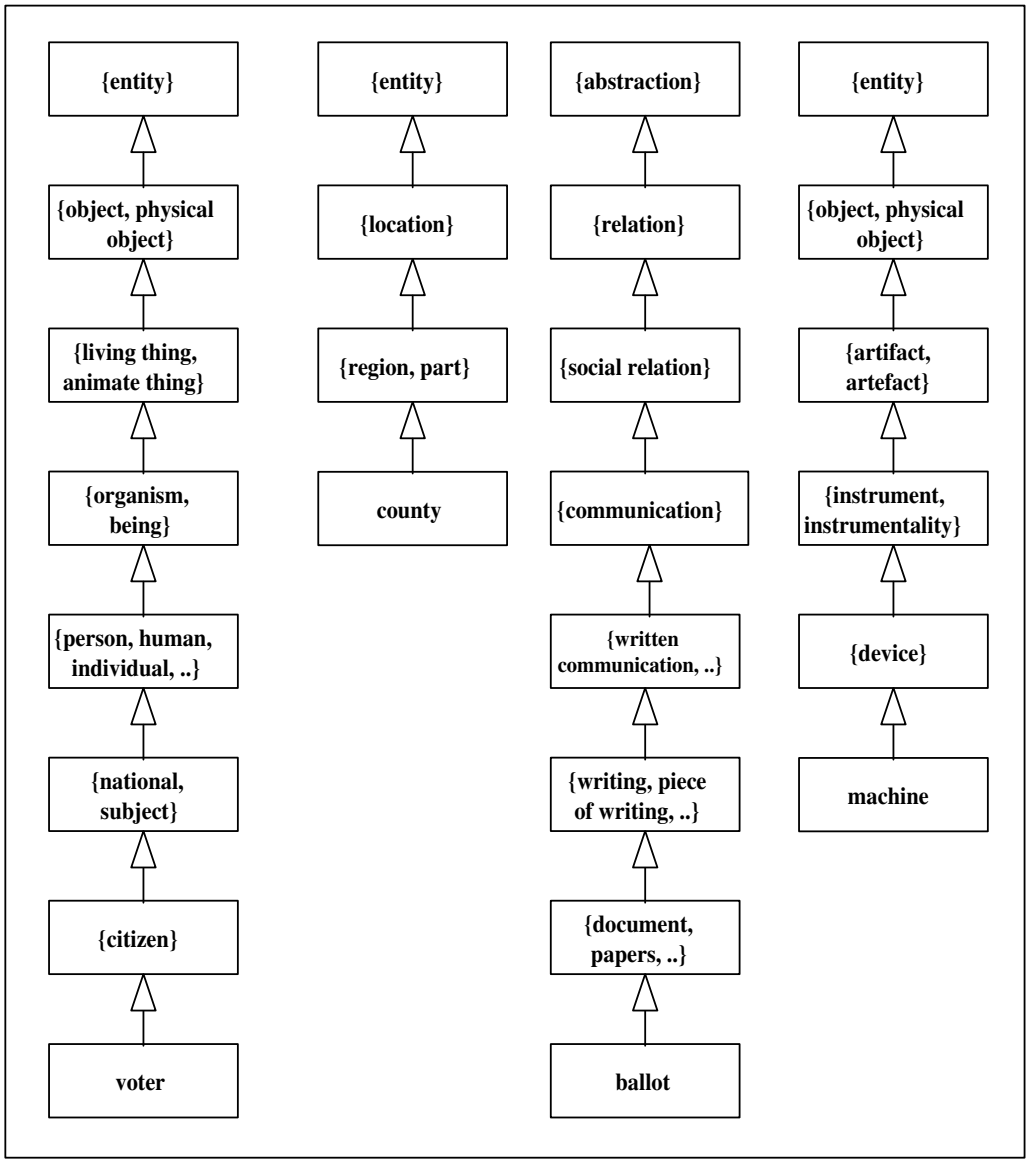

Figure 4.2: Illustration of Taxonomic Relations Extracted from WordNet

In Table 4.3, first column shows the method used for the extraction of conceptual terms from domain text. Second column shows the total number of edges in the hypernym-hyponym relations obtained from WordNet using the conceptual terms. Third column of the Table 4.3 shows count of the edges are correct in the extracted taxonomy. The last column shows the % of edges in the obtained taxonomy are correct.

From these results it is clear that most of the taxonomic relations extracted from WordNet are correct. Also, these results confirm that WSD is suitable for finding the WordNet senses of the concepts. The main constraints of this approach are as follows. Since WordNet is a general purpose knowledge base, some of the domain-specific terms may not exists in WordNet. Another constraint is lack of high accuracy of sense disambiguation algorithm. Detailed discussion of these pitfalls is presented in the following paragraph.

Constraints on Using WordNet for Taxonomy Extraction Since WordNet is a general purpose knowledge base, all of the domain-specific terms may not exists in WordNet. For example the term *phishing* often used in the internet community to refer to counterfeit websites is not listed. Even for some of the words present in WordNet, senses mentioned in the texts are not listed in WordNet. For example, the word *dialog* often used in software manuals to indicate user-interface object is not listed in the WordNet [Pantel and Ravichandran, 2004]. In addition, WordNet does not contain domain specific compound terms. For instance, terms *electronic voting machine*, *polling station* of *electronic voting domain* are not listed in the WordNet.

Another constraint is sense disambiguation problem. If the sense of a conceptual term is incorrectly identified, the whole path of the taxonomic relations obtained are incorrect. Also, it is known that sense disambiguation is a difficult task to achieve very high accuracy.

Because of the sense disambiguation problem and lack of completeness of WordNet, relying on WordNet for automatic extraction of taxonomies is not sufficient. The following sections describes a compound term heuristic and learning techniques for semantic class labeling.

4.2 Compound Term Heuristic

Since many of the domain specific terms do not exist in WordNet, it is difficult to find taxonomic relations for such concepts from WordNet. Many of such terms are compound terms. In general, conceptual terms are compositional in nature. Using the composition of the conceptual terms, one can derive the hierarchical relations between the conceptual terms. For compound terms, head words are extracted from the compound terms and taxonomic relation between the head word and the compound term is labeled. Incidentally, compound term heuristic is also explored in [Buitelar et al., 2005]. For example, for concepts `voting machine`, `electronic voting machine`, *machine*, *ballot*, *paper ballot*, *voting*, and *electronic voting* hierarchical relations obtained using compound term heuristics are shown in Figure 4.3. For concepts

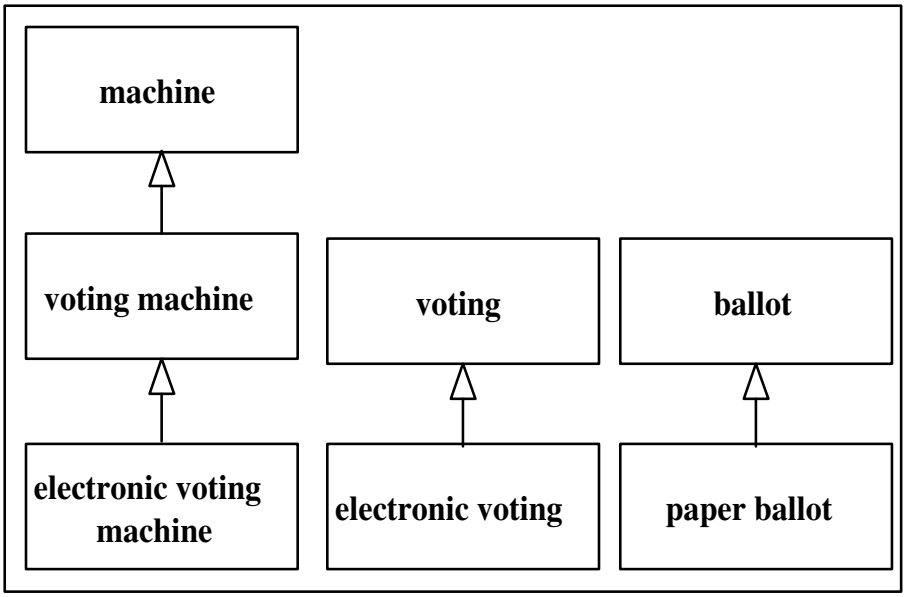

Figure 4.3: Compound Term Heuristic for Taxonomy Extraction

extracted using each of methods presented in Chapter 3, taxonomic relations obtained using the above mentioned heuristic are evaluated. The evaluation results are shown in Table 4.4.

Even though it is a reasonable heuristic to assume the compound term is a hyponym of its head word, evaluation of the above heuristic on *Electronic Voting* data resulted in 65% accuracy. Lack of very high accuracy of taxonomic relations by this heuristic has several contributing factors. One is ill-formed phrases, that is compound terms extracted from texts are not properly formed or are not valid phrases. For example, some of the ill-formed phrases are *machine voter eligibility* and *machine paper ballots*, and *hardware software*. The resultant invalid taxonomic relations are *paper ballots* ← *machine paper ballots* and *software* ←*hardware software*. Another reason is, even though the original phrase is a valid one, the sub phrase(s) obtained after removing the right most word(s) become ill-formed phrases. For example, from the phrase, *voting machine company*, the above heuristic derives taxonomic relation *voting machine company* is a hyponym of *machine company*. But here *machine company* is not a valid one. Taxonomic relations for head words are extracted from WordNet by finding their sense. Using the heuristics described in the previous section, compound terms are assigned as hyponyms of head words.

4.3 Semantic Class Labeling of Concepts(SCL)

Another technique developed for finding taxonomy of concepts is semantic class labeling of concepts. In brief, SCL problem can be illustrated as follows: Let us consider the terms listed below in **(1)** as the concepts extracted from the *Electronic Voting* domain text and **(2)** as the set of predefined top-level semantic classes. SCL's aim is

Table 4.4: Evaluation of Compound Term Heuristic

Method	Total Pairs	Correct Pairs	Accuracy
Top10P	50	31	62.
WNSCA10P	40	26	65.
WNSCA10P+PE	98	60	61.2
WNSCA10P+POP	226	143	63.2
TOP25P	113	55	48.67
WNSCA25P	80	54	67.5
WNSCA25P+PE	153	85	55.6
WNSCA25P+POP	297	174	58.6

to assign one of the classes in **(2)** to each of the concepts or terms listed in **(1)**. By assigning each domain concept to a top-level semantic class, it opens the possibility to derive more general semantic patterns for relationships extraction. It also makes it possible for the domain concepts to inherit semantic attributes defined for top-level classes. Moreover, SCL is useful for word sense disambiguation.

1. *voting machine*, *chairman*, *voter*, *polling station*, *investigation*, *voting*..

2. *Person*, *Artifact*, *Location*, *Action*, *Animal*, *Relation*..

A closely related problem to SCL and well recognized in IE communities is Named Entity Classification(NEC). SCL is different from NEC. NEC's aim is to identify the classes of the instances whereas SCL's is to identify the classes of the concepts. In detail, NEC classifies the proper nouns and numerical information(i.e Named Entities) extracted from texts into a predefined set of categories such as *person*, *location*, *date and time*, and etc. Whereas SCL identifies the super classes of conceptual terms from domain texts. In other words, SCL assigns the classes for common nouns. For example, in the sentence "`The solecisms of George W. Bush, president of the United States`", NEC categorizes `George W. Bush` as an instance of the class *person* whereas SCL labels the concept `president` as the subclass(or hyponym) of *person*. Similarly, from the sentence "`Some states, like Georgia and Maryland, have made the mistake of buying all their machines from one manufacturer,`", NEC identifies `Georgia` and `Maryland` as *locations* and SCL labels `state` as *location*, `buying` as *action*, and `machine` as *artifact*.

Some of the recent works tackled the problem of automatic concepts classification are in [Widdows and Dorow, 2002], [Pantel and Ravichandran, 2004], and [Calvo and Gelbukh, 2004]. In [Widdows and Dorow, 2002], the authors presented an unsupervised technique for extracting the nouns which are semantically related to the given seeds. The method proposed in [Widdows and Dorow, 2002] identifies nouns for the given class which occur as neighbors(in predefined syntactic patterns) with the assumption of knowing few seeds for the class. This approach can be considered

Table 4.5: Attributes for Semantic Classes

Semantic Class	Attributes
Person	name, age, height, weight, status, nature, built, nationality, religion.
Artifact	name, function, made of, manufactured by.
Location	name, size of the area, population size, surroundings.
Action	name, result, performed by.

more as an acquisition method for identifying the nouns of the given class rather than classification of nouns. Also, here the actual classes are more concrete ones such as "crimes", "tools", "diseases", and etc than the semantic classes mentioned in Table B.1 in the Appendix B. [Pantel and Ravichandran, 2004] identifies classes for concepts extracted from text using lexico-syntactic patterns. In this approach, initially, clusters of nouns are formed via computing the mutual-information between the nouns. For each cluster, the class label is identified using nouns in the syntactic patterns such as *N:appo:N*, *N:such as:N* etc. Here also, concept classes are like the ones in [Widdows and Dorow, 2002] rather than the WordNet Unique Beginners. In [Calvo and Gelbukh, 2004], the authors presented a solution to identify the WordNet Unique beginners for nouns. But here the authors classified the nouns based on their definitions in human-oriented dictionaries. In this approach, nouns are classified by forming *is-a* chains with noun definitions, until one of the semantic classes is reached. Even though this approach is useful for classification of concepts, the results indicate only 35.60% accracy. Since this technique uses dictionary definitions to identify the classes, this method can not label the concepts which are not listed in the dictionaries. Hence it may not be portable to specialized domain texts. The following paragraph gives a brief review on applications of SCL.

Applications of SCL SCL is useful in automatic extraction of ontologies and for various natural language tasks as presented below. By defining the attributes for each of the semantic classes as in Table 4.5 and finding the semantic classes for the concepts using SCL allows to inherit the attributes to the extracted concepts.

Semantic class identification reduces the complexity of the word sense disambiguation problem. In general a polysemous term may belong to various classes based on the sense in which it is used. For example, according to WordNet, the word "machine" has six senses. Out of six senses, three of them are of class *artifact*, two senses are of class *group*, and one sense of class *person*. By identifying the semantic class for the given term based on the context in which it is used, one has to verify in only a subset of the senses to find the correct sense for the term.

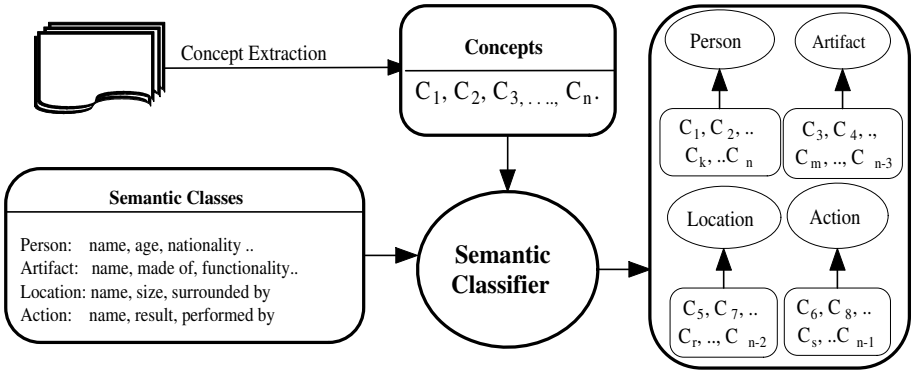

Figure 4.4: Framework for SCL

In addition, SCL is useful for various natural language processing tasks such as Selectional Restrictions [Ribas, 1994], and Identifying Part-Whole Relations [Girju et al., 2003] to name a few.

4.3.1 Supervised Learning for SCL

Here we present a method for concept classification using naive Bayes classifier. The overall framework for semantic classification is shown in in Figure 4.4. Even though in WordNet nouns are classified into 25 unique beginners,we focus on the relatively easier task is classification of concepts into one of *Person*, *Artifact*, *Location*, and *Action* classes. Some example concepts for each of the semantic classes are shown in Table 4.6.

An important issue in applying a learning algorithm for any classification problem is to choose a good set of attributes which provide the distinction between the classes. Also, it is well known that naive Bayes algorithm is quite successful in text categorization using the actual words as the attribute values. With the same intuition, we tried the context words occurring with the conceptual terms as the attribute values for classification. But the results of this approach are not satisfactory. This is probably because of the small size of the context window. We decided to come up with a set of new attributes such that the values of the attributes, for a given instance, can be extracted from text automatically.

From the observation of the concepts in Table 4.6, we noticed that concept's ending characters play a key role in identification of the classes. For example, from the observation of concepts Scient`ist`, Art`ist`, poll work`er` and vot`er`, intuitively, it is clear that concepts ending with `ist`, or `er` can be classified into *Person* class. In the same way, observing concepts, instruct`ion`, vot`ing`, fight`ing`, terms ending with `ing` or `ion` can be categorized into *Action* class. From these two observations, we considered the last two characters of a term as one of the attributes. Another attribute is the actual conceptual term itself. From the observation of the following partial sentences: 1. "*voters who show up and say they are eligible*" 2. "*voting machines to check their*" Even though the coreference resolution problem [Luo et al., 2004] is very difficult to

Table 4.6: Example Concepts for Semantic Classes

Semantic Class	Concepts
Person	poll worker, election official, voter, scientist.
Artifact	voting machine, equipment, paving material, power grid.
Location	city, street, polling station.
Action	interaction, voting, achievement, playing.

solve, we believe the pronoun immediately following the occurrence of actual concept is also useful in identifying the class of the concept. Similarly, from the examples 1. "*at the wrong polling place*" and 2. "*voting machines in" the state*, the preposition preceding the concept in the text can also be a useful attribute in classifying the concepts.

In summary the following four attributes are considered for the semantic classification.

1. Last two characters of the concept.
2. Headword of the concept.
3. Pronoun following the concept.
4. Preposition preceding the concept.

For simplicity, from now on we use the corresponding indexes to refer the attributes. For example, pronoun following the concept is referred as attribute 3.

In extracting the values for each of the four attributes the following rules are followed. Sometimes the pronoun referring to a concept may not appear in the same sentences where the concept has occurred. Hence, the the next sentence is also considered to obtain the values for the 3^{rd} attribute. If no pronoun is present in the following sentence also, then "NPRN"(i.e. no pronoun is present) is set as value for the 3^{rd} attribute. Similarly, if no preposition is present within five words preceding the target term, then "NPREP"(i.e. no preposition is present) is set as the value for the 4^{th} attribute.

To get a better intuition as to how an instance data looks like, some of the concepts along with their attribute values are shown in Table 4.7. In Table 4.7, first column shows the concept names and second column shows attribute values in the order. Attribute values are separated by commas(,). Finally, the 3^{rd} column shows the class label for each of the corresponding concepts. Training data extracted from text is feed to the supervised learning algorithm, naive Bayes Classifier, described below.

The input to the naive Bayes algorithm is n labeled examples of the form (x_i, y_i). Let m be the number of attributes and k be the number of classes. Each x_i is a vector

Table 4.7: Training Instances for SCL

Concept	Attribute Values	Class
intimidation	on, intimidation, NPRN, NPREP	Action
precinct	ct, precinct, their, to	Location
machine	ne, machine, it, NPREP	Artifact
keyboard	rd, keyboard, its, to	Artifact
governor	or, governor, it, NPREP	Person
prison	on, prison, their, NPREP	Location
manipulate	te, manipulate, their, NPREP	Action
county	ty, county, NPRN, in	Location
resident	nt, resident, we, NPREP	Person

of values for the attributes i.e. $x_i =< x_{i1}, x_{i2}, .., x_{im} >$. Each y_i consists one of the values from $Y = \{y_1, y_2, .., y_k\}$ and $p(y_i) = 1/k$. The objective is to find the class y_i of a given example, say $x =< x_1, x_2, .., x_m >$. Let f be a function which maps x to y i.e $f(x) = y$.

$$f(x) = \arg\max_{y_j \in Y} h(x, y_j) \tag{4.4}$$

$$h(x, y_j) = p(y_j) * \prod_{l=1}^{m} p(x_l|y_j) \tag{4.5}$$

$$p(x_l|y_j) = \frac{Count(x_l, y_j) + \alpha}{Count(y_j) + k\alpha} \tag{4.6}$$

$Count(x_l, y_j)$ is the number of occurrences of the attribute value x_l present in the examples with class y_j. $Count(y_j)$ is the number of examples classified as y_j. α is a smoothing parameter and is set $\alpha = 0.1$.

For SCL, $m = 4$ and $k = 4$. As shown above, the naive Bayes approach returns the class such that the product of probabilities for the given instance with the given class is maximum. Experimental results of naive Bayes approach for SCL are presented in Section 4.3.2.

4.3.2 Evaluation of Supervised SCL

Naive Bayes Classifier described in Section 4.3.1 is tested against a set of conceptual terms extracted from *Electronic Voting* domain text appeared in *NY Times*. Six-fold cross validation of the data(1287 instances) resulted 96.8% average accuracy. Among the 1287 instances, many of them posses the same values for each of the corresponding attributes. After the elimination of duplicates, total data consists of

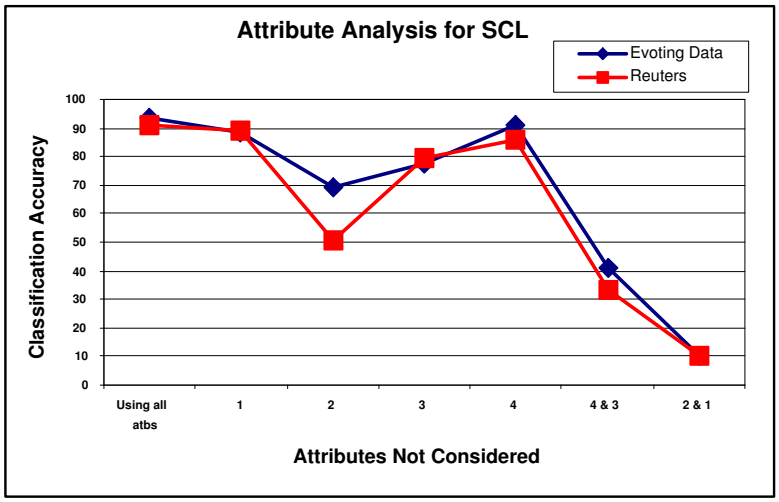

Figure 4.5: Attribute Analysis for SCL

only 622 instances. With the resulted training data, naive Bayes classifier produced the average accuracy of 93.6%.

To further confirm the performance of naive Bayes classifier for SCL, the classifier is experimented with 2326 terms of *Person* category, 447 of *Object* category, 196 of *Location* category, and 351 of *Action* category extracted from the WordNet. Training data is extracted, by searching each of the collected terms, from the Reuters Data [Lewis, 1997]. It resulted 40633 instances. After the elimination of duplicates, the training data size is reduced to 2624(some of the terms collected from WordNet did not present in Reuters data). Six-fold cross validation of the resulting training data produced 91.0% average accuracy.

Evaluation of the Attributes Even though the experiments suggest the presented solution for SCL works quite well, in this section we verify the significance of the attributes both experimentally and statistically. To perform the experimental evaluation, we tested the classifier by considering only a subset of the four attributes. From the observation of Figure 4.5, it is clear that attributes 2 and 3 are key attributes in the classification. When the attributes 3 and 4 are removed individually, there is not much reduction in the classification accuracy. Whereas both 3 and 4 are not considered, the classification accuracy falls below 40%. Even though initially we thought the first attribute(i.e word endings) might play a key role for SCL, Experimental results suggest that it many not be significant for classification.

Along with the empirical tests, we also verified the significance of the attributes using the t-statistic. The statistical t-test for the significance of each of the attributes is defined as follows.

- H_0 : Attribute(i) is not useful for SCL.
- H_a : Attribute(i) may be useful for SCL.

Table 4.8: t-test Results for Attribute Eliminataion

Attribute Index	t-statistic
1	1.59
2	4.12
3	2.48
4	0.82

The t-statistic values are computed considering each instance in the test data as an individual experiment. The t-statistic values with respect to each of the attributes are shown in Table 4.8. From Table 4.8, it is clear that null hypothesis for attributes 2 and 3 can be rejected at 5% level of significance since $t_{\inf,0.05} = 1.96$.

Both the experimental and the statistical results indicate that only attributes 2 and 3 are useful for SCL. These two tests suggest that ending characters attribute is not useful for SCL and the occurrence of such endings may be coincidental. Similarly, the test indicate that the preceding preposition attribute is also not useful for classification. This probably due to that large number of instances, irrespective of their classes, consists of "NPREP" as the attribute value. It means for most of the concepts, there is no preposition preceding within 5 words from the occurrence of concept in the text.

4.3.3 Unsupervised Learning of SCL

The main pitfall of the supervised approaches is that such methods require large amounts of training data. The proposed approach for SCL is based on the intuition that concepts belong to same class occur in similar contexts. Based on the above hypothesis, the following unsupervised algorithm is proposed for identification of the semantic classes of the concepts. To start with, initially, a few(3 to 4) seed concepts are provided for each of the semantic classes. For each of the target concepts, similarity value is computed with the already classified concepts in each of the semantic classes. Semantic class whose members have the highest average similarity with the target concept is assigned as the class for target concept. To compute the similarity between the concepts, contextual words are considered as the features of the concepts. For each concept and feature pair, mutual information is computed. Mutual information determines how often the given concept occurs along with the given feature. Cosine similarity metric using mutual information is used to find the similarity between the concepts.

Formally, algorithm for unsupervised semantic classification can be described as follows. Let $C = C_1, C_2, .., C_n$ be the set of concepts to be classified as members of the semantic classes and $F = f_1, f_2, .., f_m$ be the set of contextual words occur as neighbors(within two words either in left or right) of at least one of the concepts in C. For each concept and feature pair (C_i, f_j) mutual information is computed as follows:

1. for each concept C_i in C
2. for each SC_j in SC
3. $avg_j = \frac{\sum_{C_k \in SC_j} Sim(C_k, C_i)}{|SC_i|}$
4. $SCI = \arg\max_{1 \leq j \leq 4} avg_j$
5. $C_i \in SC_{SCI}$
6. end.

Figure 4.6: Algorithm for Unsupervised SCL

$$MI(C_i, f_j) = \log_2 \frac{C(c_i, f_j)}{C(c_i) * C(f_j)} \tag{4.7}$$

In equation 4.7, $C(c_i, f_j)$ is the count of occurrence of feature f_j with concept c_i as context, $C(c_i)$ is the total count of the occurrence of c_i in domain texts, and $C(f_j)$ is the count of the occurrence of f_j as a neighbor to at least one of the concepts in C. Let $SC = \{SC_1, SC_2, SC_3, SC_4\}$ be the sets of semantic classes defined in the WordNet, where each SC_i is a set concepts belong to the class i. Initially, to start with each SC_i consists of 3 to 4 concepts as seeds. In step 3 of Figure 4.6, function $Sim(C_k, C_i)$ computes the similarity between concepts C_k and C_i.

Experimentation of Unsupervised SCL For experimentation of unsupervised SCL,concepts of the of the four classes *person*, *artifact*, *location*, and *action* are used for classification. Here $|SC| = 4$. A total of 622 instances belong to one of the four classes extracted from *Electronic Voting* domain are used as the test data for classification. Similarity threshold for unsupervised SCL is set as .15. Precision and recall measurements of this experiment are 73% and 24% respectively. From the results, it is clear that unsupervised SCL algorithm is able to classify only a few set of items with high accuracy. We believe further research is needed in finding the most suitable features for classification most of the concepts in to there correct classes.

4.4 Summary

In this chapter, WordNet based approach, compound term heuristic, and naive Bayes classifier are presented for taxonomic relations extraction. WSD approach is presented to find the WordNet senses for the concepts. For each of the concepts extracted using WNSCA+{PE, POP}, Taxonomy is extracted from WordNet by identifying concept's using WSD. Taxonomy extraction approach is evaluated by measuring the count of edges in the extracted taxonomy are in valid taxonomic relation. Experimental results indicate that WordNet based approach results in high accuracy for finding taxonomic relations. Further more, this high accuracy indicates that WSD is able to identify senses for most of the concepts correctly.

Another method presented for taxonomy extraction is compound term heuristic. This method labels the taxonomic relation between compound terms and their head words. Experimental results with the *Electronic Voting* domain indicate that compound term heuristic is useful for finding taxonomic relations in compound terms which are generally not listed in the WordNet.

The other method developed is SCL. SCL is a naive Bayes classifier approach. SCL finds semantic classes of the concepts based on their position of occurrence in the text. Even though SCL is a supervised learning approach, the values for the attributes can be extracted automatically from the concept's position in the text. The proposed naive Bayes solution for SCL results in average accuracies 93.6% and 91.0% for *electronic voting domain* and *Reuters* data respectively. Empirical and statistical evaluations on the significance of the attributes are performed. Experimental results suggest that the naive Bayes solution is quite useful for SCL. SCL, at present, classifies concepts of `person`, `location`, `object`, and `action` classes only. Hence, further investigation is needed to extend SCL for classification of concepts into other classes such `abstraction`, `psychological feature`, etc. We also investigated the clustering based approach for SCL. But experimental results indicate the features considered for unsupervised-SCL are not sufficient.

Chapter 5

Non-Taxonomic Relations Extraction

One of the important and probably the least tackled task in ontology acquisition is extraction of non-taxonomic relations. As mentioned in the literature, most of the existing techniques are focused on extracting concept pairs for a given relation type. Some of those relations are part-whole [Girju et al., 2003] [Berland and Charniak, 1999] and cause-effect [Girju and Moldovan, 2002]. Very few techniques exist for identification of relationships for a given set of concepts.

In this dissertation, relations of the form $C_i \rightarrow Rl \rightarrow C_j$ are considered as the instances for non-taxonomic relations. In $C_i \rightarrow Rl \rightarrow C_j$, concepts C_i and C_j are related and Rl is a relation label indicating the relationship from C_i to C_j. The above relation is represented as an ordered triple (C_i, Rl, C_j). For example, the triple (`voter`, `cast`, `ballot`) indicates a valid non-hierarchical relation from `voter`→ `ballot`. For simplicity from here on we use the word relations to refer to non-hierarchical relations. With the above notion of relations, in this dissertation, we have developed two different methods for extracting the relations. One is by using log-likelihood estimate based on the association between subject, verb, and objects in sentences and the other is using prepositional phrases. Detailed discussion of each of the methods with the experimentation is shown in the following sections.

5.1 The SVO Triples Method

Before going into details of the SVO triples method, we briefly review several similar approaches for extracting non-taxonomic relations.

Some of the existing methods for finding non-taxonomic relations are [Faure and Nedellec, 1998] [Ciaramita et al., 2005] [Kavalec et al., 2004] [Schutz and Buitelaar, 2005]. [Faure and Nedellec, 1998] considers relation extraction problem as learning selection restrictions for verbs. In this method terms occurring with the same verb are clustered and each of the clusters are manually labeled. Whereas methods presented in [Ciaramita et al., 2005], [Kavalec et al., 2004] and [Schutz and Buitelaar, 2005] exploit the syntactic structure and dependencies between the words for relations

extraction. Both [Ciaramita et al., 2005] and [Schutz and Buitelaar, 2005] extract concept pairs which are in pre-specified dependency relations and use the chi-square test to verify the statistical significance on the occurrence of concept pair and the verb together. In [Schutz and Buitelaar, 2005], relation triples are constructed by extracting relevant pairs(predicate and concept pairs). This technique used the football domain texts for experimentation. Ciaramita et al's work is experimented with the molecular biology domain texts. In [Ciaramita et al., 2005], chi-square test is employed to learn the patterns such as $SUBJ \rightarrow \text{bind} \rightarrow DIR\_OBJ$. And the learned patterns are used to extract semantic relations. Kavalec et al's [Kavalec et al., 2004] approach, initially, forms candidate triples(C_1, V, C_2) such that concepts C_1 and C_2 occur within the predefined distance from V in the domain text. Using the triples constructed, labels for the relations between the concepts are identified based on the "above expectation"(AE) measure defined in equation 5.1. This measure emphasizes that if the occurrence of a verb, V, with a given pair of concepts(C_1, C_2) is greater than its occurrence with the individual concepts then the verb V is considered as the candidate label for the relation between the concepts. Here, Kavalec et al used the tourism domain texts for experimentation of the AE measure approach.

$$AE((C_1 \bigwedge C_2)|V) = \frac{P((C_1 \bigwedge C_2)|V)}{P(C_1|V).P(C_2|V)} \tag{5.1}$$

A problem with Kavalec et al's approach is that the AE measure method does not suggest the direction of the relationship. That is, it is not known whether the verb(V) indicates the relation $C_1 \rightarrow C_2$ or $C_1 \leftarrow C_2$. To overcome the problem of relationship direction identification, we considered concept pairs(C_1, C_2) such that the relationship $C_1 \rightarrow C_2$ is ensured before assigning the label for the relationship. In this dissertation, we compared the results of the AE measure with our presented approach.

We consider the problem of identification of relations as two sub problems. One is identification of the concept pairs(C_i, C_j) such that some relationship holds from C_i to C_j. And the other is identification of labels for the relations from C_i to C_j. Concept pairs are extracted based on the concepts' position of occurrence in domain texts. Candidate relationship labels are identified using VF*ICF metric. And log-likelihood ratio method is used to assign the relation labels between for concept pairs. The main advantage of the proposed method is that it is completely an unsupervised technique. That is, it does not require any pre-labeled training data. Also, it is a domain independent approach and does not use any external knowledge bases like WordNet [Miller, 1990].

Overview The SVO Triples method for extracting relations is divided in to the following four steps.

1. Extraction of domain specific concepts.
2. Identification concept pairs(C_i, C_j) such that C_i and C_j are related.
3. Extraction of the candidate labels, Rl, for the relations.

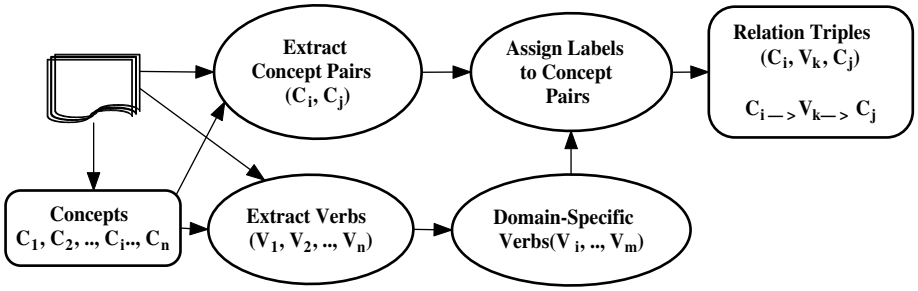

Figure 5.1: Framework for Relations Extraction

4. Assignment of labels, Rl, for the relations between the concepts.

In the SVO Triples method, domain specific concepts are extracted using the methods described in chapter 3. Concept pairs are identified based on the position of occurrence of concepts in texts. Candidate labels for relations are identified from texts and are assigned to concept pairs to obtain the final relations. The overall system architecture for obtaining the relations is shown in Figure 5.1.

To identify concept pairs(C_i, C_j) such that there exists a relationship from C_i to C_j, we maintain two sets CS and CO of concepts. Here, set CS consists of concepts which occur as subjects in sentences. Similarly, CO consists of a set of concepts which occur as objects in sentences. From CS and CO sets, concept pairs of the form (C_i, C_j) are constructed using the following two conditions:

1. $C_i \in CS$ and $C_j \in CO$.

2. There exists a sentence S such that C_i is subject and C_j is an object of S.

Here, condition 1 ensures the direction of the relationship from C_i to C_j. Condition 2 prevents unrelated pairs from getting added to candidate pairs.

To determine the subject and object(s) of a sentence, the MINIPAR [Lin, 1999] shallow parser is used. MINIPAR* produces the dependency relations with 88% precision and 80% recall. Dependency triples produced by MINIPAR are analyzed to identify the subject and object(s) of a sentence. Example, for the sentence "`It is critical that the lawmakers resolve the issue in the next few weeks, before the June adjournment.`", some of the dependency triples produced by MINIPAR are shown in Table 5.1. In Table 5.1, the first and the third columns show two words in grammatical relation. The second column shows the grammatical category of the word in first column, the grammatical relation, and the grammatical category of the word in the third column. In the triple "`resolve V:subj:N lawmaker`", word `resolve` is a verb(`V`), `lawmaker` is a noun(`N`), and the `lawmaker` is the subject(`subj`) of the verb(`V`) `resolve`.

*www.cs.ualberta.ca/ lindek/minipar.htm

Table 5.1: Illustration of MINIPAR Dependency Triples

$Word_1$	GW_1:RL:GW_2	$Word_2$
be	VBE:pred:A	critical
critical	A:subj:N	it
resolve	V:s:N	lawmaker
resolve	V:subj:N	lawmaker
resolve	V:obj:N	issue
few	N:post:PostDet	next
resolve	V:mod:N	week
resolve	V:mod:Prep	before
before	Prep:pcomp-n:N	adjournment
adjournment	N:nn:N	June

Candidate Relation Labels To label the relations between concepts in concept pairs, we first identify the candidate labels for relations and then map the labels to concept pairs. This section describes the method employed to identify candidate labels for relations. It is quite intuitive to believe that verbs which occur more often with the concepts in sentences could be useful for labeling the relations. Thus, it is reasonable to consider frequent verbs as candidates for labeling the relations. But most of the high frequency verbs are of the form *do*, *is*, *have*,..etc; which do not signify much semantic information of the domain. To find the domain-specific verbs, we have defined the VF*ICF metric, similar to the tf.idf used in information retrieval, as shown in equation 5.2. Informally, the VF*ICF metric can be explained as follows. Verbs which occur with only a few set of concepts are more significant compared to the verbs which occur with all the concepts.

$$VF * ICF(V) = (1 + logVF(V)) * log(\frac{|C|}{CF(V)}) \tag{5.2}$$

In equation 5.2, $|C|$ is the total number of concepts, $VF(V)$ is the count of occurrence of verb V in domain texts and $CF(V)$ is the count of the concepts with which the verb V is associated. A verb V is considered to be associated with concept C, if both of them occur in a sentence. Table 5.2 shows the top 10 verbs with their VF*ICF values. Evaluation of VF*ICF metric for identification of domain specific verbs is presented in the experiments section.

Assignment of Relation Labels Another component in relations identification is assigning relation labels to concept pairs. Here we use domain specific verbs extracted using the VF*ICF metric as candidate labels. Assignment of relation labels is performed using log-likelihood ratios. Before going into the details on the formulation for computing the log-likelihood ratios, here, we describe the notations used. Let $S(C_1, C_2)$ be the set of sentences in which both C_1 and C_2 occur. Similarly, let $S(V)$ be the set of sentences in which verb V occurs. Let $n_C = |S(C_1, C_2)|$, $n_V =$

Table 5.2: Top 10 Verbs with High VF*ICF Value

Verb(V)	VF*ICF(V)
produce	25.010
check	24.674
ensure	23.971
purge	23.863
create	23.160
include	23.160
say	23.151
restore	23.088
certify	23.047
pass	23.047

$|S(V)|$, $n_{CV} = |S(V) \cap S(C_1, C_2)|$, and $N = \sum_{i=1}^{n} \sum_{j,k=1}^{|C|} |S(V_i) \cap S(C_j, C_k)|$. Where n is the count of domain-specific verbs and $|C|$ is the count of concepts in relevant concept pairs.

The log-likelihood ratios are computed with the assumption of hypotheses H_1 and H_2 separately. Here hypothesis H_1 formalizes that the occurrence of a verb V is independent of the occurrence of the concept pair(C_1, C_2). Whereas H_2 formalizes that the occurrence of V is dependent on the occurrence of (C_1, C_2) pair.

- **Hypothesis1 (H_1).** $P(V|(C_1, C_2)) = P(V|\neg(C_1, C_2))$
- **Hypothesis2 (H_2).** $P(V|(C_1, C_2)) \neq P(V|\neg(C_1, C_2))$

Now the log-likelihood ratio is computed using the equation 5.3.

$$log\lambda = log\frac{L(H_1)}{L(H_2)} \tag{5.3}$$

Assuming H_1 is true, $P(V|(C_1, C_2)) = P(V|\neg(C_1, C_2)) = p = \frac{n_V}{N}$. The likelihood of H_1 is

$$L(H_1) = b(n_{CV}; n_C, p)b(n_V - n_{CV}; N - n_C, p). \tag{5.4}$$

In the same way, assuming H_2 is true, $P(V|(C_1, C_2)) = p_1 = \frac{n_{CV}}{n_C}$ and $P(V|\neg(C_1, C_2)) = p_2 = \frac{n_V - n_{CV}}{N - n_C}$. The likelihood of H_2 is

$$L(H_2) = b(n_{CV}; n_C, p_1)b(n_V - n_{CV}; N - n_C, p_2) \tag{5.5}$$

In equations 5.4 and 5.5, $b(k; n, x) = \binom{n}{k} x^k (1 - x)^{n-k}$. $L(H_1)$ and $L(H_2)$ are computed assuming binomial distribution of the observed frequencies.

Similar formulation for collocation discovery using log-likelihood ratios is described in [Manning and Schutze, 1999](§5.3.4§) and [Dunning, 1993]. Since we want triples with high $L(H_2)$ and low $L(H_1)$ scores, we multiplied $\log \lambda$ with -2. It is also mentioned in [Manning and Schutze, 1999] that if λ is the likelihood ratio then the

```
1. Input: Candidate concept pairs(C_i, C_j) of CP
2. For each pair (C_i, C_j) in CP
3.       maxLambda =0; relLbl = "";
4.       Extract set L of candidate labels associated with C_i and C_j
5.       For each v in L
6.            vLambda = -2*logλ of (C_i, v, C_j);
7.            if maxLambda < vLambda
8.                 maxLambda = vLambda;
9.                 relLbl = v;
10.      output (C_i, v, C_j);
```

Figure 5.2: Procedure for Relationship Labeling

quantity $-2log\lambda$ is asymptotically χ^2 distributed. For our purposes, we consider the triples(C_1, V, C_2) with high $-2log\lambda$ score as valid non-taxonomic relations of the domain.

Putting It All Together The Subject-Verb-Object Triples (SVO Triples) method is developed, combining individual components described in the previous paragraphs, to extract non-hierarchical relations from domain texts. As presented in Figure 5.2, the algorithm labels the relationship between the concepts for each of the concept pairs. For each concept pair$(C_i,\ C_j)$, a set of candidate labels(L) are extracted. Among candidate labels, the label(v) with highest log likelihood ratio is determined and assigned to the concept pair to output the triple $(C_i,\ v,\ C_j)$.

5.2 Experimentation of the SVO Triples Method

The presented approach is experimented with the *Electronic Voting* domain texts collected from *New York Times* website. From the voting domain texts, a total of 164 concepts are extracted. Experimental results of VF*ICF metric for extraction of domain specific verbs and relationships assignment method(SVO) for concept pairs are shown as follows.

5.2.1 Evaluation of the VF*ICF Metric

Using the extracted concepts and the domain texts, for each of the verbs in the text, the VF*ICF scores are computed. We initially removed the stop words from the extracted verbs. From the remaining verbs, top 20% of them with high VF*ICF scores are considered as candidate labels for the relationships. To evaluate the performance of VF*ICF metric, each of the verbs are manually classified as either relevant or not. Whether a given verb is considered as relevant or not is determined based on the authors knowledge about of the domain. After the manual classification, the precision score for top 20% of verbs is 57%. To provide an intuition on what kind of verbs we

Table 5.3: Illustration of Relevant and Irrelevant Labels

Relevant	Irrelevant
make	say
vote	try
produce	ensure
cast	know
certify	tell
install	help
count	believe
elect	want

considered as relevant, some of the relevant and irrelevant verbs for relation labeling in *Electronic Voting* domain are shown in Table 5.3.

From the observation of the VF*ICF metric results, it is clear that verbs are good candidates for labeling the relations. But we think that using only verbs for labeling the relations may not be sufficient. Further research is needed in this direction.

5.2.2 Evaluation of the SVO Triples Method

Because of the lack of gold standard for identification of the conceptual relationships of the domain, it is difficult to verify the performance of the SVO Triples method. To compute the recall for the presented method, it is required to have all possible relations of the domain. In this experiment we evaluate the performance of the methods using the accuracy of the results produced. Here accuracy is defined as the percentage of the relations obtained being correct. Further more, accuracy of the method is evaluated based on the following three constraints.

- **Constraint 1**. In a concept pair(C_1, C_2), C_1 and C_2 are non-hierarchically related.
- **Constraint 2**. In a triple(C_1, V, C_2), V indicates the relation from $C_1 \rightarrow C_2$ or $C_1 \leftarrow C_2$.
- **Constraint 3**. In a triple(C_1, V, C_2), V is the label for the relationship from $C_1 \rightarrow C_2$ only.

Constraint **1** verifies whether the concepts in the concept pair are non-taxonomically related. Since SVO Triples method extracts concept pairs initially and then assigns the label for the relationship between the concepts, this evaluation is useful to verify the accuracy in extraction of concept pairs. Constraint **2** is useful for identification of whether the assigned label is a valid one for the relation between the concepts in the concept pair without considering the direction of the relationship. Similarly constraint **3** verifies the direction of the relationship. Verification with respect to constraint **3** is also needed because the direction of the relationship should also be

Table 5.4: Example Concept Pairs

Concept Pairs(C_i, C_j)
(election, official)
(company, voting machine)
(ballot, voter)
(manufacturer, voting machine)
(polling place, worker)
(polling place, precinct)
(poll, security)

maintained by the methods developed for automating the ontological relations extraction process. For example in the triple `(voter, cast, ballot)`, the label `cast` indicates the relationship from `voter` to `ballot` but not in reverse.

As mentioned in Section 5.1, two concepts which occur together at least once in a sentence are considered as valid pairs. With the above notion, a total of 184 concept pairs are obtained. Of these pairs, top 20% pairs with high log-likelihood score are considered as the candidate pairs. For illustration, some of the concept pairs such that their constituents are non-taxonomically related are shown in Table 5.4.

Now the verbs with high VF*ICF metric are used to determine the relation labels for each of the candidate pairs. For each pair of concepts extracted, verbs which occur in at least one sentence along with the concepts in the pair and having high VF*ICF value are considered as candidate labels. Among all the candidate verbs, the verb with highest likelihood score is considered as the label for the relationship between the concepts.

In each of the candidate triples(C_1, V, C_2) obtained using the SVO Triples method, C_1 has to be subject and C_2 has to be of object of the verb V. And also V has to have high VF*ICF score. Because of the above restrictions, very few(only 19) triples are obtained. Among the triples obtained, most of them are valid semantic relations. In SVO Triples approach, even though very few relations are obtained, most of them satisfied the constraint **3** i.e. direction of the relationship maintained. For illustration, some of the triples obtained with the SVO Triples approach are shown in Table 5.5.

According to each of the above constraints, we evaluated the SVO Triples approach. Its accuracy with respect to three constraints is shown in Table 5.6. In Table 5.6, the initial column shows the method applied. The second column shows accuracy of the methods according to the constraint **1**. Similarly, columns 2 and 3 indicate accuracies of the corresponding methods with respect to constraints **2** and **3** respectively.

In Table 5.6, the first row shows the results of the AE measure presented in [Kavalec et al., 2004]. The AE measure identifies the candidate triples(C_1, V, C_2) such that C_1 and C_2 appear within a pre-defined distance(8 words) from V. We also implemented the AE measure and applied to our domain texts. From Table 5.6, the results of AE measure indicate that even though it is able to extract related concepts with high accuracy, it performed very poorly in identification of the labels for the relations and in

Table 5.5: Illustration of SVO Triples Method Resultants

Concept(C_1)	Label(V)	Concept(C_2)
machine	produce	paper
voter	cast	ballot
voter	record	vote
official	tell	voter
voter	Trust	machine
worker	direct	voter
county	adopt	machine
company	provide	machine
machine	record	ballot

Table 5.6: Evaluation of AE and SVO Triples Methods

Method	(C_1,C_2)	(C_1, V, C_2)	($C_1 \rightarrow V \rightarrow C_2$)
AE Measure	89.00	6.00	4.00
SVO Triples	89.47	68.42	68.42

maintaining the direction of the relationship as well. From the results in column 2 of Table 5.6, it is clear that VF*ICF measure is useful for filtering some of the irrelevant relation labels.

We believe that the main reasons for such a low accuracy on finding the labels in AE measure are as follows. Concepts in some of the concept pairs occur more as part of compound terms in texts rather than connected by some verb. For example, the compound term `voting machine` occurred more often on its own than the concepts `voting` and `machine` are connected by some verb. Another reason is some of the concepts which occur together more often are connected by a preposition or a conjunction rather than a verb showing the relationship between them. For example, in the sentence `there were constant problems with the hardware and software`, the occurrence of concepts, `hardware` and `software`, does not signify semantic relation between them to label. Sometimes the verb occurring with the concepts in the concept pair may indicate the relation between some other concepts rather than the concepts in the pair.

Enforcing the conditions mentioned in section 5.1, most of the concept pairs obtained are indeed related. Among the obtained concept pairs, very few of them got invalid labels. The few invalid labels might have been obtained due to parse errors. Further more, all of the valid relations obtained using SVO Triples method maintained the direction of the relationship ($C_1 \rightarrow C_2$). Even though most of the relations obtained by the SVO Triples method are valid, the SVO Triples method extracts only a small fraction of the total relations from domain texts. Hence the SVO Triples method gives poor coverage. From the experiments, we believe that even though SVO Triples method is useful for extracting semantic relations, it is not sufficient to find

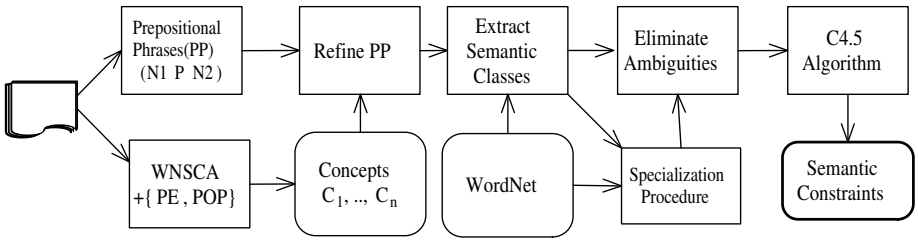

Figure 5.3: Architecture for Learning Semantic Constraints

all of the relations of the domain. Further research is needed to find the relations and relation labels between concepts which does not occur as subject and object(s) in the texts.

Even though the SVO Triples method is able to identify relations with high accuracy, the count of relations obtained does not represent the whole domain. To improve the coverage of non-taxonomic relations, in this dissertation research, we have developed a supervised learning technique to find the semantic relations from prepositional phrases.

5.3 Using Prepositional Phrases

In addition to as subject and object(s) of sentences, concepts may also appear in other syntactic positions in the text such as prepositional phrases and appositives. Also, it is clear that prepositions in the prepositional phrases do indicate semantic relations. For example, from the observation of the phrases `management of company` and `precinct of county`, it is clear that the relation between the concepts `company` and `management` can be labeled as `possess` whereas the relation between the concepts `precinct` and `county` can be assigned as `part of`. From the observation of the above two examples it is clear that each prepositional phrase indicate a different semantic relation based on the occurrence of the concepts in the phrase.

To identify the semantic relations between the concepts appearing in prepositional phrases, in this section, we present a supervised learning technique for finding semantic constraints. The learned semantic constraints are used to label the relations between the concepts occurring in the given prepositional phrases.

The detailed architecture for obtaining the training data and learning the semantic constraints is presented in the Figure 5.3. This supervised approach for labeling the relations is named is SC for PP in the Figure 1.2. A detailed description of each of the tasks listed in this figure is presented with the illustrations in the following sub sections.

Overview The supervised approach for learning semantic constraints for semantic relations can be divided into the following four steps.

1. Extraction of unambiguous prepositional phrases.

2. Selection of attributes and their values for training instances.

3. Eliminating inconsistencies in the training data.

4. Learning rules for labeling the relations.

5.3.1 Ambiguity in Prepositional Phrases

One of the major difficulties in identifying semantic relations from prepositional phrases is the ambiguity in preposition attachment. For example, from the observation of the following two sentences,

1. I bought the shirt with pockets

2. I washed the shirt with soap

it is clear that in sentence 1, *with pockets* describes the *shirt*. However, in sentence 2, *with soap* modifies the verb *wash*. Hence, it is difficult to identify automatically whether the given prepositional phrase modifies the preceding noun or verb. For automatic resolution of this ambiguity, prepositional phrase disambiguation is modeled as (N_1, V_1, P, N_2) in natural language processing. Here the task is to identify whether the given prepositional phrase $[P, N_2]$ to be attached to the noun(N_1) or the verb(V_1). For automatic prepositional phrase disambiguation, various supervised and unsupervised learning methods are presented in the literature [Brill and Resnik, 1994], [Ratnaparkhi, 1998], and [Pantel and Lin, 2000].

Because of the ambiguity of prepositional phrase attachment, it requires to disambiguate the given prepositional phrase before labeling the associated relationship. Since we want to find the relationships from prepositional phrases, in this dissertation, we consider the only unambiguous prepositional phrases for learning the constraints.

Unambiguous prepositional phrases are extracted from the part of speech tagged text using a simplified chunker and the extracted concepts list. The input text is initially part of speech tagged using Brill's part of speech tagger [Brill, 1992]. Words occurring as determiners, adjectives, and cardinal numbers are removed from the part of speech tagged text. From the filtered text, for each occurrence of the prepositional phrase(i.e preposition and the following noun), the preceding text(up to 5 words) is verified for the occurrence of noun, verb, or both. If both noun and verb are present then such prepositional phrase is considered as an ambiguous one. If only a verb occurs in the preceding text, even though such prepositional phrase is not an ambiguous one, those phrases are ignored. Prepositional phrases of the form (V_1, P, N_2) are ignored because all the conceptual terms considered are noun phrases. From the extracted prepositional phrases of the form (N_1, P, N_2), either N_1 or N_2 are must exist in the extracted concepts list.

5.3.2 Relationship Labeling

For each of the unambiguous prepositional phrases selected, the relationship between the concepts occurring in the phrase is need to be labeled. For labeling the relations,

a fixed set of candidate relation labels are selected. The selected candidate labels are listed in Table 5.7. The relation labels are collected observing several semantic relations listed in [Rosario and Hearst, 2001], [Girju et al., 2005], and [O'Hara and Wiebe, 2003].

In Table 5.7, the first column gives the serial number for each relation, the second column gives semantic relation name. Third column gives an example for each relation such that the given relationship holds from concept on the left side to the one on the right. Fourth column gives an example for the corresponding relation from the concept on the right to the one on the left.

As illustrated in Table 5.7, each of the unambiguous prepositional phrases are manually labeled with one of the relations. The assigned label indicates the semantic relation between the concepts in the prepositional phrase along with the direction of the relationship.

5.3.3 Construction of Training Data

Each of the manually labeled prepositional phrases is considered as an instance for supervised learning algorithm. For learning semantic constraints, we used the C4.5 [Quinlan, 1993] decision tree algorithm. To make use of C4.5 algorithm, we need to determine the attributes and possible values for each of the attributes. In our proposed approach, three attributes namely, *Source Class*, *Preposition*, and *Target Class* are defined. For a given prepositional phrase, the *Source Class* attribute takes the semantic class of the noun preceding the preposition, *Preposition* attribute takes actual preposition in the phrase, and *Target Class* attribute takes the semantic class of the noun following the preposition. Example, for the phrase `maker of air-conditioning`, attribute values are *Source Class* = `entity#1`, *Preposition* = `of`, and *Target Class* = `entity#1`. Attributes values for *Source Class* and *Target Class* attributes are extracted from the WordNet by identifying the senses of concepts. Weighted sense disambiguation approach(§4.1.1§) is used to find the senses of the concepts. As shown in the above example, semantic classes of the concepts appear as `class name#sense number`. Here, `class name` is the top-level class in the WordNet and `sense number` is the sense index in the WordNet for `class name`. In the above example `entity#1` indicates the top-level class as `entity` with sense number `1`. Each of the instances in the training data are classified manually with one of the relationship label presented in the Table 5.7.

Eliminating Inconsistencies Training data obtained using the above described procedure may posses inconsistent instances. That is there may exist two or more instances with same attribute values but with different class labels. For example,

1. For `position of player` attribute values are `entity#1`, `of`, `entity#1`, 3
2. For `distributor of equipment` attribute values are `entity#1`, `of`, `entity#1`, 217
3. For `supplier of material` attribute values are `entity#1` ,`of`, `entity#1`, 11

Table 5.7: Semantic Relations for Prepositional Phrases

No.	Semantic Relation	$A \rightarrow B$	$A \leftarrow B$
1	subtype		*transaction of purchase*
2	part of		
3	attribute	*model of computer*	
4	procedure	*construction of plant*	*project in production*
5	perform by	*authorization from director*	*company for distribution*
6	cause		*debt for investment*
7	measurement	*billion in investment*	*amount in dollar*
8	use		
9	location		*bank in city*
10	require		*knowledge of negotiation*
11	produce	*processor of product*	
12	antonym	*breakup of conglomerate*	
13	synonym		
14	performed on	*marketing of satellite*	*market for acquisition*
15	source		*support from Board*
16	member	*head of planning*	
17	possess	*person with deposit*	*tax on income*
18	recipient		*share by affiliate*
19	constraint		
20	associated with	*bid after trading*	
21	temporal	*Working through weekend*	*budget on schedule*
22	collection	*syndicate of investor*	

In example 1, relation label 3 indicates the "attribute" relationship from $A \rightarrow B$. That is `position` is an attribute for the concept `player`. In example 2, relation label 217 indicates the "possess" relationship from $B \rightarrow A$. That is the semantic relation between from `equipment` to `distributor` is "possess". Similarly, the relation label 11 indicates the "produce" relation between `supplier` and `material`.

Even though above examples consist of same values for each of the corresponding attributes but their relation labels are different. These ambiguities need to be eliminated before feeding the training data to C4.5 algorithm. To resolve ambiguities the specialization procedure is applied. Thus the specialization of semantic classes of the concepts in a relation often helps to eliminate ambiguities. Similar specialization procedure is also used in [Girju et al., 2003] for learning constraints for part-whole relations.

For each set of ambiguous examples, *Source Class* attribute value is replaced with immediate hyponym of its current value in the hierarchy of the noun preceding the preposition. If ambiguity is not eliminated, *Target Class* attribute value is replaced with the immediate hyponym of its current value in the hierarchy of noun following the preposition. This process repeated until the ambiguity is resolved or no more specialization can be done. The flow diagram for the specialization procedure is shown in the Figure 5.4.

The specialization procedure can be illustrated using the hierarchy listed in 5.5 as follows. For the above three examples, after the replacement of *Source Class* attribute values, the new attribute values for each of the examples are as follows: 1. `location#1`, `of`, `entity#1`, 3, 2. `object#1`, `of`, `entity#1`, 217, and 3. `object#1`, `of entity#1`, 11. With the new attribute values, ambiguity in example 1 is resolved. If example 1 is not ambiguous with other unambiguous examples, it is added to unambiguous set of training examples with the new attribute values. For eliminating the ambiguity in examples 2 and 3, values for the *Target Class* attribute are replaced with the corresponding hyponyms. After this replacement, attributes values for examples 2 and 3 are as follows: 2. `object#1`, `of`, `object#1`, 217, and 3. `object#1`, `of substance#1`, 11. Now, the ambiguity in the examples 2 and 3 is also resolved. This procedure is repeated for each set of the ambiguous examples. After the resolution of ambiguity in training data, the obtained training data is feed to the C4.5 algorithm to learn the rules.

Extracting Rules for Classification When we feed the training data with the attributes and their values to C4.5 algorithm, it constructs a decision tree with attributes as internal nodes and relation labels as leaves. Decision tree is constructed in such a way that the tree classifies maximum number of instances in the training data correctly. From constructed tree, rules are generated by considering each path from the root as a rule. In each of the paths, attributes with their values at the internal nodes are considered as preconditions and classification label at the leaf is considered as the target label. These rules are sorted based on their accuracy for classification of the new instances.

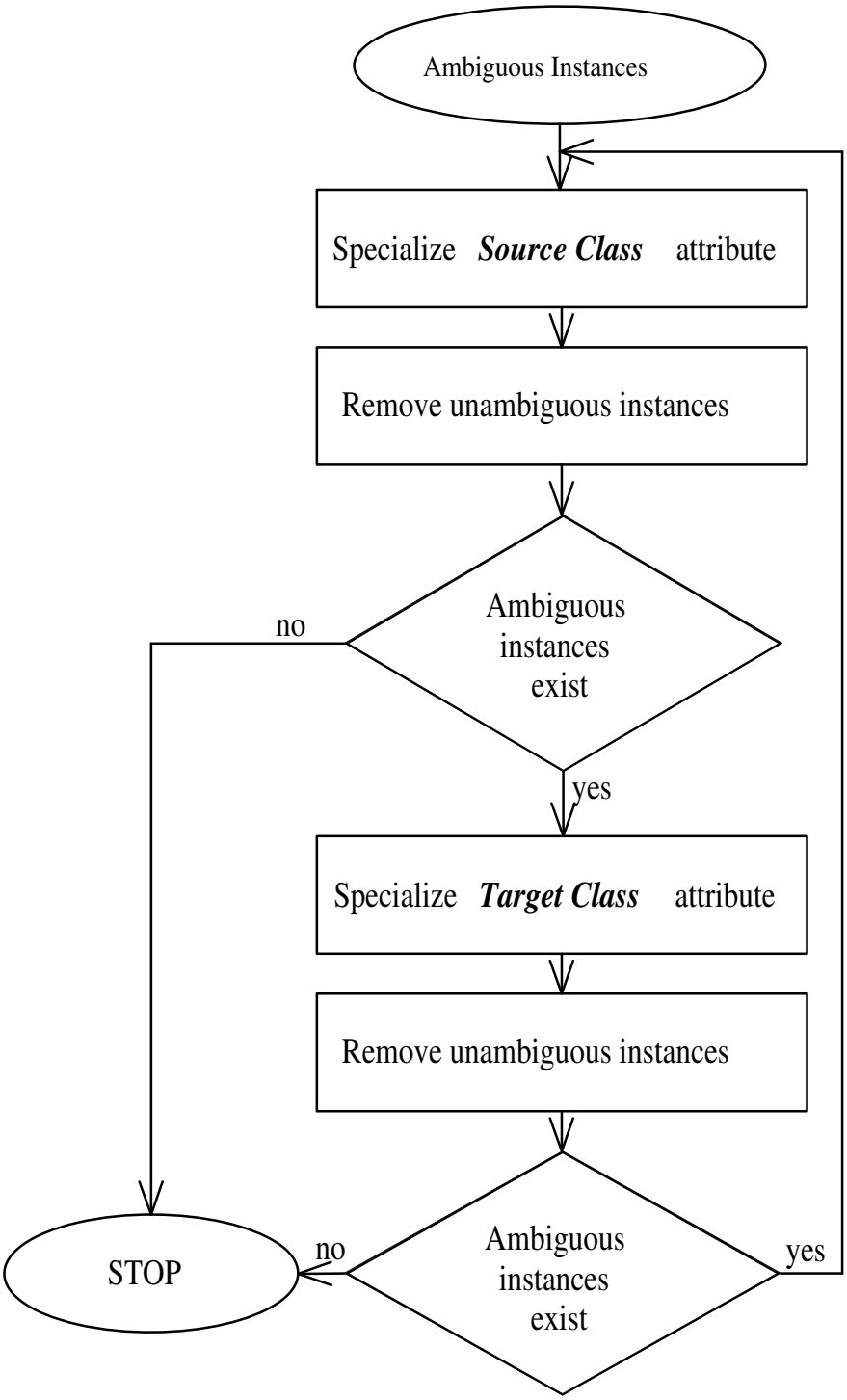

Figure 5.4: Specialization Procedure for Ambiguous Instances

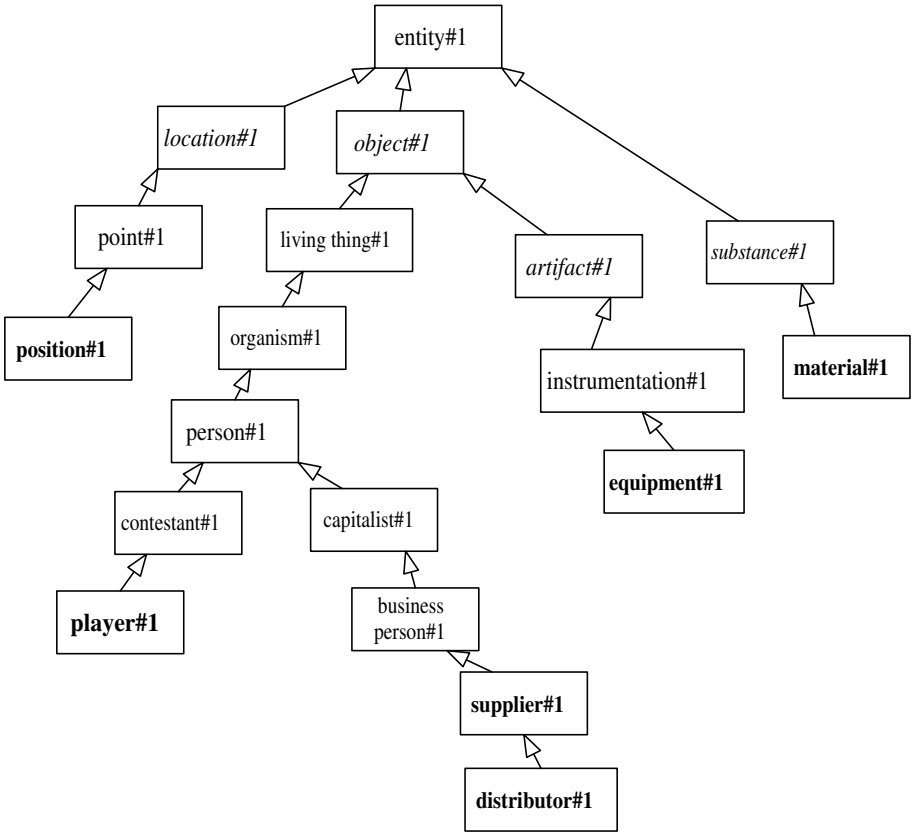

Figure 5.5: WordNet Taxonomy for Prepositional Phrases

For labeling the relation between the concepts in a given prepositional phrase, it is required to get the attribute values. To obtain the attribute values, senses of each of the concepts are initially identified. The taxonomy path for each of the concepts is extracted from WordNet using WNTE(§ 4.1§). Each rule is checked to find whether the given instance satisfies the rule. For this verification, *Source Class* and *Target Class* attributes values in the rule are verified for their existence in the taxonomy paths of the corresponding attributes. If any of the rules preconditions match with the attribute values, then the corresponding label is assigned as the target label for the relationship between the concepts.

This supervised learning algorithm is experimented with *Electronic Voting* domain data and *Tenders, Offers, Mergers and Acquisitions* data. Experimental results are discussed in the following section.

5.3.4 Evaluation of the Learned Constraints

Learning semantic constraints for labeling relations between the concepts in the prepositional phrases is experimented with *Electronic Voting* and *Tenders, Offers, and Mergers* domains text.

As mentioned before the extraction of relations from prepositional phrases suffers from the ambiguity in the prepositional phrase attachment. The prepositional phrases extracted from the text appear in one of the following three forms:1. (N_1, V_1, P, N_2), 2. (N_1, P, N_2), or 3. (V_1, P, N_2). In this dissertation, prepositional phrases of the form (N_1, P, N_2) are considered for learning semantic constraints. From the unambiguous prepositional phrases extracted, ill-formed phrases are eliminated by checking each of the prepositional phrases such that at least one of the nouns exists in the candidate concepts list extracted using techniques presented in chapter 3. From the observation of part-of-speech tagged text, we found that the word "that" is identified as preposition rather than conjunction. Hence the phrases of the form $(N_1, \text{“}that\text{”}, N_2)$ are also extracted as prepositional phrases. But nouns in such prepositional phrases does not posses a valid semantic relation. Hence, phrases of the form $(N_1, \text{“}that\text{”}, N_2)$ are eliminated from the obtained prepositional phrases. To construct the training data, semantic classes are extracted from the WordNet for each of the nouns in the remaining prepositional phrases. Inconsistencies in the training data are eliminated using the specialization procedure. The obtained training data is fed to C4.5 algorithms to learn the semantic constraints for each of the relations. Experimental results and classification accuracies for learned rules are shown in the following sections.

Experiments on Electronic Voting Domain For *Electronic Voting* domain text, statistics of the prepositional phrases extracted are shown in Table 5.8. In Table 5.8, "Counts after Filtering" column indicates the count of prepositional phrases obtained after the elimination of phrases which consist of either preposition as *"that"* or both N_1 and N_2 are not in the candidate concepts list.

Each of the 180 prepositional phrases of the form (N_1, P, N_2) obtained for *Electronic Voting* data, are manually labeled with one of the relations listed in Table 5.7. After manual labeling, 72 of 180 phrases has got label 0. That is 72 phrases does

Table 5.8: Prepositional Phrase Counts for Electronic Voting Corpus

Form	Counts	Counts after Filtering
(N_1, V_1, P, N_2)	392	223
(N_1, P, N_2)	306	180
(V_1, P, N_2)	62	21

Table 5.9: Learned Rules for Relation Labeling

No.	Pre-Condition(s)	Relation	Accuracy(%)
1.	Source Class = act#2	14	63.8
2.	Preposition = in	29	63.4
3.	Source Class = state#4	3	61.2
4.	Source Class = abstraction#6 AND Preposition = of	3	54.6
5.	Source Class = group#1	217	50.0

not indicate any of the semantic relations listed in Table 5.7. Observation of the remaining the examples, there exists a very few(2 or 3) instances for most of the relations. The most important constraint on using the supervised learning algorithms is the requirements of large training data. Because of the lack of sufficient training data, we have selected the instances of the top four frequent relations only as training data. Resultant training data is used to learn the rules for each of the four relations. For learning the rules for each of the relations, two different approaches are developed. One is frequent relations approach and the other is multiple binary classifiers approach.

Frequent Relations Approach The top four frequent relations in the training data are labeled by 3, 14, 29, and 217 in Table 5.7. That is "attribute", "performed on", "location", and "possess" are the most common semantic relations in the training data. Intuitively, it is clear that most of the prepositional phrases does indicate one of the above four relations. Among the 108(180-72) valid prepositional phrases, 64 instances are in one of the above four relations. The resulting 64 instances are fed to the C4.5 algorithm to construct the decision tree and the rules. To evaluate the performance of the learned tree, four-fold cross validation is performed. We separated the 64 instances into four different sets randomly. From the fours sets, three sets are used for training and the remaining set is used for testing. This experiment is repeated such that each of the four sets are used for testing. The average accuracy of the four-fold cross validation of the learned decision tree is (43.8+50.0+68.8+68.8)/4 = 57.8%. From the learned tree rules are extracted. These rules are used for labeling test data. For illustration some of the rules learned for *Electronic Voting* domain are shown in the Table 5.9.

Multiple Binary Classifiers Approach(MBCA) Another approach we implemented for learning rules for semantic relation labeling is MBCA. In this approach, C4.5 algorithm is used for learning rules for each relation separately from the training data. Training data is constructed in such a way that all instances are belong to either the target relation or not. Modified training data is fed to C4.5 algorithm to learn the rules for the target relation. This procedure is repeated for each of the four relations. The learned rules for each of the relations are combined and sorted based on their accuracies. The sorted rule set is used for classification of unknown instances.

Evaluation of the MBCA on *Electronic Voting* domain is performed with four-fold cross validation. Average precision, recall and accuracy of the four-fold cross validation are 42.9%, 38.4 %, and 49.1% respectively. Precision, recall, and accuracy are computed using the equations 5.6, 5.7, and 5.8 respectively. Here, CR_i is the set of phrases such that both the actual label and the label assigned by MBCA is i, A_i is the number of relations in the test data with actual label i, R_i is the number of phrases labeled by the MBCA with the relation label i. Where $i \in \{0, 3, 14, 29, 217\}$.

$$Precision = \frac{CR_3 + CR_{14} + CR_{29} + CR_{217}}{R_3 + R_{14} + R_{29} + R_{217}} \tag{5.6}$$

$$Recall = \frac{CR_3 + CR_{14} + CR_{29} + CR_{217}}{A_3 + A_{14} + A_{29} + A_{217}} \tag{5.7}$$

$$Accuracy = \frac{CR_3 + CR_{14} + CR_{29} + CR_{217} + C_0}{R_3 + R_{14} + R_{29} + R_{217} + R_0} \tag{5.8}$$

Low accuracy of the MBCA compared to the frequent relations approach is due to rules learned for each relation are approximations of the actual tree over the training data. For a binary classifier designed for a specific relation, most of the instances will be negative examples for that relation label, because these are the instances of all other relations labels. Hence the optimal tree learned favored the negative class, which leads to high rejection rate.

Experiments on TNM Data To further confirm the accuracy of the learning algorithm for extracting semantic relations from prepositional phrases, experiments are conducted on *Tenders Offers, Mergers and Acquisitions(TNM)* data also. In the same way, as the input data obtained for *Electronic Voting* domain, similar procedure is applied for extracting the prepositional phrases from TNM domain data also. After removing the phrases which consist of preposition as "that" or at least one of the nouns is not in the conceptual terms extracted using WNSCA+POP (where WNSCA is applied on top 1% of terms and POP is on top 10% of terms).

It is difficult to label the 116350 phrases with corresponding relationship labels manually. To obtain the training data, we randomly selected 2000 prepositional phrases and labeled each of them manually with the one of the relationship labels. Among the 2000 phrases, 692 phrases has got label 0. That is these phrases are either ill-formed, nouns in the prepositional phrase are not valid terms or are not

Table 5.10: Prepositional Phrase Counts for TNM data

Form	Counts after Filtering
(N_1, V_1, P, N_2)	145303
(N_1, P, N_2)	116350
(V_1, P, N_2)	7190

defined in the WordNet. After the elimination of 692 invalid phrases, the remaining 1308 prepositional phrases are used for learning the rules to identify the semantic relationships of the nouns in the prepositional phrases.

The resultant 1308 prepositional phrases, when manually labeled, got 28 distinct relation labels. Maximum possible number of relation labels are 22(from $A \rightarrow B$) +22 (from $A \leftarrow B$) = 44 as listed in Table 5.7. Among the 28 distinct relationships, there exists only 3 to 5 instances for most of the relations. Because of the lack of enough training data for learning the rules for each of the relations listed in the Table 5.7, Here we extracted a subset of the instances such that relationship label for each of the instances must be in one of the top four frequent relationship labels. Here, top four frequent relations are 3, 14, 29, and 217.

Frequent Relations Approach Among the 1308 examples labeled above from TNM data, relationships with labels 3, 14, 29, and 217 constituted 386 examples. These 386 examples are used as the training data and fed to the C4.5 machine learning algorithm. The main objective of this learning approach is to learn the rules for each type of relation that can occur in the prepositional phrase. Accuracy of the learned decision tree on testing the same data used for training has produced 87% accuracy before pruning. For classification of unknown instances it is not known whether a given instance contains one of the above four relations or not. Hence for the remaining 922(1308-386) instances, label 0 is assigned indicating that given instance does not hold any of the four relation labels mentioned above. The modified training data is used to learn the rules for each relations. To identify the accuracy on unknown data, four-fold cross validation is performed. To perform four-fold cross validation, training data is separated into four disjoint sets of equal size. Among the four sets, three sets of training examples are used for training and the remaining set is used for testing. Using the four-fold cross validation, average accuracy of the C4.5 decision tree on test data is $(61.1 + 50.3 + 53.1 + 51)/4) = 53.8\%$. From the obtained decision tree, high accuracy rules are extracted. Some of the extracted rules generated by the C4.5 decision tree are shown in the Table 5.11

The accuracy of the frequent relations approach on *Electronic Voting* and *TNM* domains is summarized in the Table 5.12.

Multiple Binary Classifiers Approach Another approach we employed is the application of decision tree algorithm (C4.5) for each relation type individually. From

Table 5.11: Learned Rules for Relations in TNM data

No.	Pre-Condition(s)	Relation	Accuracy(%)
1.	Source Class = entity#1 AND Preposition = in	29	79.4
2.	Source Class = act#2 AND Preposition = from	0	82.0
3.	Source Class = act#2 AND Preposition = of	14	70.7
4.	Source Class = entity#1 AND Preposition = for	217	91.2
5.	Source Class = event#1	3	61.0
6.	Source Class = entity#1 AND Preposition = in	29	89.9

Table 5.12: Evaluation of the Frequent Relations Approach

Domain Text	Accuracy(%)
Electronic Voting	57.8
TNM	53.8

Table 5.13: Evaluation of the MBCA

Domain Text	Precision(%)	Recall(%)	Accuracy(%)
Electronic Voting	42.9	38.4	49.1
TNM	45.7	57.7	47.8

each learned trees, rules are extracted for each relation type. Rules obtained are combined and sorted based on their classification accuracy in training data.

To estimate the accuracy of the classifier, for each target instance relation label is identified using rules learned. Accuracy is measured by comparing the label assigned by rules with the actual label. Evaluation of MBCA is performed using four-fold cross validation. Average precision, recall, and accuracy on on TNM data are 45.7%, 57.7%, and 47.8% respectively.

The accuracy of the MBCA on *Electronic Voting* and *TNM* domains is summarized in the Table 5.13.

5.4 Summary

For the extraction of non-taxonomic relations between the concepts, the SVO Triples method is presented. The SVO Triples method finds the relations between concepts occurring as subject(s) and object(s) in text. The VF*ICF metric is defined to find relevant verbs as the candidate labels for the relations. The log-likelihood ratios are computed based on the co-occurrence of a label with a given pair of concepts. Each concept pair is assigned a relation label with the highest likelihood ratio. The SVO Triples method is experimented with *Electronic Voting* domain texts. Experimental results are also compared with the AE measure [Kavalec et al., 2004].

We also investigated a method for learning semantic constraints for labeling the relations between the concepts occurred in prepositional phrases. This approach learns the semantic constraints using C4.5 algorithm. The learned constraints are represented in terms of the semantic classes of the concepts in the prepositional phrases. Semantic classes of the concepts are extracted from the WordNet using sense disambiguation technique. This approach is experimented with *Electronic Voting* and *TNM* data. The experimental results indicate the presented method is useful for learning semantic constraints. Because of the lack of sufficient training data, experiments are conducted to learn the constraints for "attribute", "performed on", "constraint", and "posses" relations only. Experimental testing of the learned constraints on *Electronic Voting* and *TNM* produced 57.8% and 53.8% accuracies respectively. Experimental results emphasize that further experiments need to be conducted on larger training data.

Chapter 6

Discussions

The set of techniques developed for automatic extraction of ontological components from domain texts are presented in the preceding chapters. These methods combine the lexical knowledge base, heuristics, machine learning techniques, and statistical approaches. The complete list of the methods along with the tools and techniques utilized by each of the methods are presented in the Figure 6.1. In Figure 6.1, the rectangles represent the methods and techniques developed in this dissertation research and circular rectangles indicate the existing tools and techniques utilized by the ontology extraction methods. The presented techniques are experimented with *Electronic Voting* and *Tender Offers, Mergers, and Acquisitions(TNM)* domain texts. A detailed discussion of the methods listed in the Figure 6.1 for extracting each of the components is presented in the following sections.

6.1 Concept Extraction

One of the major components of ontologies is concepts. In general, the concepts depend on the context in which they are used. Because of the lack of formal notion for a concept, it is difficult to judge whether a given term can be a concept or not. For example, some of the terms may appear as candidates for concepts if observed with the context in which they are used. But when the term alone is considered, it may not be the case. Considering the difficulty in identifying concepts, concept extraction is a difficult task.

In this research, we considered the relevant terms occurring in domain texts as the concepts. To extract the relevant terms, we have used the information retrieval metrics such as raw frequency counting and tf.idf metrics. To further filter out irrelevant terms, we have presented the WNSCA method. The WNSCA uses WordNet sense information. Basically, WordNet sense information provides how many different contexts in which a given term can occur. From this notion, we proposed the hypothesis that a word with fewer number of senses is more relevant to the domain rather than a word with very high number of senses. We also experimentally verified the proposed hypothesis. To identify the concepts which are filtered out due to frequency thresholds, we presented the Phrase Ending(PE) and the Part of Phrase(POP) heuristics.

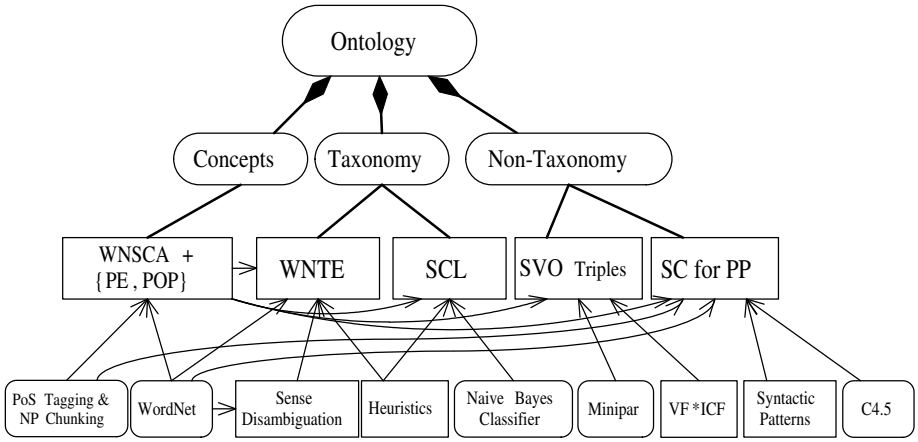

Figure 6.1: Methods and Approaches for Ontology Extraction

These heuristics also produced a high accuracy performance. These heuristics identified more relevant terms compared to the irrelevant terms as concepts of the domain. The combination of WNSCA, PE and POP heuristics is named as WNSCA+{PE, POP} as shown in the Figure 6.1.

To estimate the accuracy of the proposed methods, we experimented with two different domain texts. One is *Electronic Voting* domain texts and the other is *Tender Offers, Mergers, and Acquisitions* data. The main difficulty in evaluating the performance of the proposed methods is the lack of gold standard. To estimate the performance of our methods, we manually classified each of the terms in the domain texts. But manual labeling is a tedious task, if not impossible, for larger domains. Hence, further investigation is needed for automatic evaluations of the proposed methods. Even though the proposed methods achieve high precision and high recall in experiments, we believe further research is needed to improve the accuracy of the methods in larger domains where only a small percent of the terms are relevant to the domain.

The WNSCA+{PE, POP} possesses the following advantages over the existing methods. The WNSCA+{PE, POP} extracts the conceptual terms in full automatic way. It does not require pre-defined syntactic or domain-specific patterns to be defined. Hence, the WNSCA+{PE, POP} is a domain independent approach. On the other hand, since WNSCA+{PE, POP} approach relies on the WordNet for sense information, this approach may not be suitable for specialized domains like medical informatics.

Some of the future directions on the concept extraction task would be combining the synonymously related concepts as one category or one concept. For example, terms, "lecturer", "teacher", "instructor", and "professor" represent a person whose occupation is teaching. Hence, combining all those four terms as a single concept would be more useful for conceptualization of the domain. To combine such terms, one solution would be using conceptual clustering techniques. From our experiments, we realized that relying on conceptual clustering techniques with domain-specific concepts, which use contextual clues as the features, may not produce effective results.

This observation is also mentioned in [Brewster and Wilks, 2004]. Another task is identification of the concepts' instances. Example, `George W. Bush` is an instance for `president`. In information extraction research, various techniques are proposed for identifying the instances for a fixed set of concepts such as `person` or `location` irrespective of the domain. In ontology extraction tasks, concepts vary with the domain text considered and concepts of the domain are more specific terms than the named entities. Hence, we believe instance identification for ontology concepts is a more complex task than named entity recognition. We believe using named entity recognition approaches along with the existing name databases like TAP* might be useful.

In summary, WNSCA+{PE, POP} method is useful to apply on top of frequency based techniques to extract the conceptual terms of the domain. Experimental results also confirmed that sense count information is useful for concept extraction.

6.2 Taxonomy Extraction

In chapter 4, we presented the three different techniques for finding taxonomic relations between the concepts. The presented methods contrast with the existing methods. These methods find the taxonomy of the concepts identified beforehand rather than identifying term pairs present in taxonomic relations. Hence, the presented methods emphasize on both high recall and high precision. Whereas the existing techniques emphasize only on high precision.

For finding taxonomic relations between the concepts, initially, we presented the WordNet based method named as WNTE in the Figure 6.1. This method uses sense disambiguation technique to identify the context of the concepts according to WordNet senses. Taxonomic relations for the given concepts are automatically extracted once the senses of the concept are identified. WordNet based technique relies heavily on the effectiveness of sense disambiguation algorithm. If sense of a concept is incorrectly identified then the whole path of the taxonomy extracted from WordNet is incorrect. In our experimentation with *Electronic Voting* domain concepts, WSD algorithm identified the most of the concepts' senses correctly. Hence the taxonomy extracted from WordNet is also produced a high accuracy.

Even though WordNet is a large knowledge base, it does not contain all domain-specific concepts because of its general purpose nature. For concepts not listed in WordNet, the above presented technique can not find the taxonomic relations. It is known that compound terms are generally not listed in the WordNet. At the same time, concepts may also be compound terms. To find the taxonomic relations for compound terms, compound term heuristic is presented. This heuristic also finds the taxonomic relations with high accuracy.

Further more, to find the taxonomic relations between concepts which are neither in the WordNet nor in compound terms, a naive Bayes classifier and a clustering based approach are presented. In Figure 6.1, the naive Bayes approach for semantic class identification is named as SCL. The SCL possesses the following advantages. The

*http://tap.stanford.edu

attribute values for each of the attributes can be extracted automatically from raw text. It does not use any of the expensive natural language processing techniques. It relies only on the contextual clues from text. This makes the method quite simple and efficient. Even though, naive Bayes approach for classification results high accuracy, it relies on large amount of training data. In automatic ontology learning tasks, since concepts of the whole domain need to be classified, it may not be feasible to manually classify a partial set of concepts for training. Hence, we investigated an unsupervised technique for classification of concepts. Even though unsupervised technique produced high precision, it classified only a few set of concepts. Further research is needed in finding the better features for unsupervised classification of given concepts. Another limitation with supervised and unsupervised approaches is that these approaches classify concepts of `person`, `artifact`, `location`, and `action` classes only.

Existing approaches mentioned in the literature rely on either the repetitive appearance of the conceptual terms or the presence of concepts in the pre-specified patterns for finding taxonomic relations. Hence, these approaches demand very large corpus. Whereas WordNet based approach and compound term heuristic does not rely the on corpus size for finding the relations. Even though the existing methods use a large corpus of text, these methods able to extract only a few concept pairs which are in taxonomic relations. Whereas with the WordNet based approach we can find full taxonomy path from WordNet. The main constraint of the WordNet based approach is its reliance on sense disambiguation. In addition, WordNet based approach can not find the taxonomic relations between the concepts which are not listed in the WordNet.

As part of the future work for finding taxonomic relations, unsupervised techniques with high accuracy are desirable. The presented techniques find the taxonomic relations for each of the concepts separately. Hence, it remains a task to merge the individual paths of the taxonomy to find the complete taxonomy of the domain. Further more, extending supervised and unsupervised methods for other classes such as `abstraction`, `psychological feature`, and `natural phenomenon` is also essential.

6.3 Non-Taxonomic Relations Extraction

Two different techniques are developed for extracting non-taxonomic relations between the concepts. One is the SVO Triples method and the other is learning semantic constraints for relation labeling between the concepts in the prepositional phrases.

In the SVO Triples method, relations are extracted by identifying the subject(s) and object(s) of sentences. the candidate relation labels are identified using verbs. The VF*ICF metric is defined to extract the relevant verbs. But VF*ICF metric produces only 57% accuracy in identifying the relevant labels. The resultant accuracy shows that better methods are needed for identification of domain specific verbs or labels. Further more, we believe considering only verbs as candidates for relation labels is not sufficient. For the extracted concept pairs, relation labels are assigned based on their likelihood ratios. Experimental results with the *Electronic Voting*

domain text show that the SVO Triples method identified the non-taxonomic relations with high accuracy.

As mentioned in the chapter 2, most of the existing methods extract relations between the instances of the named entities or find the concept pairs which are in pre-specified semantic relation. Very few techniques find the relations between the given concepts. These techniques expect concepts to appear in pre-learned dependency patterns. These approaches are similar to our proposed SVO Triples method. The SVO Triples method expects the concepts to appear in subject or object positions in text. One of the existing approaches explored and implemented in this research is the AE measure. The AE measure finds the non-taxonomic relations based on the conditional probabilities. The main disadvantage of the AE measure is it is not able to identify the direction of the relationship. This constraint is resolved in SVO Triples method by selecting concept pairs which follow pre-specified conditions. Even though the SVO Triples method able to identify relations with high accuracy, the count of relations obtained does not represent the whole domain.

Because of the lack of coverage for the SVO Triples method, another technique investigated, in this dissertation, is finding relations between the concepts based on their occurrence in prepositional phrases. This approach is named as SC for PP in Figure 6.1. The presented approach learns the semantic constraints for labeling the relations between the concepts appearing in prepositional phrases. The semantic constraints are learned using C4.5 supervised learning algorithm. The main constraint of this approach is the requirement of large training data. Hence it requires large amount of manual effort to construct sufficient number of instances for each of the relations. Because of this requirement, we trained the system to learn the constraints for only four different relations. Also, the training data used for learning is very small(386 instances). In further research, testing of the presented method on larger data is needed. From the accuracy results of the classifier, it is clear that using only WordNet hierarchy as attribute values is not sufficient. The preprocessing of the extracted prepositional phrases may be needed to obtain the representative instances for learning the constraints.

Another constraint with the presented method is the learned semantic rules consist of nouns in the WordNet hierarchy. Hence, for classification of an unknown instance, it is required to find the taxonomy hierarchy for the each of the concepts in the prepositional phrase. As mentioned in chapter 4, senses of the concepts need to be identified to find the hierarchy from WordNet. Again as mentioned before, it is very difficult to achieve high accuracy for sense disambiguation using dictionary based approaches. In addition, sometimes WordNet does not contain the sense described in the text or some of the concepts appearing in prepositional phrases are not listed in the WordNet. Hence, it is difficult to find the taxonomy for concepts whose senses are not described in the WordNet or which are not listed in the WordNet.

Further more, the presented approach learns the semantic constraints for "attribute", "performed on", "constraint", and "posses" relations only. In general, prepositional phrases express other semantic relations as listed in the Table 5.7. Hence, further investigation is needed for finding the constraints for other relations also. Further more, as a new direction, learning algorithms can be developed to learn

the constraints for each of the relations considering phrases with the same preposition only.

In summary, this chapter provided the detailed discussion along with the future directions on the WNSCA+{PE, POP} for concept extraction, the WNTE and SCL for taxonomy extraction, and the SVO Triples method and SC for PP for non-taxonomic relations extraction. The following chapter concludes the dissertation.

Chapter 7

Conclusions

In this dissertation research, the WNSCA+{PE, POP}, WNTE, SCL, SVO Triples, and SC for PP methods are developed for extracting ontological components. These techniques utilize information retrieval metrics, WordNet, machine learning methods, term heuristics, and statistical methods. In addition, the presented techniques does not use the expensive natural language processing techniques such as deep parsing. Experimentation of the each of the approaches is also performed.

For the extraction of concepts Word Count Approach(WCA) and WNSCA+{PE, POP} are developed. The WCA and WNSCA+{PE, POP} rely on the compound structure of the terms. In addition, the WNSCA+{PE, POP} utilizes sense information from the WordNet. Experimentation with the *Electronic Voting* and *TNM* domain texts show that the WNSCA+{PE, POP} produces both high precision and high recall over the traditional raw frequency counting and tf.idf metrics. Further more, it is a domain independent and a fully automatic approach. Hence, the WNSCA+{PE, POP} is an effective approach for extracting the concepts. As part of the future work, concept extraction task can be extended to cluster synonymously related concepts. Also, instances for each of the concepts can be extracted from text.

The WNTE, compound term heuristic, and SCL are developed for extracting the taxonomic relations between the concepts. The WNTE extracts the taxonomic relations by identifying the senses of the concepts using the WSD technique. The WSD is a dictionary based sense disambiguation technique developed as part of this research. The accuracy of WNTE on experimentation with *Electronic Voting* domain concepts produced 92.9%. This high accuracy indicates that WNTE is effective in domain-specific taxonomic relations extraction from WordNet. Since WNTE relies on the WSD, the high accuracy of WNTE indicates that WSD is able to identify the most of the concepts' senses correctly. The compound term heuristic is presented to find the taxonomic relations between compound terms and their head words. This heuristic also produced high accuracy results for *Electronic Voting* concepts. The other technique presented for taxonomy extraction is SCL. The SCL is a naive Bayes classifier for semantic classification of concepts. The six-fold cross validation of SCL on *Electronic Voting* domain and *Reuters* data produced 93.6% and 91% accuracies respectively. Even SCL classifies concepts with high accuracy, presently, it classifies concepts of `person`, `organization`, `artifact`, `location` classes only. As part of the

future work, SCL needs to be extended to classify concepts of other classes such as `psychological feature` and `abstraction`.

Finally, non-taxonomic relations are extracted using the SVO Triples and SC for PP methods. The SVO Triples method finds the relations between the concepts appearing as subjects and objects in the text. This method labels the relations based on the likelihood ratios of the concepts' occurrence with the candidate labels. The VF*ICF metric is developed to determine the candidate labels from the verbs appear in texts. The SVO Triples method labels the relations such that the direction of the relationship between the concepts is also maintained. The experimental evaluation of the SVO Triples method on *Electronic Voting* domain produced 68.4% accuracy. The high accuracy indicates that SVO Triples method is an effective approach for finding the non-taxonomic relations between the concepts.

The SC for PP is the another method investigated for finding non-taxonomic relations. This method learns the rules for labeling the relations between the concepts based on their occurrence in prepositional phrases. The SC for PP method utilizes the WordNet hierarchy of the concepts and C4.5 machine learning algorithm for learning the semantic rules. The learned rules are used to label the relations between concepts appearing in the target prepositional phrases. The four-fold cross validation of SC for PP on *Electronic Voting* and *TNM* domains produced 57.8% and 53.8% accuracies respectively. These low accuracies are due to the small size of the corpus. These results emphasize that the requirement for large training data. Because of the lack of sufficient training data, at present SCL for PP learns the semantic constraints for "attribute", "performed on", "location", and "possess" relations only. As part of the future work, the SC for PP needs to be extended to learn the rules for other relations also.

In summary, we have proposed the WNSCA+{PE, POP} technique to extract concepts utilizing WordNet sense information. WordNet based approach, compound term heuristic, and naive Bayes and clustering approaches for semantic classification are presented for extracting taxonomic relations between the concepts. Finally, the SVO Triples method and SC for PP are presented for finding non-taxonomic relations between the concepts. The presented methods are evaluated with *Electronic Voting* and *TNM* domain texts. Experimental results indicate that the presented methods are effective for automatic extraction of ontological components.

Bibliography

[Agichtein and Gravano, 2000] Agichtein, E. and Gravano, L. (2000). Snowball: extracting relations from large plain text collections. In *Fifth ACM Conference on Digital Libraries*.

[Banerjee and Pedersen, 2002] Banerjee, S. and Pedersen, T. (2002). An adapted lesk algorithm for word sense disambiguation using wordnet. In *Third international conference on intelligent text processing and commputational linguistics*.

[Bechhofer et al., 2001] Bechhofer, S., Horrock, I., Goble, C., and R., S. (2001). OilEd: A Reason-able ontology editor for the semantic web. In *Joint German/Australian Conf. on Artificial Intelligence*.

[Berland and Charniak, 1999] Berland, M. and Charniak, E. (1999). Finding parts in very large corpora. In *Annual meeting of Association of Computational Linguisitcs*.

[Berners-Lee et al., 2001] Berners-Lee, T., Hendler, J., and Lassila, O. (2001). The semantic web. In *Scientific American*.

[Brewster and Wilks, 2004] Brewster, C. and Wilks, Y. (2004). Ontologies, taxonomies, thesauri: Learning from texts. In *The use of computational linguistics in the Extraction of Keyword Information from Digital Library Content workshop*.

[Brill, 1992] Brill, E. (1992). A simple rule-based part-of-speech tagger. In *Third Conference on Applied Natural Language Processing*.

[Brill and Resnik, 1994] Brill, E. and Resnik, P. (1994). A rule-based approach to prepositional phrase attachment dismbiguation. In *Computational lignguistcs*.

[Buitelar et al., 2005] Buitelar, P., Cimiano, P., and Magnini, B. (2005). *Ontology learning from text: methods, evaluation and applications*. IOS Press.

[Calvo and Gelbukh, 2004] Calvo, H. and Gelbukh, A. (2004). Extracting semantic categories of nouns for syntactic disambiguation from human-oriented dictionaries. In *Computational Linguistics and Intelligent Text Processing*.

[Caraballo, 1999] Caraballo, S. A. (1999). Automatic construction of hypernym-labeled noun hierarchy from text. In *Association of Computational Linguistics*.

[Cederberg and Widdows, 2003] Cederberg, S. and Widdows, D. (2003). Using lsa and noun coordination information to improve the precision and recall of the hyponymy extraction. In *Conference on Natural Language Learning*.

[Chris, 2006] Chris (2006). http://brainplanet.com/index.php/ambiguous prob solution. Technical report.

[Ciaramita et al., 2005] Ciaramita, M., Gangemi, A., Ratsch, E., Saric, J., and Rojas, I. (2005). Unsupervised learning of semantic relations between concepts of a molecular biology ontology. In *International Joint Conference on Artificial Intelligence*.

[Daille, 2003] Daille, B. (2003). Conceptual structuring through term variations. In *ACL Workshop on Multiword Expressions: Analysis, Acquisition and Treatment*, pages 9–16.

[Dunning, 1993] Dunning, T. (1993). Accurate methods for the statistics of surprise and coincidence. *Computational linguistics*, 19:61–74.

[Farquhar et al., 1997] Farquhar, A., Fikes, R., and Rice, J. (1997). The ontolingua server: A tool for collaborative ontology construction. *International Journal of Human-Computer Studies*, 46:707–727.

[Faure and Nedellec, 1998] Faure, D. and Nedellec, C. (1998). Asium: Learning subcategorization frmaes and restrictions of selection. In *10*sup *th European Conference on Machine Learning*.

[Fotzo and Gallinari, 2004] Fotzo, H. N. and Gallinari, P. (2004). Information access via topic hierarchies and thematic annotations from document collections. In *International Conference on Enterprise Information Systems*, pages 69–76.

[Frank et al., 1999] Frank, E., Paynter, G. W., Witten, I. H., Gutwin, C., and Nevil-Manning, C. G. (1999). Domain specific keyphrase extraction. In *16th International Joint Conference on Artificial Intelligence*, pages 668–673.

[Gelfand et al., 1998] Gelfand, B., Wulfekuhler, M., and Punch, W. (1998). Automated concept extraction from plain text. In *AAAI Workshop on Learning for Text Categorization*.

[Girju et al., 2003] Girju, R., Badulescu, A., and Moldovan, D. (2003). Learning semantic constraints for the automatic discovery of part-whole relations. In *Human Language Technologies and North Ameircan Association of Computational Linguisitcs*, pages 80–87.

[Girju and Moldovan, 2002] Girju, R. and Moldovan, D. (2002). Text mining for causal relations. In *15*sup *th international Florida Artificial Intelligence Research Society Conference*, pages 360–364.

[Girju et al., 2005] Girju, R., Moldovan, D., Tatu, M., and Antohe, D. (2005). On the semantics of noun compounds. *Computer speech and language*, 19:479–496.

[Gómes-Pérez et al., 2004] Gómes-Pérez, A., Fernádez-López, M., , and et al. (2004). *Ontological Enngineering: with examples from the areas of Knowledge Management, e-commerce and the Semantic Web*. Springer.

[Gruber, 1993] Gruber, T. R. (1993). Toward principles for the design of ontologies used for knowledge sharing. *Journal of Human Computer Studies*, 43(5):907–928.

[Guarino and Giaretta, 1995] Guarino, N. and Giaretta, P. (1995). Ontologies and knowledge bases: Towards a terminological clarification. In *Towards Very Large Knowledge Bases: Knowledge Building and Knowledge Sharing*. IOS Press.

[Hasegawa et al., 2004] Hasegawa, T., Sekine, S., and Grishman, R. (2004). Discovering relations among named entities from large corpora. In *Association of Computational Linguistics*.

[Hearst, 1992] Hearst, M. (1992). Automatic acquisition of hyponyms from large text corpora. In *14*sup*th International Conference on Computational Linguistics*.

[Hodges et al., 1997] Hodges, J. C., Horner, W. B., Webb, S. S., and Miller, R. K. (1997). *Hodges' Harbrace Handbook*. Holt Rinehart and Winston.

[Iwanska et al., 1999] Iwanska, L., Mata, N., and Kruger, K. (1999). Fully automatic acquisition of taxonomic knowledge from large corpora of texts: Limited syntax knowledge representation system based on natural language. In *11*sup*th International Symposium on Methodologies for Intelligent Systems*, pages 430–438.

[Jacquemin, 1996] Jacquemin, C. (1996). A symbolic and surgical acquisition of terms through variation. In *Connectionist, Statistical, and Symbolic Approaches to Learning for Natural Language Processing*, pages 425–438.

[Kashyap et al., 2004] Kashyap, V., Ramakrishnan, C., Thomas, C., Bassu, D., Rindflesch, T. C., and Sheth, A. (2004). Taxaminer: An experimental on framework for automated taxonomy bootstrapping. Technical report, University of Georgia.

[Kavalec et al., 2004] Kavalec, M., Maedche, A., and Svatek, V. (2004). Discovery of lexical entries for non-taxonomic relations in ontology learning. In *SOFSEM*.

[Lenant, 1990] Lenant, D. B. (1990). Cyc: Toward the programs with common sense. *Communications of the ACM*, 33(8):30–49.

[Lesk, 1986] Lesk, M. (1986). Automatic sense disambiguation using machine readable dictionaries: How to tell a pine cone from an ice cream cone. In *Special interest group for design of communication*.

[Lewis, 1997] Lewis, D. D. (1997). Reuters-21578 text categorization test collection readme file manuscript. In *http://www.davidlewis.com/resources/testcollection/reuters21578*.

[Lin, 1999] Lin, D. (1999). Minipar: a minimalist parser. In *Maryland linguistics colloquium*.

[Luo et al., 2004] Luo, X., Ittycheriah, A., Jing, H., Kambhatla, N., and Roukos, S. (2004). A mention-synchronous coreference resolution algorithm based on the bell tree. In *Association of computational linguistics*.

[Maedche and Volz, 2001] Maedche, A. and Volz, R. (2001). The ontology extraction and maintenance framework text-to-onto. In *International Conference on Data Mining*.

[Manning and Schutze, 1999] Manning, C. and Schutze, H. (1999). *Foundations of Statistical Natural Language Processing*. MIT Press.

[Mihalcea, 2001] Mihalcea, R. (2001). www.senseval.org. In *Second international workshop on evaluating word sense disambiguation systems*.

[Miller, 1990] Miller, G. A. (1990). Wordnet: An on-line lexical database. *International Journal of Lexicography*, 3(4):235–312.

[Morin and Jacquemin, 2003] Morin, E. and Jacquemin, C. (2003). Automatic acquisition and expansion of hypernym links. In *Computer and Humanities*.

[Navigli and Velardi, 2005] Navigli, R. and Velardi, P. (2005). Structural semantic interconnections: A knowledge-badsed approach to word sense disambiguation. *IEEE Transctions on pattern analysis and machine intelligence*, 27(7):1075–1086.

[Noy and Hafner, 1997] Noy, N. and Hafner, C., D. (1997). The state of the art in ontology design. *AI Magazine*, 18(3):53–74.

[O'Hara and Wiebe, 2003] O'Hara, T. and Wiebe, J. (2003). Classifying functional relations in factotum via wordnet hypernym associations. In *International conference on computational linguistics*.

[Paice and Jones, 1993] Paice, C. D. and Jones, P. A. (1993). The identification of important concepts in highly structured technical papers. In *16*sup *th ACM SIGIR Conference on Research and development in information retrieval*.

[Palmer, 2004] Palmer, M. (2004). http://www.senseval.org/senseval3. In *Third International workshop on the evaluation of systems for semantic analysis of text*.

[Pantel and Lin, 2000] Pantel, P. and Lin, D. (2000). An unsupervised approach to prepositional phrase attachment using contextually similar words. In 38_{th} *meeting association of computational linguisitcs*.

[Pantel and Ravichandran, 2004] Pantel, P. and Ravichandran, D. (2004). Automatically labeling semantic classes. In *Human language technologies and north american association of computational linguistics*.

[Protege, 2001] Protege (2001). http://protege.stanford.edu. Technical report.

[Quan et al., 2004] Quan, T. T., Hui, S. C., Fong, A. C. M., and Cao, T. H. (2004). Automatic generation of ontology for scholarly semantic web. In *International Semantic Web Conference*.

[Quinlan, 1993] Quinlan, R. J. (1993). *C4.5: Programs for Machine Learning*. Morgan Kaufmann.

[Ramshaw and Marcus, 1995] Ramshaw, L. and Marcus, M. (1995). Text chunking using transformation-based learning. In *Third Association for Computational Linguisitcs Workshop on Very Large Corpora*.

[Ratnaparkhi, 1998] Ratnaparkhi, A. (1998). Unsupervised statistical models for prepositional phrase attachments. In *Association of computational linguisitcs*.

[Ribas, 1994] Ribas, F. (1994). An experiment on learning appropriate selectional restrictions from a parsed corpus. In *International conference on computational linguistics*.

[Riloff, 1996] Riloff, E. (1996). Automatically generating extraction patterns from untagged text. In 13^{th} *National Conference on Artificial Intelligence*.

[Rosario and Hearst, 2001] Rosario, B. and Hearst, M. (2001). Classifying the semantic in noun compounds vis a domain-specific lexical hierarchy. In *EMNLP*.

[Ryu and Choi, 2004] Ryu, P. and Choi, K. S. (2004). Measuring the specificity of terms for automatic hierarchy construction. In *European Conference on Artificial Intelligence Workshop on Ontology Learning and Population*.

[Sabou et al., 2005] Sabou, M., Wroe, C., and Goble, C. (2005). Learning domain ontologies for web service descriptions: an experiment in bioinformatics. In *14th International World Wide Web Conference*.

[Salton and Buckley, 1988] Salton, G. and Buckley, C. (1988). Term-weighting approaches in automatic text retrieval. 24(5):513–523.

[Schoening, 2003] Schoening, J. (2003). www.suo.ieee.org. *AI Magazine*.

[Schutz and Buitelaar, 2005] Schutz, A. and Buitelaar, P. (2005). Relext: A tool for relation extraction from text in ontology extension. In *Fourth International Semantic Web Conference*.

[Shamsfad and Barforoush, 2003] Shamsfad, M. and Barforoush, A., B. (2003). Learning ontologies from natural language texts. *International Journal of Human-Computer Studies*, 60(1):17–63.

[Snow et al., 2004] Snow, R., Jurafsky, D., and Ng, Y. A. (2004). Learning syntactic patterns for automatic hypernym discovery. In *Advances in Neural information Processing Systems*.

[Stevenson, 2004] Stevenson, M. (2004). An unsupervised wordnet-based algorithm for relation extraction. In *International Conference on Language Resources*.

[Sure et al., 2002] Sure, Y., Erdmann, M., Angele, J., Staab, S., Studer, R., and Wenke, D. (2002). Ontoedit: Collaboratve ontology development for the semantic web. In *International Semantic Web Conference*.

[Tomokiyo and Hurst, 2003] Tomokiyo, T. and Hurst, M. (2003). A language model approach for keyphrase extraction. In *ACL Workshop on Multiword Expressions: Analysis, Acquisition and Treatment*, pages 33–40.

[Turney, 2000] Turney, P. (2000). Learning algorithms for key phrase extraction. *Information Retrieval*, 2(4):303–336.

[Turney, 2006] Turney, P. D. (2006). Expressing implicit semantic relations without supervision. In *21*sup *st international conference on computational linguistics*, pages 313–320.

[Vossen, 2001] Vossen, P. (2001). Extending, trimming and fusing wordnet for technical documents. In *NAACL Workshop on WordNet and Other Lexical Resources Applications, Extensions and Customizations*.

[Widdows and Dorow, 2002] Widdows, D. and Dorow, B. (2002). A graph model for unsupervised lexical acquisition. In *International conference on computational linguistics*.

[Yangarber et al., 2000] Yangarber, R., Grishman, R., P., T., and Huttunen, S. (2000). Unsupervised discovery of scenario-level patterns for information extraction. In *Applied Natural Language Processing Conference*.

[Yates and Neto, 1999] Yates, B. R. and Neto, R. B. (1999). *Modern Information Retrieval*. Addison Wesley.

[Zelenko et al., 2000] Zelenko, D., Aone, C., and Richardella, A. (2000). Kernel methods for relation extraction. *Journal of Machine Learning Research*, 3:1083–1106.

Appendix A

Electronic Voting Ontology

Some of the concepts, taxonomic relations, and non-taxonomic relations extracted from *Electronic Voting* domain texts using the methods presented in this dissertation are listed as follows:

Concepts *provisional ballot*, *voting machine*, *poll worker*, *manufacturer*, *county*, *fraud*, *electronic machine*, *voter*, *paper trail*, *electronic voting*, *minority group*, *ballot*, *certification law*, *national ID card*, *election monitor*, *laws*, *electronic voting technology*, *conspiracy theory*, *equipment*, *technicality*, *voter*, *voter fraud*, *voting machine manufacturer*, *minority vote suppression*, *eligible voter*, *election official*, *electronic voting machine*, *supervisory election judge*, *white poll watcher*, ...

Taxonomic Relations (*provisional ballot*, *ballot*), (*electronic machine*, *machine*), (*electronic voting*, *voting*), (*voting*, *action*), (*national ID card*, *ID card*), (*election official*, *official*), (*official*, *person*), (*voting machine*, *machine*), (*poll worker*, *worker*), (*election judge*, *judge*), (*county*, *location*), (*machine*, *artifact*), (*artifact*, *entity*), (*manufacturer*, *enterprise*), (*enterprise*, *organization*), (*worker*, *person*), (*person*, *organism*), (*organism*, *object*), (*voting*, *choice*), (*voting*, *action*), (*ballot*, *written document*), (*white poll watcher*, *poll watcher*), (*fund-raising letter*, *letter*), (*written document*, *social relation*), (*social relation*, *abstraction*), (*conspiracy theory*, *theory*), (*theory*, *cognitive process*), (*voter fraud*, *fraud*), (*fraud*, *crime*), (*crime*, *action*), (*minority vote suppression*, *vote suppression*), (*vote suppression*, *suppression*), (*eligible voter*, *voter*), ...

Non-Taxonomic Relations (*machine*→`produce`→*paper*), (*voter*→`cast`→*ballot*), (*voter*→`record`→*vote*), (*official*→`tell`→*voter*), (*voter*→`trust`→*machine*), (*worker*→`direct`→*voter*), (*county*→`adopt`→*machine*), (*company*→`provide`→*machine*), (*machine*→`record`→*ballot*), (*legal duty*→`protect`→*minority*), (*equipment*→`miscount`→*ballot*), (*worker*→`understand`→*duty*), (*official*→`buy`→*equipment*), (*voter*→`check`→*vote*), (*intimidation*→`performed on`→*voter*) (*state*→`buy`→*machine*),

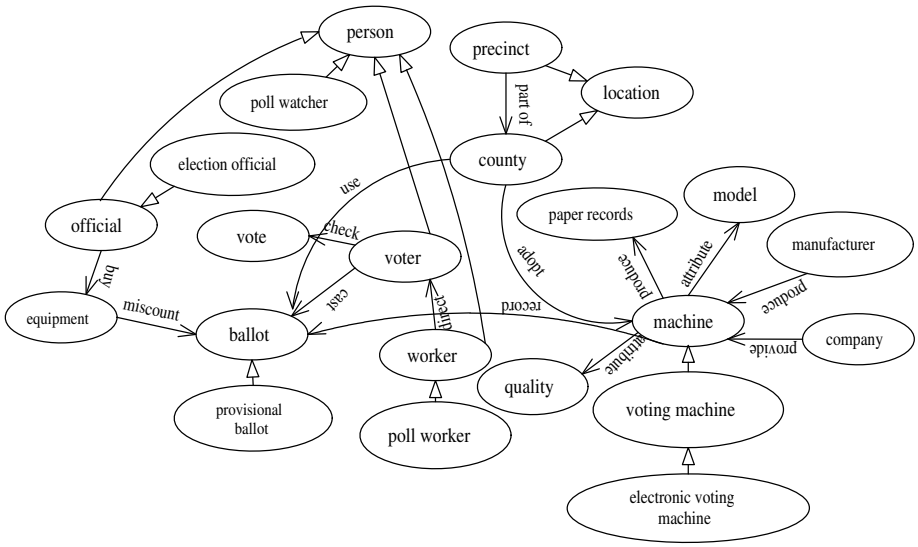

Figure A.1: Electronic Voting Domain Ontology

(*county*→`use`→*ballot*), (*state*→`use`→*machine*),
(*machine*→`attribute`→*quality*), (*campaign*→`member`→*co-chairwoman*),
(*testing*→`performed on`→*machine*), (*machine*→`attribute`→*model*),
(*precinct* →`part of`→*county*), (*tabulation*→`performed on`→*vote*),
(*party*→`posses`→*member*), (*state*→`posses`→*secretary*),
(*race*→`performed on`→*governor*),...

A snapshot of the *Electronic Voting* domain ontology is shown in the following Figure A.1.

Appendix B

WordNet

WordNet is an online lexical reference system whose design is inspired by the psycholinguistic theories of human lexical memory. WordNet contains 150,000 entries as nouns, verbs, adjectives and adverbs. In WordNet all nouns are categorized into a set of 25 unique beginners listed in Table B.1. Each of the lexical concepts is represented as a synonym set. Synonym sets are linked by semantic relations such as hypernym, meronym etc. Words in the WordNet are represented as synonyms sets. Each synonym set contains synonymously related words. The synonym sets are arranged in hierarchies along with several semantic relations. Each synonym set represents the unique sense for the given word. WordNet contains wealth of semantic information about English lexical categories. Since WordNet lists the semantic information for all possible senses, one has to know which sense of a given word is used in the context to utilize the semantic information. WordNet provides standard library to search the WordNet database at runtime. WordNet can be down loaded from http://wordnet.princeton.edu and complete list of bibliography involving WordNet can be accessed at http://lit.csci.unt.edu/ wordnet/.

Table B.1: WordNet Unique Beginners

1. {act, action, activity}	2. {animal, fauna}	3. {artifact}
4. {attribute, property}	5. {body, corpus}	6. {cognition, knowledge}
7. {communication}	8. {event, happening}	9. {feeling, emotion}
10. {food}	11. {group, collection}	12. {location, place}
13. {motive}	14. {natural object}	15. {natural phenomenon}
16. {person, human being}	17. {plant, flora}	18. {possession}
19. {process}	20. {quantity, amount}	21. {relation}
22. {shape}	23. {state, condition}	24. {substance}
25. {time}		

Appendix C

Corpus Description

Electronic Voting Domain Corpus Electronic Voting & Voting machines domain text is extracted from New York Times website www.nytimes.com in 2004. The text consists of fifteen documents and a total of more than 10,000 words. These documents describe using electronic voting machines for voting.

Tender Offers, Mergers, and Acquisitions Corpus(TNM) TNM Corpus is collected from TIPSTER Volume 1 corpus distributed by NIST. TIPSTER Volume 1 corpus consists of three year(1987, 1988, and 1989) news articles from Wall Street Journal. In TIPSTER corpus data each news article is labeled with the topic it describes. TNM Corpus is obtained by collecting news articles with the topic label *Tender offers, mergers, and acquisitions(TNM)*. TNM corpus consists of 270 articles and the size of the corpus is 29.9 MB.

SensEval-3 Corpus SensEval-3 corpus is collected from www.senseval.org. SensEval-3 test data consists of approximately 5000 words of running texts from two Wall Street Journal articles and an excerpt of the Brown Corpus. In SensEval-3 corpus, a total of 2212 words were manually annotated, with a reported agreement of 72.5%.

Vita

Janardhana Reddy Punuru was born in Madapalli, India, on July 1, 1976, the son of Nagi Reddy Punuru and Tulasamma Punuru. He attended his high school in Andhra Pradesh Residential School from 1989 to 1992. Punuru received a bachelor of technology degree in computer science from Sri Krishna Devaraya University of India in 1998. He completed a master's degree in computer science at Louisiana State University in 2002. In August 2002, he enrolled in the doctoral program in Computer Science Department at Louisiana State University and is working under the supervision of Dr.Jianhua Chen since then. He will receive the degree of Doctor of Philosophy at spring 2007 commencement.