u Ottawa

L'Université canadienne
Canada's university

FACULTÉ DES ÉTUDES SUPÉRIEURES ET POSTOCTORALES

uOttawa

L'Université canadienne
Canada's university

FACULTY OF GRADUATE AND POSDOCTORAL STUDIES

Ramanjot Singh Bhatia
AUTEUR DE LA THÈSE / AUTHOR OF THESIS

M.C.S. (Master of Computer Science)
GRADE / DEGREE

School of Information Technology and Engineering
FACULTÉ, ÉCOLE, DÉPARTEMENT / FACULTY, SCHOOL, DEPARTMENT

Automating Assignment of Semantic Relations in TANKA

TITRE DE LA THÈSE / TITLE OF THESIS

Stan Szpakowicz
DIRECTEUR (DIRECTRICE) DE LA THÈSE / THESIS SUPERVISOR

CO-DIRECTEUR (CO-DIRECTRICE) DE LA THÈSE / THESIS CO-SUPERVISOR

EXAMINATEURS (EXAMINATRICES) DE LA THÈSE / THESIS EXAMINERS

Diana Inkpen

Franz Oppacher

Gary W. Slater
Le Doyen de la Faculté des études supérieures et postdoctorales / Dean of the Faculty of Graduate and Postdoctoral Studies

Automating Assignment of Semantic Relations in TANKA

Ramanjot Singh Bhatia

Thesis Submitted to the Faculty of Graduate and Postdoctoral Studies
in Partial Fulfillment of the Requirements for the Degree of
Master of Computer Science

Ottawa-Carleton Institute for Computer Science
School of Information Technology and Engineering
University of Ottawa
Ottawa, Ontario, Canada

Your file Votre référence
ISBN: 978-0-494-18400-4
Our file Notre référence
ISBN: 978-0-494-18400-4

ABSTRACT

Theory of Semantic Relations is one of the oldest linguistic theories dating back to at least 5th century BC. Over the years a multitude of different theories and sets of relations have been developed in order to capture the semantics of a language. This has further gained importance with the desire to have artificial intelligence systems with the ability to comprehend human language. The focus of this thesis is such a system, TANKA. TANKA is a knowledge engineering aid which helps extract semantic relations from a semi-technical text as pairs of syntactic units and assigns them semantic relations. This thesis presents a feasibility study of automating the assignment of semantic relations in TANKA. The thesis looks at different methods and resources that will help to progress the system from a semi-automatic process to a fully automatic process.

For the purposes of analyzing the feasibility of this task, this thesis constructs a new dataset and looks into the use of two feature-rich parsers as sources of syntactic features. It also studies the utility of semantic resources such as WordNet, LDOCE and Roget's in order to find features for automating the assignment of semantic relations. A variety of combinations of these are tested to get an insight into the feasibility of this task.

To My Father, without whose advices this arduous journey would not have a destination

ACKNOWLEDGEMENTS

I would like to thank my supervisor Dr. Stan Szpakowicz, for his guidance and excellent feedback throughout my research. A special thanks goes to his approach of methodical thoroughness which has eventually become a natural way of thinking for me.

I would also like to thank

- Connexor for providing their parser Machinese Syntax for this research for free.
- Lianne Beth Rossman for lending a diligent set of eyes and ears over and over again for editing this thesis.
- My friends Quintin, Anna and Alexei for being there for me when I needed them.

I would like to thank the members of my committee:

- Dr. Franz Oppacher, Carelton University
- Dr. Diana Inkpen, University of Ottawa
- Dr. Stan Szpakowicz (Supervisor), University of Ottawa

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

Chapter 1: Introduction

1.1 Semantic Analysis

The word *Semantic* is derived from the Greek *σημαντικός (semantikos)* which means *"significant"* and from *'semainein'* which means *"to show, signify, indicate by a sign"* and was first applied by Michel Bréal [1883] to the psychology of language (fr. Sémantique). It refers to the study of the relationship between linguistic symbols and their meanings. The term *Semantic* itself had existed before and was used in seventeenth century by philosophers to denote the 'science of prediction of fate on the basis of weather signs'. Bréal's application of this term marked the beginning of the field of semantics as it known now. Traditionally semantics started out as the study of the relationship between linguistic expression and a topic of reference. The field grew to study the relationship between words and the thoughts they represented. Ogden and Richards [1923] were the first to introduce the difference between the symbol and the meaning to linguistic semantics, representing this as a semantic triangle relationship. They presented the indirect relationship between the symbols used to express the thought and the referent of the thought. They brought forward the point that symbolism in the form of language or any other symbols can be used to direct, organize, record and communicate the thought.

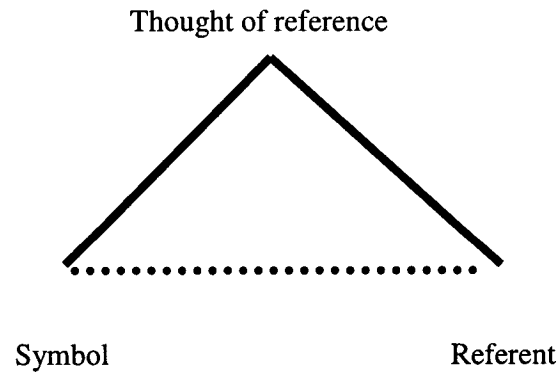


Figure 1.1 The Semantic Triangle

Logicians extended the semantic triangle to include logic terminology and various others have modified it to their domain and requirement (De Saussure's Triangle of signification, Dahlberg Concept triangle).

Ogden and Richards with their Semantic triangle re-asserted the observations recorded by Aristotle centuries ago:

> *"Spoken words are symbols of experiences in the psyche. Written words are symbols of the spoken words."*

Frege [1952[1]] also stressed this distinction between meaning and reference in the field of logic. He illustrated this with an example of *'Morning Star'* and *'Evening Star';* these two have different meanings but refer to the same object, the planet Venus. The Semantic Triangle was used by Ogden and Richards to postulate that the symbols on their own have no meaning and are only useful when put into context based on the speakers past and current experiences. They further argued that the confusion of symbol/word and associating it with things or objects in reality is the major cause of misuse and abuse of language. Conscious human effort is required to improve the utility of language as a medium of communication. This thought was in accordance with other semanticists of their time, which included Carnap [1937] and Wittgenstein [1953]. This study of semantics forms part of the philosophical viewpoint, whereas the linguistic viewpoint concerns itself with the study of conventional meaning and pragmatics. Computational linguists deal with the study of the semantics involving the process of automatic generation and automatic understanding of language. In these different streams of semantics the idea is the same: to understand the conceptualization of a thought and language but with varying goals.

The semantics this thesis deals with is the one as defined by Martin and Jurafsky [2000, p. 501]. They define semantic analysis as a process wherein formal structures are

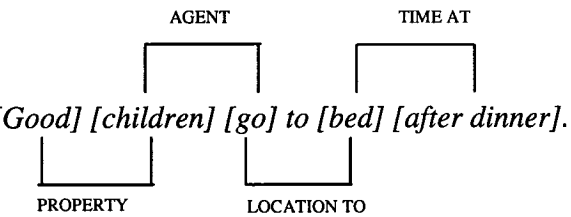
[1] Translation of Frege 1892

generated to capture the meaning of linguistic utterances. These formal structures form a higher level of understanding that help us map the world knowledge contained in the physical representation into a contained set which can be manipulated to perform desired operations.

The focus of this thesis is a system that extracts the world knowledge present in a text and assigns relations to the different pairs of concepts that occur in a sentence. The term "concept" is used in the similar sense as defined in Nastase [2003]:

> *"Occurrences, entities, attributes of occurrences and entities. Concepts are what underlies a linguistic expression, the idea in the mind of the utterer before he projects it into a verbal form, and the idea in the mind of the listener after he has processed this form"*

The following is an example of how the system codes this knowledge in terms of a set of relations.



In the above sentence *Good* signifies a PROPERTY possessed by *children*, and *children* is the AGENT of the verb *go* which signifies the LOCATION TO where the agent is headed AT A GIVEN TIME. In this example the tags PROPERTY, AGENT, LOCATION TO and TIME AT represent the semantic relations that help elucidate the underlying meaning in the sentence. The relations depend on how the underlying syntactic markers interact with each other and produce the resultant concept; changing the interplaying markers may or may not result in the change of the relation being expressed. In the given example, the

time concept marked by *dinner* can be replaced by a different marker without changing the relation expressed by it. For example

[Good] [children] [go] to [bed] by [9'O clock].

This thesis aims to automate the process of assigning meaning tags (semantic relations) to the parts of physical representation (represented here as pairs of syntactic units) which signify that meaning. The physical representation of a language is a set of rules which bounds and hence facilitates the expression of the thought by providing a medium for it. As long as the rules are consistent and understood by the parties in question the thought can be communicated. Consider the sentence *"All dogs are animals"*. This can be represented in various forms.

English: *All dogs are animals*
French: *tous les chiens sont des animaux*
First order predicate logic: $\forall x\ dog(x) => Animal(x).$

These representations mean the same thing as they convey the same thought. An important question here is: If there exists a symbolic representation, can meaning be extracted from it? After all, the representation is supposed to express the thought. It seems implied that if one has symbols to represent the thought that one would be able to extract thought from the symbols. However this can prove to be a Herculean task. As an example, English sees a heavy use of metaphor:

Lata Mangeshkar is the nightingale of India.

This does not mean that Lata Mangeshkar (a singer) is the bird nightingale. Also English demonstrates ambiguity:

Flying planes can be dangerous.

The chicken is ready to eat.

Small dogs and cats.

These and many other reasons make it hard to build systems that efficiently understand and process language automatically. Some of the large-scale efforts undertaken to produce systems that encompass everyday common sense include the likes of the CYC project [Lenat 1995], which is a knowledge base that has a domain that spans all everyday objects and actions. The huge amount of resources required to realize such a project has made this a one-and-only effort with the exception of the Open Mind Common Sense project [Singh *et al.* 2002] which relies on the knowledge acquired from the general public. The system analyzed here is not as grandiose as the aforementioned systems, but nonetheless offers a domain-independent process that extracts knowledge from text in the form of semantic relations with the difference that it requires minimum hard-coded knowledge.

1.2 Semantic Relations

The task of understanding meaning is partially solved by mapping the concepts from a source onto a bounded and understandable set of rules or symbols. These sets of rules or symbols then form the backbone of the artificially created language understanding system. The main requirement on any such set is that it should provide a complete coverage of the domain it deals with. There have been many such sets proposed for this cause, ranging from strict domain sets like the Air Traffic Information System (ATIS) [Miller *et al.* 1996], which has semantic frames like ORIGIN and DESTINATION which are filled by location names, to projects like PropBank [Kingsbury *et al.* 2002] which tag semantic roles in about 20 different numbered roles in a general semantic domain. In this thesis, the set of relations used is the one developed by Delisle [1994] and extended by Baker [1998], and Nastase [2003]. This relation set is derived from Fillmore's Case

theory [Fillmore 1968, 1977]. A brief overview of semantic relations and the set of relations used in this thesis can be found in the next chapter.

1.3 Introduction to TANKA

TANKA (Text Analysis for Knowledge Acquisition), [Copeck *et al.*, 1992] started out as a case-based system [Fillmore 1968, 1977] which assigns semantic relations to pairs of concepts extracted from a semi-technical text. The system was extended by Delisle [1994] and Barker [1998]. It was further refined by Nastase [2003] to perform semantic analysis at all three syntactic levels (clause, intra-clause and noun modifier level) as one combined process. The list of relations originally at 28 due to Delisle, was extended to 47 by Nastase.

The system performs the task of book-keeping and aids in assigning relations to pairs extracted from semi-technical text. The process is semi-automatic, with the user onus higher to begin with, but with subsequent rote learning this decreases and the system performs more or less automatically with the aid of a supervising eye.

The system attempts to create mappings between the syntax and the semantics in an English text. For this purpose the system has two distinct modules:

A) DIPETT (Domain Independent Parser for English Technical Text): The parser component which gives detailed parse trees based on comprehensive theoretical analysis presented in Quirk *et al.* [1985].
B) HAIKU : The Semantic Analysis component

The final output of the process is a partial model of knowledge contained in the text in form of semantic relations. The system was tested using a small text having 2000 pairs of annotated data [Barker and Delisle 1996]. The text was quite simplistic (a Junior Science

Book titled *Rain, Hail, Sleet and Snow* by Larrick [1961]), but nevertheless provided a comprehensive coverage over relations for testing and development purposes.

1.4 Goals of this Thesis

Text annotated with semantic information, in particular with semantic relations, could be an important help for language understanding systems. Such annotations have two major applications, information extraction and knowledge acquisition. They facilitate the discovery of patterns of information [Wallis and Nelson 2001, Hearst 1999, Moschitti et al.2003, Surdeanu et al.2003], helps in question answering systems [Narayanan and Harabagiu, 2004], helps in Machine Translation [Boas 2002], and helps for construction of lexicons [Barker 1998]. It has also been used for the purposes of word sense disambiguation and text summarization. It also has potential utility in enhancing the state of the art in Speech Recognition. The huge amount of knowledge available in textual format makes this task even more desirable.

The challenge undertaken in this thesis is to study the feasibility of automating the process of assigning semantic relations in texts. This thesis presents an investigation into multiple sources as possible directions that will help lead to a fully automated system in future.

The main goals of this thesis are summarized by the following points:

1) To create an annotated dataset for testing and development of features that would help in automatically classifying semantic relations.
2) Find a suitable parser that will provide dependency relations and syntactic features which help in automatically classifying the semantic relations.
3) Explore syntactic and semantic resources as potential sources of features.
4) Lastly and the most important is to prove the feasibility of automating assignment of semantic relations.

In order to achieve these goals, a thorough investigation of various sources of features has been conducted. This ranges from the use of pure syntactic features to the use of simple statistical features extracted from semantic resources like WordNet [Fellbaum 1998], Roget's Thesaurus [Kirkpatrick 1992] and Longmans Dictionary of Contemporary English [1997] (Figure 1.2). A hybrid approach by combining the syntactic features with the semantic features is also considered to achieve any improvement that might be possible.

Also in this process the dependency of such a task on the parses and the syntactic attributes has been studied. For this purpose two parsers have been used: Connexor's Machinese Syntax [Tapanainen and Järvinen. 1997] and Xerox Incremental Parser [Chanod, Ait-Mokhtar, Roux, 2004].

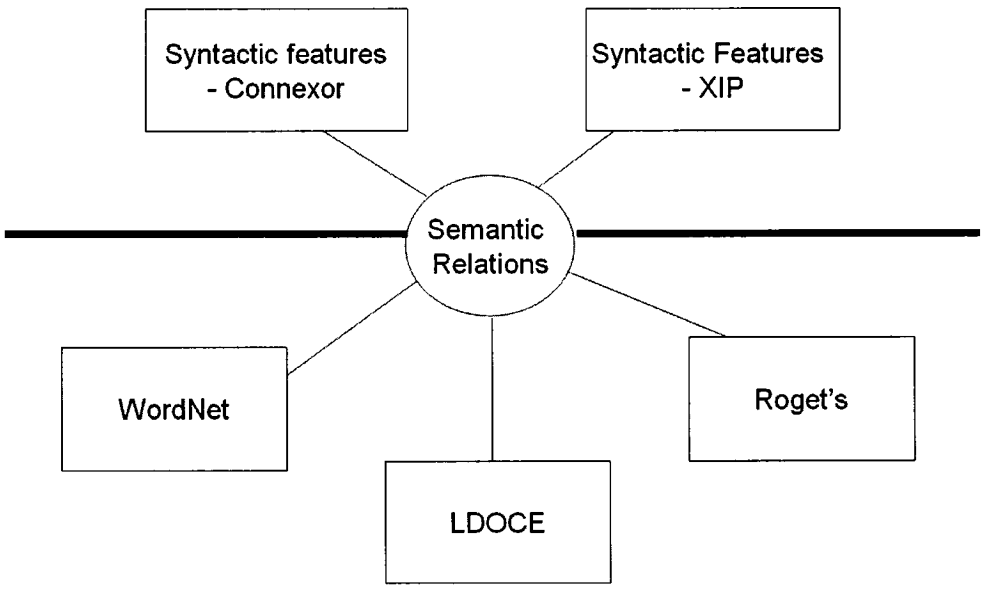


Figure 1.2 Sources of features

The choice of sources for the features represents two different categories: Syntactic parser-derived features versus semantic features generated from resources of world knowledge. As discussed in chapter 2, the syntactic features have been successfully used

in similar tasks whereas the semantic resources have mostly been ignored for this purpose.

1.5 Organization of the Thesis

The second chapter details the current status of the system TANKA, discussing the design, functionality and the set of semantic relations. It also describes some of the related work done in this area.

Chapter 3 reviews the tools and resources used in this thesis, discussing the merits of these resources and the reasons for their selection.

Chapter 4 describes the process of data preparation used in the learning task. This is followed by a discussion on the distribution of the relation classes in the dataset. It concludes with an evaluation of the data set, calculating a Kappa co-efficient of agreement between the parser outputs and expert annotators.

Chapter 5 examines the syntactic and semantic feature sets, discussing the experiments performed and the results achieved.

Chapter 6 presents the final results obtained in this thesis.

Chapter 7 offers a discussion of the implications of the findings and suggests future work. This chapter is followed by the references and the appendices.

Chapter 2: Literature Review and System Background

The aim of this work is to find features (both syntactic and semantic) which will allow the long standing project TANKA to move to the next step, that being towards automation. As it stands now, it is an autonomously developed system which processes semi-technical text to gain knowledge. The driving force behind the project was to have a system that would allow for easy knowledge acquisition and which would serve as an assistant to a knowledge engineer for the purposes of building a knowledge base of concepts from a domain. The natural path of progression dictates that the project moves away from its semi-automatic dependence. This thesis serves as the first step in this task. A new and bigger dataset has been constructed and a variety of experiments have been run to test the task's feasibility and to find features which will allow automation. This task is dubbed as semantic role labeling in the natural language processing field but is slightly different in its approach and working.

This chapter begins with an introduction to the specific area of natural language processing dealt with in this thesis and then focuses on some of the work related to this field. This is then followed up with a description of the current status of the system.

2.1 Introduction to the Theory of Semantic Relations

The driving principle for having semantic relations is the need to understand the meaning represented in text. The presence of semantic roles in linguistic theory dates back to as early as 5th century BC. Panini's Karaka theory [Misra 1966] identifies verbal relationships in the Sanskrit language. The Karakas form the basics of semantic notions that pivot sentence constructions. These are similar to Chomsky's case roles/relations and to Fillmore's case grammars. The verbal relations are divided into six cases: *Apadana* ('take off'), *Sampradana* ('bestowal'), *Karana* ("instrument"), *Adhikarana* ('location'), *Karman* ('deed'/'object'), and *Karta* ('agent'). On top of these six Karakas, Bhartrhari [1971] discusses a possible set of cases under the name of *Sesa* (the rest or the extras).

Sesa does not represent a karaka relation but may involve or be preceded by one of the karaka relations. The possessive and the vocative cases fall under *Sesa*.

In the more recent times, Tesnière [1959] offers a verb-centered grammatical model which marks dependency between the elements of the sentence that enter into a certain governor-subordinate relationship. Tesnière studies the nature of the relationship between the verbal governor and its immediate subordinates. This is divided by Tesnière into two distinct categories the 'actants' and the 'circumstantials'. The actants are the elements which are central participants in the process. They are divided into three types: prime actant, second actant and the third actant. The prime actant is the one that performs the action (the subject), the second undergoes the action (the direct object) and the third is the one to whose benefit or detriment the action takes place (indirect object). The circumstantials express the associated adverbial (temporal, locational, etc.) relation.

Gruber's [1965, 1967] work essentially focused on the verbs of motion and the extension of motional and locational prepositions to other verbs. Gruber gave a smaller set of relations that comprised of agent, theme (an object that is in motion or that is located), location, source, and goal.

Fillmore [1968, 1977] proposed a set of thematic relations different from that of Gruber's. Fillmore offered a deeper semantic treatment of the syntactic cases and gave initially a list of six cases (Agentive, Instrumental, Dative, Factitive, Locative and Objective). This was later extended to include the Benefactive case. These by no means were termed as complete therefore did not preclude the possibility of more cases. The verbs in a sentence are classified according to which "case frames" they may be inserted into; essentially, this means that each verb specifies which roles it assigns its arguments. This is also called as the "One Case per Argument Rule".

Jackendoff [1972, 1976] builds on Gruber's work and bases his framework of conceptual structures on it. He proposes a similar list of relations: theme, source, goal, agent and experiencer. He also discusses additional semantic fields like musical pitch, a

circumstantial field (with events and states as locations), and an existential field (with a degenerate space containing just one location – existence).

The list of thematic roles is a much debated issue with the number varying from a few to hundreds; but on the whole there is a stable core which is consistent across all representations. This core includes relations expressed in terms of themes, locations, paths, agents, and causation. The modern theory of semantic relations stems from Fillmore's and Jackendoff's work. An excellent treatment of case theory can be found in Somers [1987]. The next section looks at how these cases/relations have been used in different systems and also discusses some of the systems which deal with automatic assignment of these relations in a sentence.

2.2 Automatic Role Labeling

The work in automatic role labeling can be divided into three main categories:

- Early domain-dependent systems
- Systems based on FrameNet cases and data
- Systems based on PropBank roles and data

Some of the early work in semantic role labeling features systems which were tailor-made for particular domains. A prominent example of such a system is Air Traveler Information System (ATIS) [Miller *et al.* 1996]. Miller gives a statistical system that parses sentences from the Air Traveler Information data and creates frames, assigning information like destination location, departure location and time of the flight. These labels are then used to understand any query and to output appropriate responses.

Riloff [1993] proposes a system that deals with terrorist attacks and automatically picks up patterns to fill in slots with information of the trigger of the attack, target of the attack, etc. Riloff and Schmelzenbach[1998] extend this approach to derive the entire case-frames from an automatically generated lexicon. The lexicon used for this purpose has

nine categories associated with terrorism: *Building, Civilian, Gov-Official, Military People, Location, Terrorist, Dates, Vehicle and Weapon.*

The generic domain-independent systems which deal with unrestricted text like the system which is the focus of this study (TANKA) have used two main annotated English corpora: the PropBank [Kingsbury *et al.* 2002] and the FrameNet [Baker *et al.* 1998].

a) Proposition bank (PropBank)

The PropBank [Kingsbury *et al.* 2002] project assigns predicate-argument information (semantic roles) to the verbs in the Penn Treebank data. PropBank takes a conservative approach as far as the set of semantic roles it uses and defines them on a verb by verb basis thereby sidestepping any questions about completeness of the set. The semantic roles in PropBank are numbered arguments Arg0-Arg5, AA and 13 AM- adjuncts types. These map on to the usual semantic roles. For example the Arg0 exhibits the Agent relation and Arg1 the relation of patient or theme, but there is no consistent mapping for the higher-numbered arguments.

A verb can have more than one semantic role assigned to it and this set of roles is called a *roleset.* On an average rolesets contain two to four roles. They are associated with a set of syntactic frames indicating allowable syntactic variations in the expression of that set of roles. The role set with its frame is called Frameset. A polysemous verb may have more than one Frameset. The collection of Frameset entries for a verb is called as the Frame File of that verb. PropBank has 4,500 frames defined for over 3,300 verbs.

The Febuary 2004 release of PropBank comprises of 85,000 sentences with about 250,000 arguments in the development set and 5000 sentences with 12,000 arguments in the test set.

The sheer size of the dataset makes it a valuable resource for the task of learning the assignment of semantic relations. However, there are considerable differences which

make the use of PropBank data inadequate for the purpose of this thesis. In comparison to PropBank, the set of semantic relations used here is not restricted to only verbs and in addition incorporates noun-modifier relations. Further, due to the variability of the assignment of numbered arguments as semantic roles, no fixed mapping can be automatically synthesized. In order for the PropBank data to be useful here, one would need to manually re-label it and augment the set with noun-modifier relations. This extra work takes away any advantage PropBank could provide.

b) FrameNet

FrameNet [Baker *et al.* 1998] is a large-scale domain-independent project that marks the semantic roles in terms of Frames. A Frame is a schematic representation of situations involving various participants, props and other conceptual roles. Examples of Frames include JUDGEMENT, CONVERSATION, TRANSPORTATION, etc. Frames consist of multiple lexical units – an item corresponding to a sense of a word. Also each frame is associated with a set of semantic roles. FrameNet project currently has over 625 frames and has annotated 135,000 sentences drawn from the British National Corpus [2001]. FrameNet frames, as opposed PropBank roles, also contain the nouns and adjectives in addition to verbs and their arguments.

The semantic roles assigned by FrameNet are substantially different than the semantic relations being classified here, this difference makes the use of FrameNet data inappropriate, as it would be required to re-annotate it to make is useful for the task here.

2.2.1 Study of Existing Automatic Semantic Role Labeling Systems

Automatic assignment of semantic roles is an important part of language understanding systems and has been implemented in many different ways by different people.

Gildea and Jurafsky [2002]

The pioneering work in this area was done by Gildea and Jurafsky [2002], they used probabilistic methods to automatically identify the semantic roles filled by constituents in a sentence. They use hand-annotated data of semantic roles marked as Frame elements consisting of 50,000 sentences from British National Corpus. The data used in their work was the product of the FrameNet semantic labeling project. They use statistical techniques to learn the semantic roles. Their system has 67 frame types from 12 general semantic domains as introduced by Fillmore [1976]. The domain types include body, cognition, communication, emotion, health, motion, perception, society, space, time and transaction. The features are:

1) Phrase type: The phrase type (NP, PP, S etc) of the phrase corresponding to the semantic argument.
2) Governing category: Marks if a word is S or VP in a noun phrase, it corresponds to: if the word is a subject or a direct object in a sentence.
3) Parse tree path: The syntactic path through the parse tree from the parse constituents to the target word being classified.
4) Voice: Marks if the predicate has active or passive voice.
5) Position: A binary feature for marking of the phrase if it is before or after the predicate.
6) Head word: the syntactic head of the phrase.

These features are used to calculate the prior probabilities of the different semantic roles. They calculate probability estimates of how likely a constituent will fill a particular role for a target predicate by calculating probabilities for various combinations of the features and then combining these to form the classifier. The features presented in Gildea and Jurafsky have been used as baseline features in most of the work that has followed it.

Surdeanu et al [2003]

Surdeanu *et al.* use additional features to the ones in Gildea and Jurafsky [2002] and report performance gains. They use PropBank dataset and decision trees C5.0 classifier. The additional features used in their system are:

1. Content word: the most informative word extracted from the phrase depending on heuristics based on phrase type: For example, in a preposition phrase the right-most child is selected as the content word.
2. Part of speech of the content word
3. Named entity class of the content word, provided it belongs to one.
4. Named entity Boolean flags which are set for particular named entity types, For example, organization, person, location, money etc.
5. Phrasal verb collocation based features which record the frequency with which a verb is immediately followed by any preposition or particle and the frequency with which a verb is followed by its predominant preposition or particle.

The authors show performance increase of over 5% by using these additional features.

Thompson *et al.* [2003]

Thompson *et al.* propose a generative model which given the predicate and the sentence constituent outputs the semantic role label. It gives a joint probability distribution using Hidden Markov Models over predicates, frames, roles and constituents. They use the FrameNet data for their experiments. The system also handles any null roles present in the sentence which was not done in Gildea and Jurafsky [2002]. Null role is the default role assigned to any constituent that does not represent any semantic role as described in their system. Thompson's systems parameters are different yet comparable to the ones in Gildea and Jurafsky [2002], with the exception of two features that are missing in Thompson *et al.*: *voice* and *position*.

At a finer level of detail, there is work that emphasizes the importance and necessity of parsing for the task of automatically assigning semantic roles. Xue and Palmer [2004] state that though a variety of the systems have used syntactic parse trees, they do not fully use the information provided in the parses and they state that there is room for improvement resulting from an efficient and complete use of the parses. At the same time Gildea and Palmer [2002] argue the necessity of having to parse for achieving similar accuracy in the task. They use chunked output and compare it to using a complete parse tree output and report only a small drop in accuracy. Chen and Rambow [2003] show that by using deep syntactic features one can perform comparably to a system using surface-syntactic features, which have been the focus of majority of the systems that currently exist.

2.2.2 Conference on Computation Natural Language Learning - Shared Task 2004 and 2005

More recently the Semantic Role Labeling (SRL) task has gained a lot of importance; this is evident from the two consecutive shared task calls by the Conference on Computational Natural Language Learning (CONLL) in 2004 and 2005.

The task addressed in the conference consisted of analyzing the proposition expressed by some target verbs of a sentence. In particular, all the semantic roles filled by constituents for each target verb in the sentence were required to be recognized automatically. This task can be divided into a two-phase procedure involving the recognition and the labeling of the arguments. Both of the shared tasks used the PropBank dataset. CONLL-2004 evaluated systems based on partial syntactic information; whereas CONLL-2005 allowed the use of full syntactic parses.

A) CONLL - 2004

The CONLL-2004 task [Carreras and Marquez 2004] involved coming up with machine learning strategies which address the SRL problem on the basis of only partial syntactic information thus avoiding the use of full parsers and external semantic knowledge bases. The information provided for each sentence included:

1) Words
2) Part of speech
3) Chunks in *IOB2 format* [Ratnaparkhi 1998]
4) Clauses in *start-end format* [Carreras and Marquez 2004]
5) Named entities in *IOB2 format*
6) Target verbs marking n (can be zero) predicative verbs. This column specifies the governing verbs (in their base form) of the propositions being analyzed.
7) For each of the n target verbs a column in the *start-end* format specifying the arguments of the proposition.

Ten systems participated in the CONLL-2004 task. Seven of the systems had the F-measure at the task of automatically recognizing semantic roles for a target verb in the range of 60 - 70 and three had F-measure less than 60. The best result of 70 was obtained by a SVM-based system [Hacioglu *et al.* 2004].

B) CONLL 2005

CONLL-2005 shared task [Carreras and Marquez 2005] allowed entries in two categories: the open challenge and the closed challenge. The closed challenge only allowed the usage of the information contained in the training data, whereas the open challenge allowed the use of external sources of information and tools for the task of automating SRL. There was no entry submitted for the open challenge. This thesis work, in concordance with the open challenge, looks beyond just the syntactic features. In

comparison to the 10 systems in CONLL 2004, 19 systems participated in the closed challenge in 2005.

The standard training, development and test sets of PropBank dataset were used for the task. The annotation data provided for learning included the POS tags, chunks, clauses, full syntactic trees and named entities. All of the systems participating in this task made strong use of the syntax to extract features. The main types of features in this task can be divided into four general categories:

1) Features characterizing the structure of a candidate argument.
2) Features describing properties of the target verb predicate.
3) Features that capture the relation between the verb predicate and the constituent under consideration.
4) Global features describing the complete argument labeling of the predicate.

Both of the CONLL tasks use a baseline developed by Erik Tjong Kim Sang [Sang and Buchholz, 2000)] from the University of Amsterdam. The baseline processor finds the semantic roles based on the following seven rules:

- Tag target verb and successive particles as V.
- Tag *not* and *n't* in the target verb chunk as AM-Neg.
- Tag modal verbs in the target verb chunk as AM-MOD.
- Tag first NP before target verb as A0.
- Tag first NP after the target verb as A1.
- Tag *that, which* and *who* before target verb as R-A0
- Switch A0 and A1, and R-A0 and R-A1 if the target verb is a part of the passive VP chunk.

The average baseline reported in the task was 36.0. The final performance in the 2005 task had seven systems with F-measure in the range of 75-78, seven in the range of 70-75 and five having scores between 65 and 70. The best performance was obtained by

Punyakanok *et al.* [2005] and had a score of 80. This score is equal to the highest published score on the same dataset and task [Pradhan *et al.* 2005]. The improvement in 2005 has been attributed to a larger training set and detailed parse trees.

2.2.3. The Noun-Modifier[2] Data and Automatic Recognition of Noun-Modifier Relations

The noun-modifier data is a collection of 600 pairs of base noun phrases (modifier-noun pairs) annotated with the semantic relations. The set of relations is the subset of the list discussed in section 2.4. Some of the relations do not occur in a noun-modifier form and do not find themselves in this data. This dataset has annotations for 30 of the 47 relations. A discussion of relations and the levels at which they can exist can be seen in Nastase [2003].

The noun-modifier data has been collected from the following sources:

1) Manually from Judith Levi's "The syntax and semantics of complex nominals" (1978).
2) Automatically from Nancy Larrick's "The junior science book of rain, hail, sleet and snow" (1961).
3) Semi-Automatically from SemCor - the version annotated with WordNet 1.6 senses.

Some of the examples were constructed and added for relations infrequent in these texts. Nastase [2003b] looked at classifying semantic relations which occur only as noun-modifiers using this dataset. She used feature vectors derived from WordNet and Roget's Thesaurus and showed that such learning is possible. Turney [2003, 2004] also looked at automatically classifying the same set of relations under the noun modifier category using the same dataset. Turney used a statistical approach and learned the assignment of noun modifier relations by finding a relational similarity measure for the words in the pair.

[2] http://www.site.uottawa.ca/~vnastase/

Turney reported an F-measure of 36.6% for the 30 relations and an F-measure of 54.6% for a condensed set of five super-classes. Turney uses five of the six super-classes and leaves out the CONJUNCTIVE super class. The set of relations is divided into six super-classes as discussed in section 2.4. Even though Turney and Nastase use the same set of semantic relations, it still does not offer any comparison to work done in this thesis as these results are limited to only noun-modifier relations.

The next section presents the system TANKA and the set of semantic relations used in this thesis.

2.3 The System - TANKA

TANKA has two main components: a syntactic analyzer (DIPETT) and a semantic analyzer (HAIKU). The syntactic analyzer parses the data and produces parse trees which are then fed into the semantic analyzer - HAIKU. HAIKU processes these trees and starts an interactive process with the user in order to assign relations to the pairs extracted from the parse trees. HAIKU is a memory-based learner which, with sufficient learning, can accurately guess the relations. Thus an incremental knowledge acquisition is conducted with an external user support.

The system was originally designed to handle relations at three levels independently. The three levels at which the relations were considered in this system were: clause, intra-clause and noun phrase. This changed with Nastase's [2003] work where she remodeled the system to process all the relations together. The rationale for this decision was the fact that the same concepts marked by any of the relations could exist at any of the three different levels and it was felt to be unnatural to decompose them into three different lists. This assimilation of the three levels also resulted in a gain in the responsiveness of the system's learning process. The number of good suggestions made by the semi-automatic process was higher than the number of erroneous suggestions made after only

seeing 10 % of the data. This number was as high as 40% when using different lists for the three levels.

The working of HAIKU involves extracting all of the syntactic units from the parse tree. The syntactic units HAIKU searches for, based on the information extracted from DIPETT, are Nouns, adverbial modifiers, adjectives, head of the verb phrase, type of clauses, sentences and anything that can be conceived as an entity. The processing is ordered in such a fashion that embedded structures are processed first.

For each syntactic unit the system extracts the following information:

i) Head word: It is an open-class word in its uninflected form.
ii) Part of speech: For a phrase the POS of the head is used as its POS.
iii) Syntactic role: DIPETT assigns syntactic roles to the units. Possible role values are subject, attributes, adverbial, and modifiers for the noun phrase, sentence qualifier, types of clause, sub-ordinate and relative clauses.
iv) Indicator for the position of a syntactic unit in the sentence, e.g., the preposition for a noun in prepositional phrase.
v) Structure-dependent information: For verbs, information such as tense, number, modals etc and for nouns number, determiners, etc.

Any of these elements if not present is assigned the value "nil" in the structure.

HAIKU assigns each of the syntactic units extracted a unique identifier and builds from it a database on the word forms. This process also involves the simplification of the parse tree into a structure which consists of identifiers instead of syntactic units. HAIKU then extracts the pair information from the tree. These pairs are analyzed using the simplified tree structure and the database. HAIKU uses a set of heuristics to come up with a possible relation for individual pairs. The heuristic process uses any syntactic indicator present, makes use of the syntactic roles of the units involved and of any previously assigned relation that matches the pair. The heuristics operation is a four-step process:

i) Using Relation Match: If the system finds that the pair under scrutiny has already been assigned a relation and is available in the database, it presents the user with the same relation and exits the heuristic process.

ii) Pattern Match [Delisle 1993]: The system performs a pattern match for both noun and verbs as described in Nastase [2003] and tries to find a suitable relation in the existing database.

iii) Marker Dictionary: This step uses a dictionary of syntactic markers to guess the relation. The syntactic marker could be a preposition, subordinator or a coordinator. Each syntactic marker present in the dictionary, signals the presence of one or more possible relations, e.g., the presence of the word *"also"* indicates two relations namely CONJUCTION and ACCOMPANIMENT.

iv) No Result: If no relation is discovered by the above-mentioned steps the system returns an empty list and presents the user with the option of selecting a relation from the complete set.

There are other tasks performed by the system as discussed below:

Interaction with users

The user is presented with a possible pair and asked to accept the pair, reject the pair or switch the order of the head and modifier in the pair. After the pair has been selected, the heuristic process of the system either creates a select list of relations or a full list of relations depending on the success of the process. This list is then presented to the user. The user is asked about the validity of the selected list and in case of multiple options asked to choose one. At this stage the user can accept the suggestion made or reject it and provide a different alternative.

The system notes the time in seconds from the start to the end of this interactive session. It also asks the user to note the level of difficulty of the relation assignment.

Collecting statistics

The system records various parameters during the course of the experiment. These include change in the order of the pairs made by the user and onus as provided by the user. The user must choose a value corresponding to the onus required in the assignment of the relation to the pair. The options for the values are 0-*obvious relation*, 1- *some reflection required*, 2- *serious thought required* and 3- *no appropriate relation found.* In addition to this, the time and number of choices (the number of relations proposed by the system), user action, and heuristic type are also recorded. These statistics are used to calculate the performance of the system.

The task approached in this dissertation is the automation of the assignment of semantic relations assigned in TANKA. These relations are assigned to a pair of units extracted from the sentence. The pairs extracted from the parser are under the heading of HEAD (H) and MODIFIER (M). The task is generally considered hard for humans as well, and hence the automation of these is considerably complex. The next section details the relation list used in this process.

2.4 The Relation Set

The set of relations consist of 47 relations divided into 6 major categories (Six Super-classes). These relations are assigned to concepts occurring within a sentence. The next section briefly describes this set of relations. Most of examples presented here are taken from Nastase [2003].

A. Causality

The causality relations encompass relations which describe the interaction of occurrences. The occurrences can be expressed by clauses, noun phrases or modifiers. There are seven relations under Causality and the differences between these are defined by how the occurrences interact.

Relation	Abbreviation	Description/example	Paraphrase
CAUSALITY			
Cause	*caus*	Cause relation signifies occurrence of an event. Here the head signifies the cause event and the modifier expresses the result of that event. e.g. *[snow] [blindness]*	H makes M occur or exist. H is the necessary and sufficient condition for M to exist.
Effect	*effe*	Effect is the inverse of Cause. The head is the effect of modifier. e.g. *[print] [command]*	M makes H occur or exist
Purpose	*purp*	Purpose relation expresses the relation between a pair where the modifier defines the reason for the head. e.g. *[pain-relief] [medication]*	H is for V-ing M
Entailment	*Entl*	Entailment describes a relation between two events, where the event signified by the head is the one that propels the event signified by the modifier. e.g. *[Hard work] ensures [passing of the exam].*	H makes M occur or to exist
Enablement	*Enbl*	Enablement is similar to entailment, but in this case the head event signifies a conditional modifier event. e.g. *[car runs] if there is [gas in the tank].*	H enables M
Prevention	*prev*	Prevention relation is one in which the head prevents from the modifier event from occurring. E.g. *[The party did not arrive] because [the flight was cancelled].*	H opposes M and is sufficient to prevent M.
Detraction	*detr*	Detraction is to prevention what enablement is to entailment. Detraction the head opposes but not necessarily stops the event from occurring. e.g. *The [Indians won] despite [the fierce bowling attack by Pakistan].*	H opposes M

Table 2.1: Causality Relations

B. Temporality

The relations emphasizing time dependency (occurrence of time as a point of time or an interval) find themselves in this class. Only the relations occurring within the sentence are dealt with in here. Temporal occurrences are detected based on the markers that convey temporality (since, before, after, etc.).

Relation	Abbrv.	Description/example	Paraphrase
TEMPORALITY			
Co-occurrence	*Cooc*	The occurrences marked by H and M occur at the same time. e.g. *[writing novels] while [listening to music]*	H and M occur or exist at the same time.
Frequency	*Freq*	H occurs as marked by M. e.g. *[weekly] [game]*	H occurs every time M occurs
Precedence	*Prec*	One occurrence precedes the other marked by M. e.g. *[Watering the plants] before [leaving for holidays]*	H occurs before M occurs
Time at	*Tat*	The occurrence of H is at the time marked by M. e.g. *[winter] [travel]*	H occurs when M begins to occur
Time from	*Tfrm*	H started to occur when M occurred. e.g *[playing well] since [January]*	H began to occur when M began.
Time through	*Tthr*	H is an occurrence which is delimited by M, M can be a occurrence or time interval. e.g. *[lunch hour] [practice]*	H existed while M existed
Time to	*Tto*	H exists until M exists, H is an occurrence and M can be an occurrence or a point in time. e.g. *[party] until [dawn]*	H exists till M existed

Table 2.2: Temporality Relations

C. Spatiality

This set of relations describes the link between an occurrence or an entity in space. The location in space can be expressed by an explicit spatial location (*near the river*) or by an occurrence marked by a spatial preposition (*where the two rivers meet*).

Relation	Abbrv.	Description/example	Paraphrase
SPATIALITY			
Direction	*Dir*	H is in the direction as marked by M e.g. *[inward] [look]*	H is directed towards M
Location from	*Lfrm*	H starts from the location indicated by M e.g. *[foreign] [capital]*	H originates at M
Location to	*Lto*	H ends at the location indicated by M. e.g. *[Homeward] [journey]*	The destination of H is M
Location through	*Lthr*	This relation specifies that H is located anywhere in the region as indicated by M. e.g. *[travel] [all over Europe]*	H occurred through M
Location At	*Lat*	H is located at M. *e.g. [desert] [storm]*	H is located at M
Located	*Ltd*	The location of H is marked by M. *e.g.[home] [town]*	H is the location of M
Oreintation	*Ornt*	H is oriented like M. *e.g. [Erect] [tree]*	H is oriented like M

Table 2.3: Spatiality Relations

D. Conjunctive

The relations in this class describe the conjunction or disjunction of occurrences, entities or attributes.

Relation	Abbrv.	Description/example	Paraphrase
CONJUNCTIVE			
Conjunction	*Conj*	Conjunction e.g. *[Running] and [swimming]*	Both H and M exist
Disjunction	*Disj*	Disjunction e.g *[painting] or [drawing]*	Either one or both H and M exist

Table 2.4: Conjunctive Relations

E. Participant

The participant set consists of relations which link the occurrence and the participants. The occurrence is expressed through a verb, deverbal noun or a deverbal adjective. The participant can be an entity or an occurrence.

Relation	Abbreviation	Description/example	Paraphrase
PARTICIPANT			
Agent	*Agt*	M is the subject of H *e.g.[student][protest]*	M performs H
Beneficiary	*Benf*	M benefits from H *e.g. [student] [discount]*	M benefits from H
Instrument	*Inst*	*e.g. [laser] [printer]*	H uses M
Oject	*obj*	*e.g.[metal] [separator]*	M is acted upon by H
Part	*part*	*e.g.[printer][tray]*	H is part of M
Possessor	*posr*	*e.g.[national][debt]*	M has H
Possession	*poss*	*e.g. [Bearded] [man]*	H has M
Product	*Prod*	*e.g. [plum] [tree]*	H produces M
Exclusion	*Excl*	H replaces M *e.g. Walked instead of running*	M is excluded from H
Stative	*St*	H is the state as defined by M *e.g.[sleeping] [dog]*	H is in a state of M
Accompaniment	*Acmp*	M and H occur together *e.g [We eat supper] with [my family]*	M is accompanied by H
Object Property	*Obj_prop*	*e.g. [Sunken] [Ship]*	H underwent M
Source	*Src*	H is derived from M *e.g. [Olive] [Oil]*	M is source of H
Whole	*Wh*	M is a part of H *e.g. [Daisy] [Chain]*	M is part of H
Recipient	*Recp*	H receives from M, but may not be the beneficiary *e.g. [Rohan wrote] Sham a reference letter to [prospective employees]*	H is the recipient of M
Property	*Prop*	H defines one of the attributes of M *e.g. [Blue] [book]*	H is M

Table 2.5: Participant Relations

F. Quality

The quality set contains relations with different attributes as opposed to the other classes whose composition was more coherent. These relations link an occurrence to one of its arguments.

Relation	Abbreviation	Description	Paraphrase
QUALITY			
Container	*Cntr*	*E.g. The [box] of [chocolates]*	M contains H
Content	*Cont*	The modifier is contained in Head. This is the inverse relation of the Container relation. *E.g. milk bottle*	M is contained in H
Equative	*Eq*	*e.g. [player] [coach]*	H is also M
Order	*Ord*	*e.g. [Flowers] [in front of window]*	H is before M in physical space
Material	*Matr*	*e.g. [Steel] [Helmet]*	H is made of M
Measure	*Meas*	*e.g. [Big] [Barn]*	M is a measure of H
Manner	*Manr*	*e.g. [walk] [quietly]*	H occurs in a way indicated by M
Type	*Type*	*e.g.[Oak] [tree]*	M is a type of H
Topic	*Top*	*e.g. [weather] [report]*	H is concerned with M

Table 2.6: Quality Relations

Chapter 3: Resources and Tools

This chapter looks at the various tools and resources that are used throughout the rest of this thesis. Each tool and resource named here is supported with an introduction to the tool/resource, the theory behind it and the rationale for its choice.

3.1 Syntactic Resources

The creation of a new dataset requires identifying the constituents of a sentence as potential pairs for the assignment of semantic relations. The parser plays an important role in the assignment of semantic relations as it identifies the pairs in addition to providing the syntactic information about the constituents. A new dataset for the purposes of analyzing features that will help automate the process of assigning semantic relations has been constructed in this thesis. The construction of this dataset required parsing the text from which the relations are extracted. The next few sections discuss the parsers considered for this task. The first and foremost choice was to use the system's internal parser, DIPETT, as it is fully integrated with the system and its output is automatically understood and utilized by the semantic analysis module HAIKU.

3.1.1. DIPETT

DIPETT is the parser which was designed as a part of the TANKA system. The DIPETT parser provides excellent syntactic detail of the constituents in the pair. These pairs are then fed into the semantic analysis module. The semantic analysis module, HAIKU, aids in the process of assigning the relations in a semi-automatic fashion. The parser's output is fully integrated with the semantic analyzer. In spite of all its advantages DIPETT is not without shortcomings. The parser tries to parse the sentences within a predefined period of time and if it is not able to come up with a parse, it outputs it as fragments.

Take for example the situation of parsing the sentence: '*Some of the members of this reptilian family began to live in the tops of the trees, which were then often more than a hundred feet high*'. The complexity of the sentence forces the parser to output the parses as four independent fragments as shown in Figure 3.1.

The direct consequence of getting such fragments is that any potential pairs that exist across the fragments are not selected and hence lost. This issue makes the use of DIPETT problematic and thus required further search for a suitable replacement.

The modular design of the TANKA allows for a different parser to be plugged in and only requires writing the interfacing scripts. Ideally, a variety of parsers would be tested to determine the effect of each parser's performance and quality in terms of finding semantic pairs. However, this process requires annotating data for each parser separately and then conducting a multitude of experiments to determine the best option. Considering the desire to test more than one parser and also considering the time involved in doing so, the choice of testing with two parsers was determined to be the most feasible option. The main requirement for a replacement parser is that it should output dependency functions that will allow extraction of the potential pairs. The other parsers that were considered are discussed in the next few sections.

```
parse_tree__frag_series
 sentence_fragment
  entity
   ref
    pron some_of
   number sg
   tokens some_of
  sentence_fragment
  entity
   determinatives
    deter the
   head_noun
                .................
                        .................
                noun
                 n family countnoun
                 noun_modifiers
                  pre_modif
                   n reptilian countnoun
               number sg3
      tokens this reptilian family began to live in the tops of the trees
    tokens of this reptilian family began to live in the tops of the trees
  tokens the members of this reptilian family began to live in the tops of the trees
 skipped_token ','
sentence_fragment
 nominal_clause
  wh_interro_declarative_clause
   wh_marker which
   statement
       .............
           .............
            attrs
             adj then pos
            attrs
             adj often pos
            tokens then often
     tokens were then often
skipped_token 'more_than'
 sentence_fragment
 entity
  head_noun
   noun
    n foot countnoun
    noun_modifiers
     pre_modif
      n hundred countnoun
     post_modif
      adj high pos
  number pl
  tokens hundred feet high
end_of_input period
```

Figure 3.1 DIPETT's Output

3.1.2 Machinese Syntax

Machinese Syntax [Tapanainen and Järvinen 1997] assigns functional dependencies, morphological tags and surface syntactic tags to words in a sentence. The parser is based on the Karlsson Constraint Grammar framework [Karlsson *et al.* 1995]. It assigns dependency links to words expressing head modifier relations. This is in accordance with our requirements, even though a direct map is not possible between the two. The parser output is in a tree form which is nicely detailed. The elaborate syntactic detail output by the parser makes it a good candidate for this study. The tree form with the dependency functions marked allows the potential pairs to be easily picked.

```
2> Once every thousand years a little bird comes to this rock to sharpen its beak.

1       Once            once            meta:>8         @ADVL %EH ADV
2       every           every                           @A> %>N DET
3       thousand        thousand        attr:>4         @A> %>N NUM CARD
4       years           year                            @OBJ %NH N NOM PL
5       a               a                               @A> %>N DET SG
6       little          little          attr:>7         @A> %>N A ABS
7       bird            bird            subj:>8         @SUBJ %NH N NOM SG
8       comes           come            main:>0         @+FMAINV %VA V PRES SG3
9       to              to              goa:>8          @ADVL %EH PREP
10      this            this            attr:>11        @A> %>N DET DEM SG
11      rock            rock            pcomp:>9        @<P %NH N NOM SG
12      to              to              pm:>13          @INFMARK> %AUX INFMARK>
13      sharpen         sharpen         cnt:>8          @-FMAINV %VA V INF
14      its             it              attr:>15        @A> %>N PRON GEN SG3
15      beak            beak            obj:>13         @OBJ %NH N NOM SG
16      .               .
17      <s>             <s>
```

Figure 3.2 Output from Machinese Syntax

Parsing a sentence outputs the structure in a numbered format. Every word is followed by its base form, dependency tag and the syntactic information. In the above example (Figure 3.2) the number 8 word is marked as the main element/the main verb of the clause. The tag %VA marks it is as the main verb, V signifies it to be a verb and PRES SG3 represent present tense and singular third person.

3.1.3 Xerox Incremental Parser

Xerox Incremental Parser (XIP) [Chanod *et al.* 2004] is a finite-state shallow parser. The parser promises to catch dependency relations between words which are farther spaced and also potentially in different sentences. This thesis is mostly concerned with the dependency relations within a sentence and Xerox Incremental parser captures all that is required and more. Xerox is a commercial-strength parser that ensures high quality in the parsed output. The output is in the form of annotated chunks and sets of dependency. Its modular and incremental approach allows the user to customize the grammar and also plug-in grammar for a different language. In this work only the English grammar provided by default in the parser is used. The output obtained using XIP for the same sentence used in Figure 3.2 is shown in Figure 3.3

```
2> Once every thousand years a little bird comes to this rock to sharpen its beak.
MOD_POST(<VERB:comes^come+s_subj_cont+Pres+3sg+VERB:7>,<NOUN:rock^rock^+Noun+Sg+NOU
N+hmm_verb:10>)
MOD_PRE(<VERB:comes^come^+s_subj_cont+Pres+3sg+VERB:7>,<ADV:Once^once^+CR+Adv+notly
+ADV:0>)
MOD_PRE_DUR(<VERB:comes^come^+s_subj_cont+Pres+3sg+VERB:7>,<NOUN:years^year^+Noun+P
l+NOUN:3>)
MOD_PRE(<NOUN:bird^bird^+Noun+countable+Sg+NOUN+hmm_verb:6>,<ADJ:little^little^+Adj
+ADJ+hmm_quantadv:5>)
MOD_POST_INFINIT(<NOUN:rock^rock^+Noun+Sg+NOUN+hmm_verb:10>,<VERB:sharpen^sharpen^+
Trans+Pres+Non3sg+VERB:12>)
DETD(<NOUN:bird^bird^+Noun+countable+Sg+NOUN+hmm_verb:6>,<DET:a^a^+Det+Indef+Sg+DET
:4>)
DETD(<NOUN:rock^rock^+Noun+Sg+NOUN+hmm_verb:10>,<DET:this^this^+Det+Sg+DET+hmm_pron
+hmm_adv:9>)
DETD(<NOUN:beak^beak^+Noun+countable+Sg+NOUN:14>,<PRON:its^it^+Pron+Pers+Gen+3P+Sg+
DET+hmm_pron:13>)
QUANTD(<NOUN:years^year^+Noun+Pl+NOUN:3>,<NUM:thousand^thousand^+Num+Card+CARD+hmm_
noun:2>)
SUBJ_PRE(<VERB:comes^come^+s_subj_cont+Pres+3sg+VERB:7>,<NOUN:bird^bird^+Noun+count
able+Sg+NOUN+hmm_verb:6>)
OBJ_POST(<VERB:sharpen^sharpen^+s_p_up+Trans+Pres+Non3sg+VERB:12>,<NOUN:beak^beak^+
Noun+countable+Sg+NOUN:14>)
PREPD(<NOUN:rock^rock^+Noun+Sg+NOUN+hmm_verb:10>,<PREP:to^to^+Infto+INFTO:8>)
PREPD(<VERB:sharpen^sharpen^+s_p_up+Trans+Pres+Non3sg+VERB:12>,<PREP:to^to^+Infto+I
NFTO+hmm_prep:11>)
MAIN(<VERB:comes^come^+s_subj_cont+Pres+3sg+VERB:7>)
```

Figure 3.3 Output from Xerox Incremental Parser

The parser pre-groups the words involved in a dependency relation and provides the syntactic information for them. The parser also outputs dependency relations which are not essentially useful for the purpose of this study; however these are handled later during the annotation process.

3.1.4 Link Parser

The Link parser [1995] gives a similar output to that of Machinese syntax but with less syntactic detail. For example parsing the same sentence as above with Link parser gives the output shown in Figure 3.4.

```
2> Once every thousand years a little bird comes to this rock to sharpen its beak.

          LEFT-WALL       Xp      <---Xp---->  Xp        .
 (m)      LEFT-WALL       Wd      <---Wd---->  Wd        years.n
 (m)      once            CO*s    <---CO*s-->  CO        years.n
 (m)      years.n         Sp      <---Sp---->  Sp        rock.v
 (m)      years.n         Ce      <---Ce---->  C         bird.n
 (m) (e)  a               Ds      <---Ds---->  Ds        bird.n
 (m) (e)  little.a        A       <---A----->  A         bird.n
 (m) (e)  bird.n          Ss      <---Ss---->  Ss        comes
 (m) (e)  comes           MV      <---MVp--->  MVp       to
 (m) (e)  to              J       <---Js---->  Js        this.p
 (m)      rock.v          MV      <---MVi--->  MVi       to
 (m) (e)  to              I       <---I----->  I         sharpen.v
 (m) (e)  sharpen.v       O       <---Os---->  Os        beak.n
 (m) (e)  its             D       <---Ds---->  Ds        beak.n
          .               RW      <---RW---->  RW        RIGHT-WALL

[S [SBAR Once SBAR] every thousand [S [NP years [SBAR [S [NP a little bird NP] [VP
comes [PP to [NP this NP] PP] VP] S] SBAR] NP] [VP rock [S [VP to [VP sharpen [NP
its beak NP] VP] VP] S] VP] S] . S]
```

Figure 3.4 Output from Link parser

Link parser's dependency relation output is comparable to Machinese Syntax and Xerox Incremental parser but it offers no advantage in terms of syntactic features over the other two parsers, hence resulting in it not being selected.

The other parsers that were considered included: Minipar [1998], SYNTPAR[3] and Apple Pie Parser [1996]. Minipar was ruled out due to same reasons as Link parser, whereas the other two were ruled out because they did not output dependency relations. The two parsers, Connexor's Machinese Syntax and Xerox Incremental Parser, offer detailed parses and capture the majority of the relations that are of interest in this thesis. In addition to that, both of these parsers are robust enough to handle the text being parsed here and do not suffer from the problems observed with DIPETT.

3.2 Finding features in a Knowledge Source

The majority of the work in automatic assignment of semantic labels uses only syntactic information [Gildea and Jurafsky 2002, Surdeanu *et al.* 2003, Thompson *et al.* 2003, Pradhan *et al.* 2005]. As it was discussed in chapter 2, a fair amount of work has been done in tuning the performance of such a system based on deep syntactic features, surface features and also using morphological information. With a better availability of large ontological/semantic databases, dictionaries and thesauri in machine-readable format, the importance of incorporating these resources into a system designed to capture semantics in a text cannot be challenged. These resources are the embodiment of semantic knowledge and it feels imperative to incorporate them into a system designed to capture semantics.

These resources, though possessing an incredible amount of knowledge, have their own shortcomings. The information available in machine-readable format in the majority of cases is still far from being machine tractable. WordNet is an exception to this rule and hence a widely accepted tool. In addition, at the author's university (University of Ottawa) there is access to an API [Jarmasz and Szpakowicz 2003] developed to provide an interface to Roget's Thesaurus [Kirkpatrick 1992]. This allows testing of the utility of Roget's for assigning semantic relations. In addition to Roget's and WordNet, Longman's dictionary of contemporary English (LDOCE) [1997] is also explored as a

[3] http://www.langsoft.ch/index.html

tool in learning semantic relations. The following sections review these semantic resources.

3.2.1 WordNet

WordNet [Fellbaum 1998] is an online lexical reference system whose design has been inspired by current psycholinguistic theories of human lexical memory. English nouns, verbs, adjectives and adverbs are organized into synonym sets called synsets, each representing one underlying lexical concept. WordNet provides information including definitions and examples for most of the synsets, which makes it comparable to a dictionary. It is also a thesaurus, as words belonging to the same syntactic category and having similar meaning are grouped together in synsets. WordNet includes, as a part of its synsets word collocations in the form of complex nominals, verbals adjectives and complex adverbials. Various relations are used to link the synonym sets forming a network of interrelated concepts. In WordNet the links are based on the conceptual-semantic relations that link the concepts rather than lexical relations, which link individual words. In this way WordNet provides explicit relationships among the synsets. The prominent link type in WordNet is the Hypernym relation, which divides the noun concepts into 11 hierarchies and verb concepts into 512 hierarchies. Hypernym relation allows one to go from a specific term to a generic term for example:

A car -> (is a kind of) -> automobile -> (is a kind of) -> means of transport.

WordNet is unarguably one of the most important resources in the natural language processing field and has been applied in almost all areas of research. Its application is seen in language generation [Jing, 1998], measuring semantic similarity [Pederson *et al.* 2004] and text classification [Scott and Matwin 1998] to name a few. Banarjee *et al.*[2003] look into the use of WordNet in order to find semantic relations between concepts using gloss overlap.

For this work a COM library object WNCON (http://wncom.sourceforge.net/) is used. WNCOM allows access to WordNet data via Java and Visual C++.

WordNet with all its benefits is still not without its shortcomings. Harabagiu *et al.* [1999] lists a number of weaknesses of WordNet as cited by the computer linguistics community. These include: lack of connections between noun and verb hierarchies, lack of morphological/topical relations and limited connections between topically related words. There were also some inconsistencies in the glosses, which have been fixed in the newer versions. One major shortcoming that is of most concern to this project is the unavailability of detail for adjectives and adverbs as compared to nouns and verbs. This problem prompts one to look beyond only WordNet. In order to overcome the shortcoming observed in WordNet, two types of resources which the WordNet represents: - a dictionary and a thesaurus were examined.

3.2.2 Longman Dictionary of Contemporary English (LDOCE)

Most dictionaries offer in them a huge resource containing the documented world knowledge. A conventional dictionary contains spelling, pronunciation, inflected and derivative forms, etymology, part of speech, definitions, synonyms and antonyms. With all the knowledge they contain and with the availability of dictionaries in machine-readable format the essential question still remain: Can the information in dictionaries be used to understand texts?

LDOCE dictionary has been widely used as a resource for extracting semantic information. One of the biggest advantages of LDOCE is it uses a small vocabulary of approximately 2000 words to define words. The electronic version of LDOCE in addition to containing the dictionary content also contains subject codes and activator codes for the entries. There are a total of 236 subject codes. The subject codes classify words by their subject, for example, one sense of bank is assigned the subject code 'BFB' which stands for *banking* and the other is assigned 'DN' which is the code for nature. The activator codes are assigned based on the semantic categories e.g. *Common, Beautiful, Dirty* etc. LDOCE assigns a total of 1052 key concepts as activator codes.

The subject codes and the activator codes are good potential features for automatically assigning semantic relations as they help categorize the input into 236 codes and 1052 concept words respectively.

For example consider two pairs (Source: Noun-Modifier Data[4]) in the form [Relation, Modifier, Head].

[meas, big, cloud]
[meas, giant, ball]

The two pairs, *big cloud* and *giant ball* both have the relation MEASURE assigned to them. The activator code for both *big* and *giant* is given as:

<ACTIV>BIG</ACTIV>

The activator code here helps in generalizing the modifiers for these two pairs. There are enough indicators available that point towards the utility of these two fields as potential features. Further generalization is tested by using the machine-learning systems.

LDOCE has been used in multiple ways in NLP. A thorough discussion on the utility of LDOCE was done by Michiels [1980]. Michiels [1982] uses LDOCE for automatic thesaurus generation, Nakamura [1988] explains the process of extracting the semantic information for noun and verbs. Byrd *et al.* [1985] acquire semantic features for lexical disambiguation.

[4] http://www.site.uottawa.ca/~vnastase/noun_modifier_data.html

3.2.3 Roget's Thesaurus

The word *Thesaurus* is new-Latin for treasury and is commonly used for a catalogue of words with similar, related or opposite meanings. It is also referred to as a dictionary in reverse: one searches a thesaurus for words starting from a meaning, in contrast to a dictionary which is used to look up meanings of words. The first thesaurus of English language was created by Peter Roget and since then the word 'Roget' has been synonymous with thesaurus. Roget's is not a trademark and a variety of thesauri are available under this name. The original Roget's thesaurus was published in 1852 and consisted of six primary classes namely: *Abstract relations, Space, Material World, Intellect, Volition, Sentient and Moral Powers.* The version of Roget's thesaurus used in this study is Penguin's *Roget's Thesaurus of English Words and phrases*, edited by Betty Kirkpatrick [1998]. This version divides the Roget's thesaurus into eight main classes:

1) Abstract Relation (e.g. state, relation, order, sequence etc.)
2) Space (shape, size, motion etc.)
3) Physics (heat, mechanics, physical properties etc.)
4) Matter (mankind, soil, universe etc.)
5) Sensation (Smell, sight, taste etc.)
6) Intellect (Faculties, assessment etc.)
7) Volition (will, custom, purpose etc.)
8) Affection (emotion, esteem, morals etc.)

These classes are divided into 39 sections, 79 subsections and further into 596 head groups. The head groups are broken into 990 heads. The entries within a Head in Roget's are organized into paragraphs grouped by Part-of-Speech tags (Noun, verbs, adjectives and adverbs). The paragraphs contain semicolon groups of semantically related words/phrases. An example of how this information is organized in Roget's is shown in Figure 3.5 of the head word *Wonder.* (Source: Jarmasz [2003])

Roget's thesaurus' utility in the natural language processing field is as old as the field itself. Masterman [1957], 50 years ago, stressed its use by reproducing Roget's thesaurus on punched cards and performed machine translation experiments. Sparck Jones [1964] did statistical classification by using words in the same rows in Roget's thesaurus as features. She clustered the words into head-like entities that could be identified with semantic primitives for the purpose of information retrieval or discourse analysis.

Yarowsky [1991] used Roget's thesaurus for word sense discrimination and showed 93% accuracy based on advanced statistical techniques.

Having a machine readable version of Roget's is not nearly enough in itself, extracting information from it is a big challenge. The actual Roget's data file is not easy to read and has special codes which are not easy to work with; to our advantage we have access to a Java-based API [Jarmasz 2003] which allows an easy access to Roget's. One of the outputs of the API is in the form of paths between two given entries in the thesaurus. The path is a list of words which when traversed takes us from one entry to the other in Roget's thesaurus. This output is sorted via a distance metric. An example of the output given by the API's path finder is shown in Figure 3.6.

Class six: Emotion, religion and morality

Section two: Personal emotion

Sub-section: Contemplative

Head Group: 864 Wonder - 865 Lack of wonder

Head: 864 Wonder

N. *wonder*, state of wonder, wonderment, raptness; admiration, hero worship, 887 *love*; awe, fascination; cry of wonder, gasp of admiration, whistle, wolf wolf, exclamation, exclamation mark; shocked silence, 399 *silence*; open mouth, popping eyes, eyes on stalks; shock, surprise, surprisal, 508 *lack of expectation*; astonishment, astoundment, amazement; stupor, stupefaction; bewilderment, bafflement, 474 *uncertainty*; consternation, 854 *fear*.

…

Adj. *wondering*, marvelling, admiring, etc. vb.; awed, awestruck, fascinated, spellbound, 818 *impressed*; surprised, 508 *inexpectant*; astonished, amazed, astounded; in wonderment, rapt, lost in wonder, lost in amazement, unable to believe one's eyes *or* senses; wide-eyed, round-eyed, pop-eyed, with one's eyes starting out of one's head, with eyes on stalks; open-mouthed, agape, gaping; dazzled, blinded; dumbfounded, dumb, struck dumb, inarticulate, speechless, breathless, wordless, left without words, silenced, 399 *silent*; bowled over, struck all of a heap, thunderstruck; transfixed, rooted to the spot; dazed, stupefied, bewildered, 517 *puzzled*; aghast, flabbergasted; shocked, scandalized, 924 *disapproving*.

…

Vb. *wonder*, marvel, admire, whistle; hold one's breath, gasp, gasp with admiration; hero-worship, 887 *love*; stare, gaze and gaze, goggle at, gawk, open one's eyes wide, rub one's eyes, not believe one's eyes; gape, gawp, open one's mouth, stand in amazement, look aghast, 508 *not expect*; be awestruck, be overwhelmed, 854 *fear*; have no words to express, not know what to say, be reduced to silence, be struck dumb, 399 *be silent*.

Chapter 3. The Design and Implementation of the ELKB

20

…

Adv. *wonderfully*, marvellously, remarkably, splendidly, fearfully; wondrous strange, strange to say, wonderful to relate, mirabile dictu, to the wonder of all.

Int. amazing! incredible! I don't believe it! go on! well Inever! blow me down! did you ever! gosh! wow! how about that! bless my soul! 'pon my word! goodness gracious! whatever next! never!

Figure 3.5 Roget's Thesaurus Head *Wonder*

```
Path between reptile and gigantic

Path between serpent 251 N. and huge 195 ADJ. (length = 14)

reptile --> serpent --> N. --> 251. Convolution: complex circularity -->
[250, 251] --> Special --> Section three : Form  --> Class two : Space  <--
Section two : Dimensions  <-- General <-- [195, 196] <-- 195. Size <-- ADJ.
<-- huge <-- gigantic


Path between animal 365 N. and enormous 32 ADJ. (length = 16)

reptile --> animal --> N. --> 365. Animality. Animal --> [365, 366] -->
Vitality --> Section three : Organic matter  --> Class three : Matter  --> T
<-- Class one : Abstract Relations  <-- Section three : Quantity  <--
Comparative <-- [32, 33] <-- 32. Greatness <-- ADJ. <-- enormous <-- gigantic
.............
```

Figure 3.6 Small sub-section of the paths given by the API

Here *length = 14* indicates the path was found in the same class and *length =16* the path was in the thesaurus. The word *Reptile* in the figure shown (Figure 3.6) has two references *Serpent* and *Animal* which occur under the Heads of *251.Convolution* and *365.Animal* respectively. *Animal* is found in the *Head Group: Vitality* which is in *Section Three: Organic Matter* and under the *Class: Matter*. The link *'T'* represents the root node of this hierarchical tree and links all the classes in the API. In a similar fashion, the path traverses down the root node T to *Class: Abstract Relations*, *Section: Quantity*, *Sub-section: Comparative*, *Head Group: Greatness* and the reference *Enormous* referring to *Gigantic*.

3.3 Machine Learning Algorithms

Machine learning is a process which allows the analysis of large sets of data in order to find any generalities which can used to classify data or make any predictions on that dataset. In the field of natural language processing machine learning is generally used as a tool to automatically find classification rules for data. This study uses two different machine learning algorithms to evaluate the accuracy of classification - Decision Trees and SVM. The choice of the algorithms is based on the specific requirement of the data available in this study. The data has high class imbalance and a few of the relations only have a handful of examples. Decision trees are known to perform well even when there is only a small set of data. SVMs are known to be impervious to class imbalance. These two in combination should sufficiently address the issues faced in the data.

3.3.1 Decision Trees

Decision trees are a classification mechanism in which rules for classification are constructed by creating a tree-like structure where each internal node represents a decision point. It maps an observation to an effective conclusion via the given attributes. Each internal node represents an attribute and the branches represent a possible value for the attribute. The leaf nodes are the final class labels gained by traveling the path from the root to the leaf (Figure 3.7). The construction of the tree is based on the principles of information theory. The attribute selection is based on the information gained by incorporating that attribute. The information gain of an attribute is calculated using the entropy, which gives the measure of disorder in the system. At any point the attribute which decreases the entropy by the highest value is selected as the next node. This is done till the final leaves are achieved with zero entropy and all the similar examples are labeled under the same class.

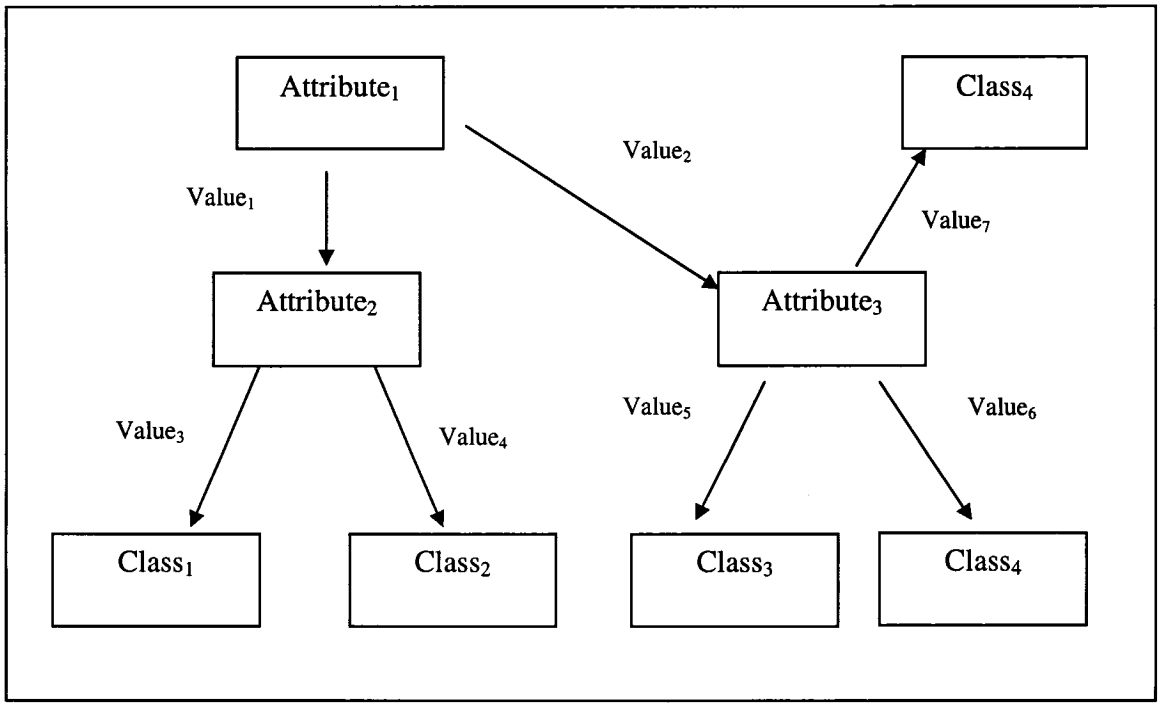


Figure 3.7 Decision Tree

The entropy or disorder is given by:

$$Entropy(T) = -\sum_{i=1}^{n}(p_i * \log(p_i))$$

T is a set of records partitioned into disjoint exhaustive classes C_1, C_2, .., C_k on the basis of the value of the categorical attribute. Pi is the portion of the i^{th} class given by:

$$P_i = \frac{|C_i|}{|T|}$$

Where $T = \sum_{i=0}^{n} C_i$.

The information gain of an attribute is the entropy reduction caused by partitioning on this attribute

$$\text{Gain}(T, A) = \text{Entropy}(T) - \sum_{v \in \text{Values}(A)} \frac{|T_v|}{|T|} \text{Entropy}(T_v)$$

This gain represents the information needed to identify an element of T after the value of attribute A has been obtained, that is the gain in information due to attribute A.

This method of calculating gain tends to favor attributes that occur in larger numbers. To compensate for this, Quinlan [1996] suggests using the following ratio instead of Gain:

$$GainRatio(T, A) = \frac{Gain(T, A)}{SplitInfo(T, A)}$$

Here SplitInfo (T, A) is the information based on the split of T on the basis of the value of categorical attribute A instead of C_i. There is also a pruning process where a sub-tree is replaced by a node, if the decision tree calculates the expected error rate of a node to be less than that of the sub-tree.

The decision trees are usually accepted to perform well even with a small set of data. The decision tree implementations used in this study are J48 from Weka [2005] and Rulequest C5.0[5].

3.3.2 Support Vector Machine

SVM (Support Vector Machine) is a machine learning classifier based on linear regression. SVM classifies data using weighted attributes by finding a dividing hyper-plane in an N-dimensional space that is equidistant from the two classes. The classifier

[5] http://www.rulequest.com/see5-info.html

can be extended to a multi-class classifier by constructing multiple two-class planes. SVM begins by first finding the linear separation between the classes and then extends it to a non-linear boundary by applying some nonlinear kernel functions. The classifier calculates support vector points in a space that are closest to an optimally dividing hyper-plane. This optimal plane has the property of having the highest margin of separation between the two classes. This process allows the algorithm to process only a subset of the data rather then using the entire data. The essential process is illustrated in Figure 3.8 below.

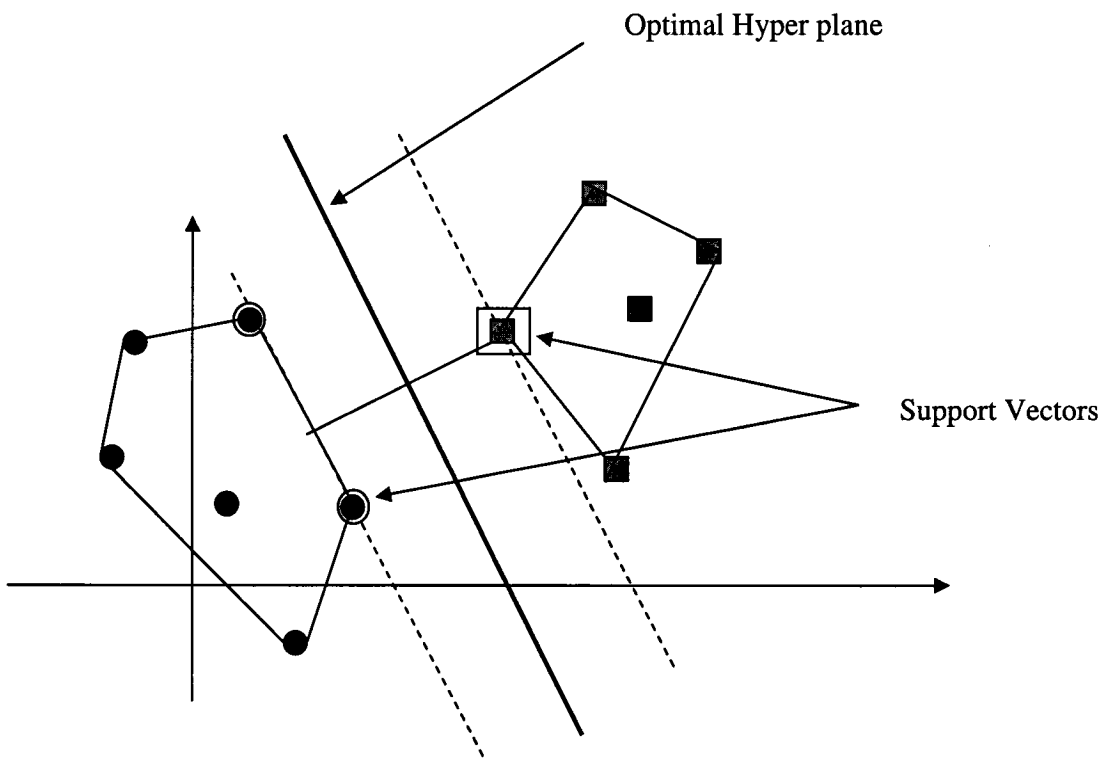


Figure 3.8 Support Vector Machines

The decision function employed in the process is:

$$F(x) = sign(\sum_{i=0}^{n} wi(x_i \bullet y) + b)$$

It uses a dot product to compute the distance between an unlabeled point and the support vectors to evaluate which class-label is the most appropriate. Here x_i is the attribute values of a support vector and y is the unlabeled instance, and wi are the weights learned by the algorithm to allow for the greatest separation of the two classes. The parameter b is a bias factor to shift the hyper-plane over by the specified amount. The weights of the attributes are learned so as to ensure the best possible fit to the set of points provided. The linear separation achieved in this manner has a limitation that it can only be applied to linearly separable problems. To extend this into a non-linear process the function is mapped on to an n-dimensional feature space F from the vector space X and can be represented as:

$$\Phi: X \rightarrow F$$

This requires the evaluation of the mapped dot product as shown in the equation

$$K(x, y) => (\Phi(x) \bullet \Phi(y)).$$

The problem with this mapping is that the number of features can become unmanageable in the n-dimensional feature space and can be computationally very expensive. Fortunately, some kernel functions (K) exist which allow to compute this dot product independent of the dimensionality of the vector.
A few of these kernels are:

• Polynomial:

$$K(x, y) = (\gamma x^T y + r)^d, \quad \gamma > 0.$$

• Radial basis function (RBF):

$$K(x, y) = \exp (-\gamma \|x - y\|^2), \gamma > 0.$$

• Sigmoid:

$$K(x, y) = \tanh(x^T y + r).$$

Here γ, r and d are training parameters. Each of these kernels creates a boundary in the hyperspace to produce a non-linear decision function. This choice in kernels gives SVM flexibility in mapping the problem over the hyperspace. This flexibility also comes with a responsibility of finding the most appropriate parameters to have an optimal solution.
In this work, an implementation of SVM called libSVM [Chang, Lin 2001] has been used. The input to the software is a scaled sparse vector with the zero-valued attributes removed. For example, for a feature vector <1, 0, 0, 6, 0, 1> the input takes the form <1:1, 4:6, 6:1>

LIBSVM was run using the polynomial kernel versus using the default option which is the RBF kernel.

3.4 Kappa: Coefficient of Agreement

The quality of any annotated data plays a very important role in achieving accurate results and also dictates whether similar results can be replicated by someone other than the author. This in effect means that one requires a mechanism with which one can measure the quality of annotated data used in their experiments by finding the level of agreement in the annotations with a human expert or against a widely accepted gold standard. Previously many different methods have been reported to find such a measure but these methods are not without shortcomings. It was only recently that a reliable measure in form of Kappa was introduced to the NLP community by Carletta [1996]. Before this, the Kappa coefficient [Siegel, Castellan 1988, Cohen 1960, and Krippendorff 1980] had been extensively used in the field of content analysis and medicine for statistical synthesis.

Kappa co-efficient (κ) gives a measure of agreement between two or more annotators on the same task. It measures the agreement on top of any chance agreement that may affect the overall agreement in the task.

It is calculated as:

$$\kappa = \frac{P(A) - P(E)}{1 - P(E)}$$

Here P(A) is the actual observed agreement between the two annotators, and P(E) is the probability of agreement by chance. The value of κ lies in the interval -1 to 1. A value of 1 shows perfect agreement and 0 means the agreement is completely based on chance. A -1 stands for a total disagreement. The negative values, though theoretically possible, are hardly ever observed, as that would mean that the actual observed agreement is smaller than agreement expected by chance.

There are different scales proposed by different communities in order to interpret the value of Kappa. The scale given by Krippendorff discounts the possibility of agreement for any values of $\kappa < 0.67$ and it rates the outcome as having medium agreement for κ values between $0.67 < \kappa < 0.8$. Values of $\kappa > 0.8$ suggest high agreement. Other scales like the one proposed by Rietveld and Van Hout [1993] consider a value of $0.20 < \kappa < 0.40$ to indicate fair agreement, and a value of $0.40 < \kappa < 0.60$ is taken to mean moderate agreement and substantial agreement for any value of $\kappa > 0.61$. The Krippendorff scale as such is the most widely accepted scale even though Krippendorff himself states that the scale is only an indicative standard and has been derived from his own and his colleagues' work. He also indicates that any scale used should not discount the importance of the data on which the analysis is being conducted. This leaves the user with the onus to decide which scale is to be used for their data. A generally accepted norm in the language community is that having a value for κ greater than 0.6 indicates a fair agreement in the annotation task.

Chapter 4: The Data Set for Learning

4.1 The Data Source – Text

The data used in previous work on the TANKA system [Barker 1998, Nastase 2003] comprised of a small and simple text of about 600 sentences [Barker, Delisle 1996] which had approximately 2000 pairs of word units annotated with semantic relations. The source of the data was a Junior Science Book titled *Rain, Hail, Sleet and Snow* [Larrick 1961]. Apart from the simplistic nature, the data was not large enough to provide for the learning task. Eight out of the total forty-seven relations had no instances, and another twenty five of them had fewer than twenty instances. In addition to this the data was plagued with severe class imbalance. With more than 50% of the relations having less than forty instances in the data set, the data proved inadequate as test and training data for the process of automatically learning semantic relations.

The severe imbalance coupled with the fact that a large number of relations had no instances in the data gave rise to a distinct need to have a new dataset. The choice of the data source was difficult; a literary but fairly factual text was chosen. The selected text finds itself under the literature category but tells a factual story of the development of mankind starting from a cellular being. The text is called 'The Story of Mankind' written by Hendrik Willem Van Loon [Project Gutenberg[6]]. The choice was dictated by the desire to have a higher number of complex representations of the relations in the sentences as opposed to having simple relations which can be classified relatively easily, e.g., agent, object etc. The text incorporates the use of metaphor, hence making the assignment of relations a challenging job for humans as well. The book has a reading level of 'Young adult' and is a harder text to read than the text previously used by the system. It is difficult to predict how the system will perform with different texts without formally experimenting with them. The relation of degree of automation with different

[6] Project Gutenberg -- http://www.gutenberg.org/etext/754

types of text is left as a future work for now. A simplistic text or a domain-specific text might be a better overall fit for the task and give better results, however the scenario examined here with a literary text versus a semi-technical text was chosen to push the system to its limits.

A small excerpt from the story is shown in Figure 4.1.

```
For millions of years it drifted aimlessly with the currents. But during
all that time it was developing certain habits that it might survive more
easily upon the inhospitable earth. Some of these cells were happiest in
the dark depths of the lakes and the pools. They took root in the slimy
sediments which had been carried down from the tops of the hills and they
became plants. Others preferred to move about and they grew strange jointed
legs, like scorpions and began to crawl along the bottom of the sea amidst
the plants and the pale green things that looked like jelly-fishes. Still
others (covered with scales) depended upon a swimming motion to go from
place to place in their search for food, and gradually they populated the
ocean with myriads of fishes.
```

Figure 4.1 Excerpt from "History of Mankind"

It has been previously shown [Delisle 1994, Barker 1998] in a semi-automatic acquisition process in TANKA that the use of continuous text helps speed up the memory-based learning of the system. This in effect decreases the work required by the user. The issue that arises from the usage of a continuous text is that it makes the problem of class imbalance unavoidable. The count of different relations present in a text can never be balanced and hence the imbalance will always prevail. The distribution of relations in the selected text is shown later in this chapter.

In keeping with the idea of having a continuous text, the text selected was longer than what is used in the experiments here. This allows for possible extension of the data set in

the future. The target requirement is to have around 5000 pairs of annotated data. The source text was divided into three parts. Each part has in excess of 15,000 relation pairs. This gave enough leeway to reject any pairs which were not semantically relevant to the work in this thesis. This is discussed in the next section. In this work, the first part of the data is annotated and analyzed. The other two parts of the text provide a possibility of continuing to create a larger data set as and when required. The statistics relating to the data are shown in the Table 4.1.

The process of extracting the pairs from the parser is detailed in section 4.2. A total of 9,910 pairs were annotated manually with 5,006 pairs for Machinese Syntax and 4,904 for Xerox Incremental Parser. The entire process was carried out separately for the two parsers, as the pairs vary depending on the parser. The pairs were then assigned relations using the semi-automatic process of TANKA.

4.2 Extraction of Pairs

The selected text is parsed by using the two parsers chosen as discussed in chapter 3. The choice of two parsers allows a comparison of the relative performance of the task as dependent on the parsing process.

The two selected parsers are:

- Machinese Syntax by Connexor
- Xerox Incremental Parser.

Before using the output of the parser, the parses must be transformed into a Prolog-friendly format which can be understood by HAIKU. The next section details this process.

Text	
Title of the text	The story of Mankind
Author	Van Loon, Hendrik Willem
Source	Project Gutenberg (Publisher: Liveright Publishing Corp)
Statistics for Machinese Syntax for the first Part of the book	
Total number of Pairs	15,521
Pairs Analyzed Manually	11000+
Pairs accepted and annotated with a relation	5006
Avg. number of pairs per Sentence output by the parser	15.45
Avg. Number of pairs accepted per sentence using TANKA	7.48
Statistics for Xerox Incremental Parser for the first Part of the book	
Total number of Pairs	15,521
Pairs Analyzed Manually	10000+
Pairs accepted and annotated with a relation	4,904
Avg. number of pairs per Sentence output by the parser	16.75
Avg. Number of pairs accepted per sentence using TANKA	7.94

Table 4.1 Statistical information of the selected text

4.2.1 Adapting Machinese Syntax

The main task is to transform the output of the parser into a format that is Prolog-friendly and can be interfaced with HAIKU. The parser outputs dependency functions as well as morphological information and functional tags. A complete list of the dependency functions is available in Appendix B. All of the information provided by the parser is used as a part of the learner input. The pairs are extracted by matching the entries in the parser's output.

An example of the parser's output is shown in the Figure 4.2. Figure 4.2 is the same as Figure 3.2 and has been reproduced here for the reader's convenience.

```
3>Once every thousand years a little bird comes to this rock to
sharpen its beak

1     Once        once        meta:>8     @ADVL %EH ADV
2     every       every                   @A> %>N DET
3     thousand    thousand    attr:>4     @A> %>N NUM CARD
4     years       year                    @OBJ %NH N NOM PL
5     a           a                       @A> %>N DET SG
6     little      little      attr:>7     @A> %>N A ABS
7     bird        bird        subj:>8     @SUBJ %NH N NOM SG
8     comes       come        main:>0     @+FMAINV %VA V PRES SG3
9     to          to          goa:>8      @ADVL %EH PREP
10    this        this        attr:>11    @A> %>N DET DEM SG
11    rock        rock        pcomp:>9    @<P %NH N NOM SG
12    to          to          pm:>13      @INFMARK> %AUX INFMARK>
13    sharpen     sharpen     cnt:>8      @-FMAINV %VA V INF
14    its         it          attr:>15    @A> %>N PRON GEN SG3
15    beak        beak        obj:>13     @OBJ %NH N NOM SG
16    .     .
17    <s>   <s>
```

Figure 4.2 Parsed output from Machinese Syntax

For example, the parser indicates that the word 'little' is related to the entry 7 which is 'bird' and the dependency function tag between the two: attr (Attribute Nominal).

In the above example the pairs picked are:

[Once],[comes]

[thousand],[years]

[little],[bird]

[bird],[comes]

[to],[comes]

[this],[rock]

[rock],[to]

[to],[sharpen]

[sharpen],[comes]

[its],[beak]

[beak],[sharpen]

At this stage a database of all the pairs is created and any rejection of unnecessary pairs is done later during the semi-automatic process of assigning relations using HAIKU. For example, in the above figure, the parser assigns 'Attribute Nominal (attr)' as the dependency function between *'This'* and *'rock'*. This particular pair is not semantically important but is still included in the database that is used as input for HAIKU. This pair is then rejected during the second phase when the relations are being assigned.

The final database predicate for Prolog is of the following format:

connx_info (SentID, Head, Modifier, Head _ fn_tag, Head _surface_syn_tag, [Head _syn_info], Modifier _fn_tag, Modifier _surface_syn_tag, [Modifier _syn_info]).

Table 4.2 provides an explanation for the variables used in the prolog predicate.

Variable	Explanation
SentId	Sentence Id; unique for each sentence
Head	Head part of the pair
Modifier	Modifier part of the pair
Head_fn_tag	Function tags for the head word

Variable	Explanation
Head_surface_syn_tag	Surface Syntactic tag for head word
Head_syn_info	POS followed by a list of 10 elements containing information like tense, number etc.
Modifier_fn_tag	Function tags for the Modifier word
Modifier_surface_syn_tag	Surface Syntactic tag for Modifier word
Modifier_syn_info	POS followed by a list of 10 elements containing information like tense, number etc.

Table 4.2: Explanation of the features in the Prolog predicate

The final output of the data structure for the pairs is shown in the Figure 4.3 below.

```
connx_info(3,freq,'once','once','come','comes','adverbial','intr_coord',[ad
verb,nil,nil,nil,nil,nil,nil,nil,nil,nil,nil],'fin_main_pred','verb_active'
,[verb,nil,nil,present,sg_third,nil,nil,nil,nil,nil,nil]).

connx_info(3,meas,'thousand','thousand','year','years','premod','det_premod
',[nil,nil,nil,nil,nil,nil,nil,nil,nil,nil,nil],'obj','nom_head',[noun,nom,
pl,nil,nil,nil,nil,nil,nil,nil,nil]).

connx_info(3,prop,'little','little','bird','bird','premod','det_premod',[ad
jective,nil,nil,nil,nil,absolutive,nil,nil,nil,nil,nil],'subj','nom_head',[
noun,nom,sg,nil,nil,nil,nil,nil,nil,nil,nil]).

connx_info(3,agt,'bird','bird','come','comes','subj','nom_head',[noun,nom,s
g,nil,nil,nil,nil,nil,nil,nil,nil],'fin_main_pred','verb_active',[verb,nil,
nil,present,sg_third,nil,nil,nil,nil,nil,nil]).


connx_info(3,purp,'come','comes','sharpen','sharpen','fin_main_pred','verb_
active',[verb,nil,nil,present,sg_third,nil,nil,nil,nil,nil,nil],'fin_main_p
red','verb_active',[verb,nil,nil,nil,nil,nil,nil,nil,nil,nil,infinitive]).

connx_info(3,obj_prop,'sharpen','sharpen','beak','beak','fin_main_pred','ve
rb_active',[verb,nil,nil,nil,nil,nil,nil,nil,nil,nil,infinitive],'obj','nom
_head',[noun,nom,sg,nil,nil,nil,nil,nil,nil,nil,nil]).
```

Figure 4.3 Final structure of the pairs selected from Machinese Syntax

4.2.2 Adapting Xerox Parser

The Xerox parser outputs the parses as shown in the Figure 4.4. This is the same as Figure 3.3 reproduced here for convenience.

```
3> Once every thousand years a little bird comes to this rock to sharpen its beak
.
MOD_POST(<VERB:comes^come^+Verb+s_subj_cont+Pres+3sg+VERB:7>,<NOUN:rock^rock^+Noun
+Sg+NOUN:10>)
MOD_PRE(<VERB:comes^come^+Verb+s_subj_cont+Pres+3sg+VERB:7>,<ADV:Once^once^+CR+Adv
+notly+ADV:0>)
MOD_PRE_DUR(<VERB:comes^come^+Verb+s_subj_cont+Pres+3sg+VERB:7>,<NOUN:years^year^+
Noun+Pl+NOUN:3>)
MOD_PRE(<NOUN:bird^bird^+Noun+countable+Sg+NOUN+hmm_verb:6>,<ADJ:little^little^+Ad
j+ADJ+hmm_quantadv:5>)
MOD_POST_INFINIT(<NOUN:rock^rock^+Noun+Sg+NOUN+hmm_verb:10>,<VERB:sharpen^sharpen^
+Verb+s_p_up+Trans+Pres+Non3sg+VERB:12>)
DETD(<NOUN:bird^bird^+Noun+countable+Sg+NOUN+hmm_verb:6>,<DET:a^a^+Det+Indef+Sg+DE
T:4>)
DETD(<NOUN:rock^rock^+Noun+Sg+NOUN+hmm_verb:10>,<DET:this^this^+Det+Sg+DET+hmm_pro
n+hmm_adv:9>)
DETD(<NOUN:beak^beak^+Noun+countable+Sg+NOUN:14>,<PRON:its^it^+Pron+Pers+Gen+3P+Sg
+DET+hmm_pron:13>)
QUANTD(<NOUN:years^year^+Noun+Pl+NOUN:3>,<NUM:thousand^thousand^+Num+Card+CARD+hmm
_noun:2>)
SUBJ_PRE(<VERB:comes^come^+Verb+s_subj_cont+Pres+3sg+VERB:7>,<NOUN:bird^bird^+Noun
+countable+Sg+NOUN+hmm_verb:6>)
OBJ_POST(<VERB:sharpen^sharpen^+Verb+s_p_up+Trans+Pres+Non3sg+VERB:12>,<NOUN:beak^
beak^+Noun+countable+Sg+NOUN:14>)
PREPD(<NOUN:rock^rock^+Noun+Sg+NOUN+hmm_verb:10>,<PREP:to^to^+Infto+INFTO:8>)
PREPD(<VERB:sharpen^sharpen^+Verb+s_p_up+Trans+Pres+Non3sg+VERB:12>,<PREP:to^to^+I
nfto+INFTO+hmm_prep:11>)
MAIN(<VERB:comes^come^+Verb+s_subj_cont+Pres+3sg+VERB:7>)

3>TOP(0:15){SC(0:7){ADV{Once^once^+CR+Adv+notly+ADV:0},QUANT{every^every^+Quant+QU
ANT:1},NP(2:3){NUM{thousand^thousand^+Num+Card+CARD+hmm_noun:2},NOUN{years^year^+N
oun+Pl+NOUN:3}},NP(4:6){DET{a^a^+Det+Indef+Sg+DET:4},AP(5:5){ADJ{little^little^+Ad
j+ADJ+hmm_quantadv:5}},NOUN{bird^bird^+Noun+countable+Sg+NOUN+hmm_verb:6}},FV(7:7)
{VERB{comes^come^+Verb+s_subj_cont+Pres+3sg+VERB:7}}},PP(8:10){PREP{to^to^+Infto+I
NFTO:8},NP(9:10){DET{this^this^+Det+Sg+DET+hmm_pron+hmm_adv:9},NOUN{rock^rock^+Nou
n+Sg+NOUN+hmm_verb:10}}},IV(11:12){PREP{to^to^+Infto+INFTO+hmm_prep:11},VERB{sharp
en^sharpen^+Verb+Trans+Pres+Non3sg+VERB:12}},NP(13:14){PRON{its^it^+Pron+Pers+Gen+
3P+Sg+DET+hmm_pron:13},NOUN{beak^beak^+Noun+countable+Sg+NOUN:14}},PUNCT{.^.^+Punc
t+Sent+SENT:15}}
```

Figure 4.4 Parsed output from Xerox Incremental Parser

Xerox parser outputs the information as pairs and also provides a dependency relation for each pair. The pairs are used as output by the Xerox parser in the preprocessing step. The rejection of any semantically unimportant pairs happens in the semi-automatic assignment phase.

The following is the list of pairs extracted for the sentence:

[comes],[rock]
[comes],[once]
[comes],[years]
[bird],[little]
[rock],[sharpen]
[bird],[a]
[rock],[this]
[beak],[its]
[years],[thousand]
[comes],[bird]
[sharpen],[beak]
[rock],[to]
[sharpen],[to]

The following illustrates the format of the final predicate for Prolog database:

xip_info (SentID, Head_stem, Head, Head _POS, Modifier _stem, Modifier, Modifier _POS, dependency_tag, [Head _syn_info], [Modifier _syn_info]).

Table 4.3 provides an explanation of the terms used in the prolog predicate shown above.

Variable	Explanation
SentId	Sentence Id; unique for each sentence
Head	Head
Head_stem	Uninflected version of the head

Variable	Explanation
Modifier	Modifier
Modifier _stem	Uninflected version of the head
Head_POS	POS
Modifier _POS	POS
Dependency tag	Dependency tag assigned by the parser.
Head_syn_info	Syntactic information obtained from the parser for the Head word
Modifier_syn_info	Syntactic information obtained from the parser for the Modifier word

Table 4.3 Explanation of the features in the prolog predicate

The final output of the data structure for the pairs is shown in the figure 4.5 below.

```
xip_info(3,lto,'come','comes','rock','rock',mod_post,verb,['pres','3sg'],
noun,['sg']).

xip_info(3,freq,'once','once','come','comes',mod_pre,adv,[nil],verb,['pre
s','3sg']).

xip_info(3,prop,'little','little','bird','bird',mod_pre,adj,[nil],noun,['
countable','sg']).

xip_info(3,purp,'rock','rock','sharpen','sharpen',mod_post_infinit,noun,[
'sg'],verb,['trans','pres','non3sg']).

xip_info(3,poss,'beak','beak','it','its',detd,noun,['countable','sg'],pro
n,['pers','gen','3p','sg']).

xip_info(3,meas,'thousand','thousand','year','years',quantd,num,['card'],
noun,['pl']).

xip_info(3,agt,'bird','bird','come','comes',subj_pre,noun,['countable','s
g'],verb,['pres','3sg']).

xip_info(3,obj_prop,'sharpen','sharpen','beak','beak',obj_post,verb,['tra
ns','pres','non3sg'],noun,['countable','sg']).
```

Figure 4.5 Final structure of the pairs selected from Xerox Increment Parser

4.3 Plugging in the Parser Output into HAIKU

The input to HAIKU from DIPETT is different than the output retrieved from the Machinese Syntax and Xerox Incremental parsers and therefore requires some preprocessing. Before plugging in the data from any other parser, a wrapper script must be created to convert the data extracted from the parser's output into a format HAIKU understands. HAIKU has an advantage in terms of its implementation in Prolog and in its use of list structures. Any new information that is not a part of the data obtained from DIPETT can be incorporated in HAIKU by appending it at the end of the list in addition to the required information. This ensures we are able to use the information provided by other parsers in addition to what the system allows to be directly plugged in it.

HAIKU expects input to be formatted as a simplified tree structure with pointers to a database which contains the pair's information. This is discussed in detail in Nastase [2003] and briefly here. The database entry has the following format:

Root (Lemma, LemmaID, Information)

Here *Lemma* represents the uninflected form of the word in the sentence. *LemmaID* represents an internally assigned ID for that word.

Information represents a list with following structure:

[SentenceID, Counter, Syntactic_information]

SentenceID is the numeric ID assigned to a sentence and the *Counter* is a sequentially increasing number marking the number of occurrences of that particular word.
The *Syntactic_information* field has two possible formats; which format gets used depends on whether the word is an entity or a predicate. The entity here means a syntactic structure whose head is an entity - a person, thing or a phenomenon, etc, which is

expressed through a noun or a pronoun. A predicate node is syntactic structure whose head is a verb.
The entity node uses the following format

> *[PartofSpeech , StructureType, WordType, s_role(syntactic role, Indicator), intensifier, predeterminer, Determiner, Post_determiner, Number)*

The predicate node follows the format:

> *[PartofSpeech, StructureType, VerbType, s_role (SyntacticRole, Indicator) , s_str(SubcategorizationStructure), Tense, Participle, Polarity, Transitivity, Voice]*

Here StructureType, s_role are elements from DIPETT's grammar and detailed in Nastase [2003].
This process is carried out sentence by sentence as the information is read into HAIKU. The information obtained from the parsers is thus converted into a database with the above mentioned structure. The pairs are indicated by passing the corresponding IDs to HAIKU. Any information that cannot be obtained from the new parsers is passed as 'nil'.

4.4 Assigning Semantic Relations

After having the database in the required format, the next step is the assignment of relations using HAIKU. The process of assignment of semantic relations is a long and difficult process. Average time taken to assign a relation is around 2-3 minutes; for 10,000 relations this amasses to 416 hours or about 10.4 weeks of work.
Figure 4.6 shows a snapshot of this process.

In the example in Figure 4.6, the system is started cold without any existing database and therefore this is the first sentence seen by the system. HAIKU first presents the user with

the sentence followed by the pair as picked from the database. The creation of this database was described in the previous section: "Plugging the parser input into HAIKU". The user is asked to accept the pair as a potential pair or reject it. Any pair that is not semantically important is manually rejected at this stage. The system also allows switching the head and the modifier if the relation is better represented by the switch. The system then runs the heuristics described in chapter two and tries to come up with a possible relation. If it cannot find any suitable relation, the user is presented with the complete list of relations and asked to pick one of the relations. After a relation has been assigned by the user, the system saves the state and presents the next pair.

```
Analysing sentence number 462:   Soon afterwards he ventured forth upon the sea and
conquered all the cities of the Aegean

Do you accept the following pair and the order: (head,modifier) conquer venture
[rj,y,n]       n

      venture conquer
Possible relations: [lat,dir,tat,freq,excl,tthr,lthr,lfrm,inst,meas,enab,ord,det
r,manr,ornt,lto,cont,purp,acmp,top,entl,prev,tfrm,caus,tto,agt,benf,obj,prop,matr,effe
,conj,recp,prec,stat,disj,ctn,ltd,poss,expr,equa,cooc,prod]
Many possibilities. Enter your choice:
[caus,effe,purp,entl,enbl,detr,prev,cooc,freq,prec,tat,tfr,tthr,tto,dir,lfrm,lto,lthr,
lat,ltd,ornt,conj,disj,acmp,agt,ben,excl,st,prop,posr,inst,obj,obj_prop,recp,part,whol
e,prod,src,cont,top,cntr,manr,matr,meas,ord,equa,type]  purp



New addition to some file:
        asgn_info(cl,purp,[venture,conquer],462,[n,43,choose])


New addition to some file:
        relation(purp,4,1,[[462,1,1]])

Do you accept the following pair and the order: (head,modifier) city conquer
[rj,y,n]       n

conquer city
Possible relations: [lat,dir,tat,freq,excl,tthr,lthr,lfrm,inst,meas,enab,ord,det
r,manr,ornt,lto,cont,purp,acmp,top,entl,prev,tfrm,caus,tto,agt,benf,obj,prop,matr,effe
,conj,recp,prec,stat,disj,ctn,ltd,poss,expr,equa,cooc,prod]
Many possibilities. Enter your choice:
[caus,effe,purp,entl,enbl,detr,prev,cooc,freq,prec,tat,tfr,tthr,tto,dir,lfrm,lto,lthr,
lat,ltd,ornt,conj,disj,acmp,agt,ben,excl,st,prop,posr,inst,obj,obj_prop,recp,part,whol
e,prod,src,cont,top,cntr,manr,matr,meas,ord,equa,type]  obj



New addition to some file:
        asgn_info(ic,obj,[conquer,city],462,[n,43,choose])


New addition to some file:
        relation(obj,5,4,[[462,1,1]])

Do you accept the following pair and the order: (head,modifier) and venture
[rj,y,n]       rj
```

Figure 4.6 Snapshot of HAIKU

4.5 Distribution of Semantic Relations in the Dataset

The following chart shows the distribution of pairs assigned using Machinese Syntax. The abbreviations for the relations are discussed in chapter 2 and are also available for reference in appendix A.

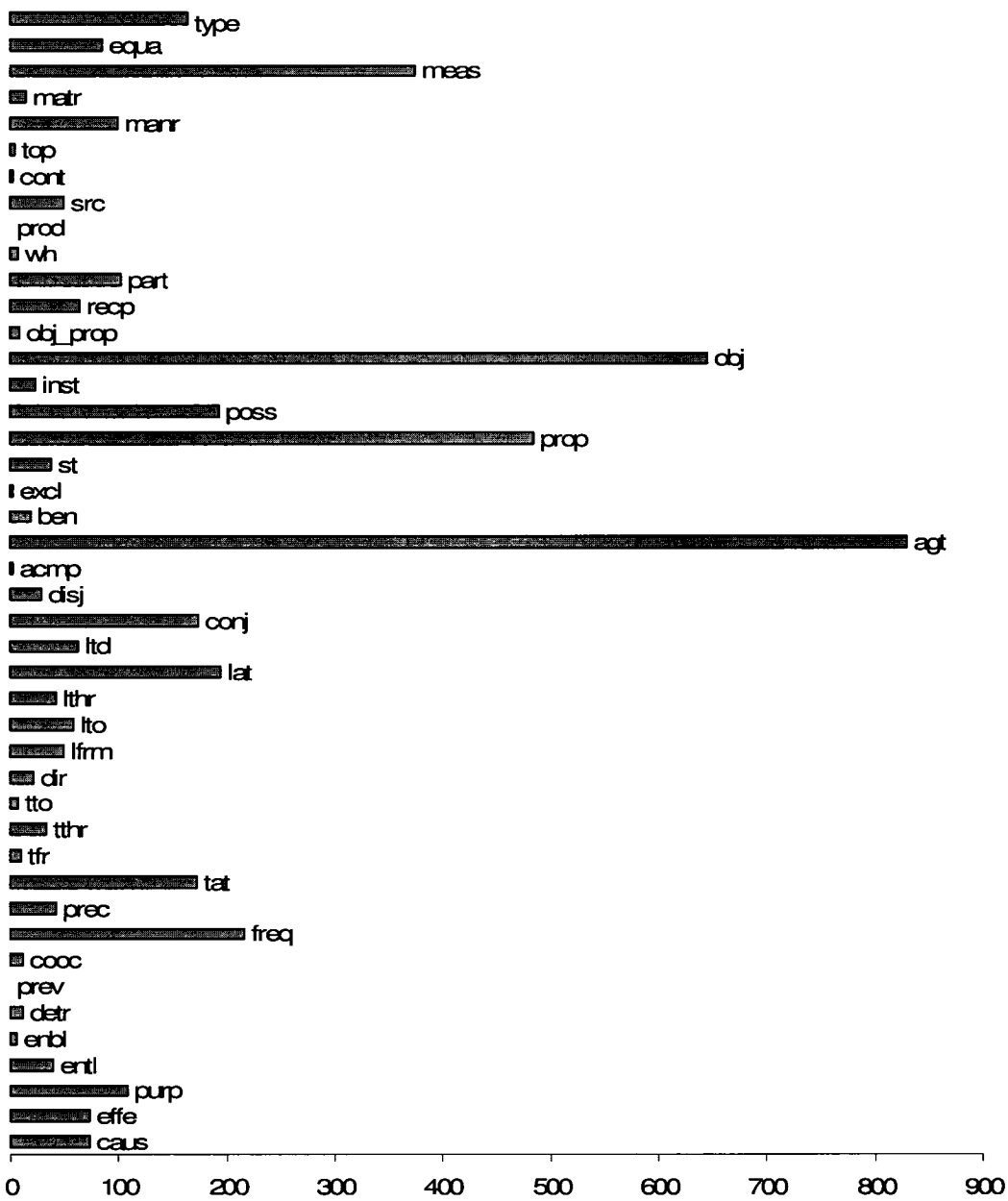


Figure 4.7 Distribution of relations in Machinese Syntax-based Dataset

The relations with the highest frequencies are agent, object, property, and measure. The relations near the bottom of the frequency chart are enablement, prevention, exclusion, order, orientation, product, co-occurrence and content. This distribution is highly data-dependent and thus depicts only the information for this particular text.

The following chart shows the distribution of pairs assigned using XIP:

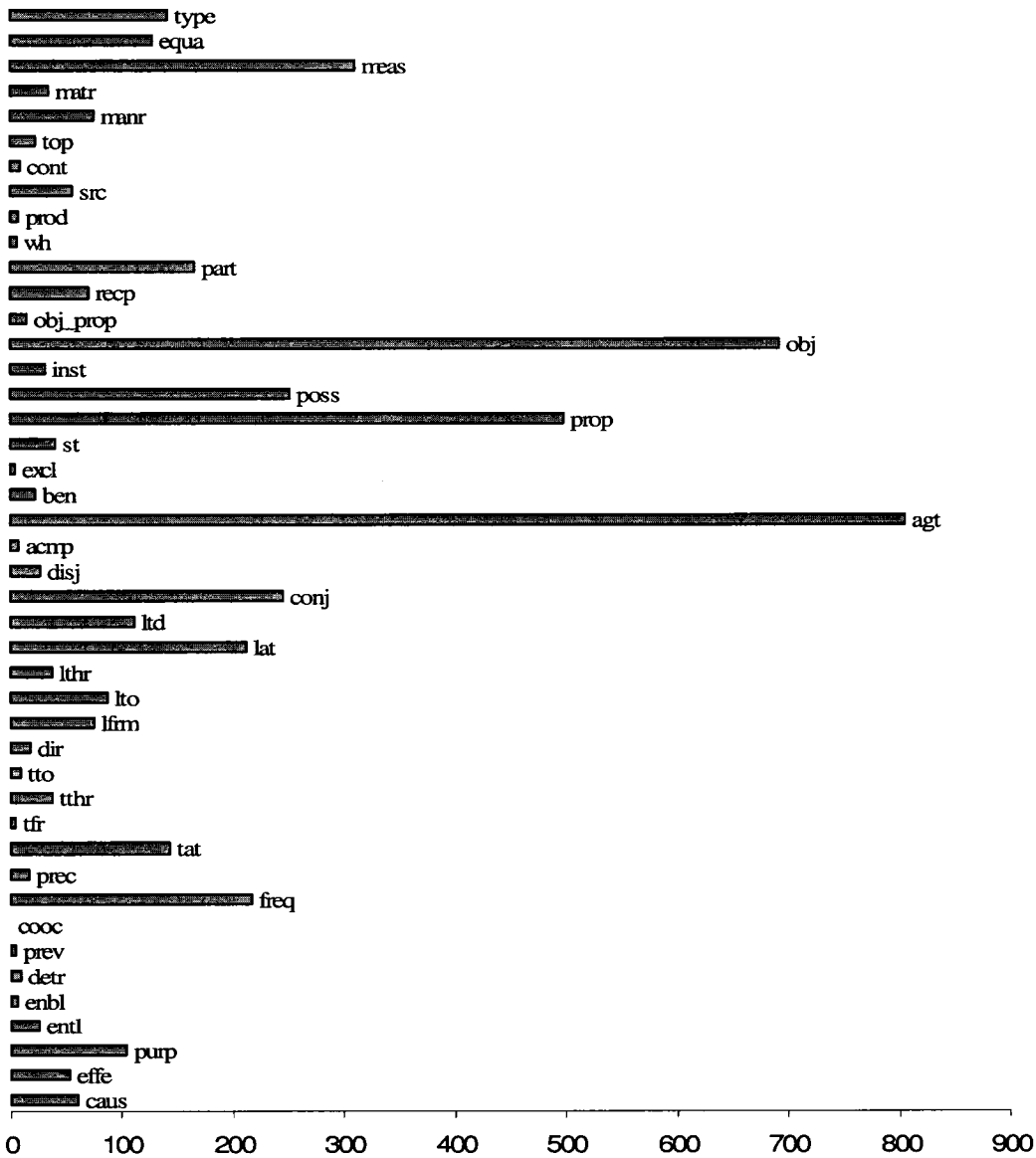


Figure 4.8 Distribution of relations in Xerox Parser based Dataset

The relations with the highest frequencies are: agent, property measure, and object. The list of lower frequency relations includes enablement, prevention, exclusion, order, orientation, product, co-occurrence and content.

Figure 4.9 gives a comparison of the distribution of relations in the two datasets – Machinese Syntax-based dataset and the Xerox Incremental Parser-based dataset. The semantic relations can be seen on the x-axis and the percentage portion of that relation on the y-axis. The relations are sorted according to their sizes, decreasing as moving along the x-axis. Please note, not all of the semantic relations are labeled on the x-axis even though all of the semantic relations have been plotted.

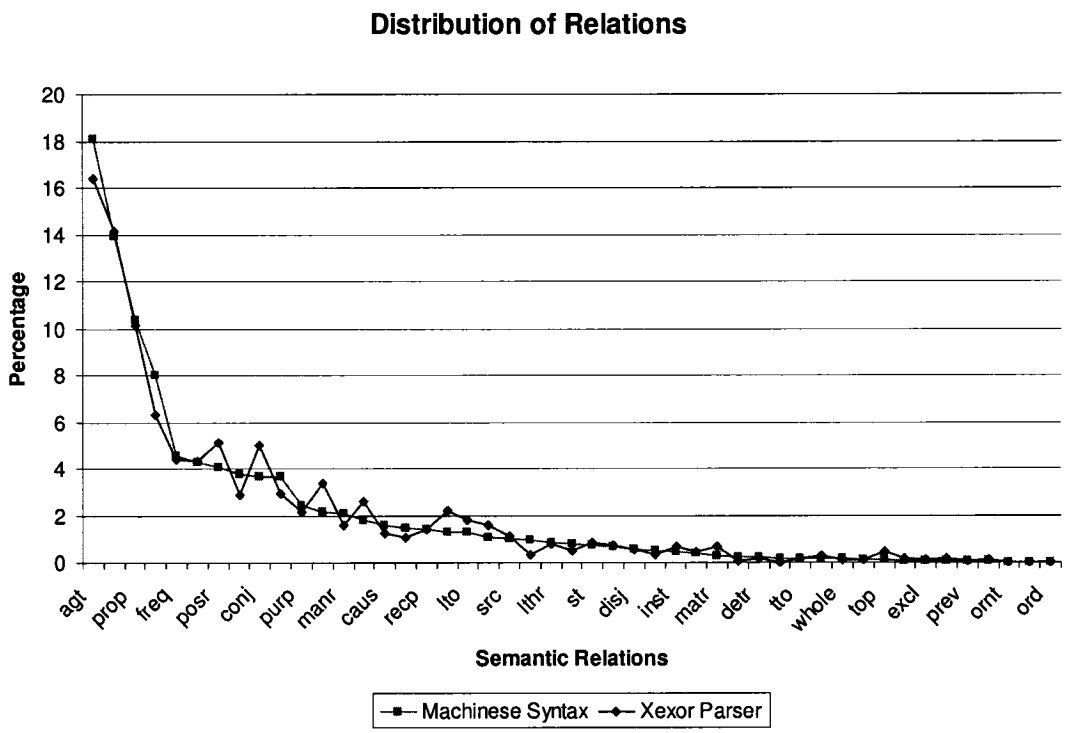


Figure 4.9 Distribution of Semantic Relations in the text

It is interesting to note that the distributions of the two datasets of semantic relations is quite similar; this similarity between the parsers' relation distributions is not surprising as the data source for both parsers is the same text. The two datasets had 61% pairs in common. The common pairs had the same relation assigned in all of the cases for the two

datasets. Irrespective of this similarity the actual assignment was done as two separate processes. A list of percentages for the frequency counts for both of the parsers can be seen in Appendix C.

4.6 Evaluating the Quality of the Dataset

In order to evaluate the data annotated by the author, a Kappa co-efficient was calculated between the assignment process done by the author and an assignment process completed by an external judge.

The external judge comprised of four graduate students, one post doctorate, one research associate and one professor. All of them have high fluency in English and are working in the field of Natural Language Processing. Each person was assigned 15 sentences to tag manually. They were instructed to determine the constituent pairs and assign a semantic relation from a predetermined list of relations (Appendix A) provided to them by the author. In total, 105 sentences were tagged and 771 pairs were assigned relations.

The objective of this evaluation task was to obtain a qualitative measure of the agreement in the annotating task between the author and several other people who are well-versed in the field. The outcome of this task offers a glimpse into how the different perspectives in assigning relations would reflect on the final data created by the author. This is a very important measure as the task of assigning relations has no set standards.

Table 4.4 shows the statistics relating to the evaluation process. The number of semantic relations extracted from the 105 sentences selected for Xerox parser and Machinese Syntax are 875 and 798 respectively. The total number of semantic relations assigned by all of the judges collectively was 771. As the external judges were allowed to pick the pairs themselves and were not given the output obtained from the parser's dependency relations, these three lists (the two lists from the parsers and one from the external judge)

of annotated relations did not fully overlap. The overlapping common set of pairs between the parser and the external judge is also shown in Table 4.4.

Source	**Number of Pairs**
Xerox	875
Machinese Syntax	798
External Annotator	771
Common set of relations with Machinese Syntax	447
Common set with XIP	514

Table 4.4 Table of pair counts given by parser and the external judge.

The relatively large difference in the number of relations obtained from the parser and the external judge brings about an interesting point for further research; testing the parser's ability to pick relations against the ability of people to manually pick them. The author uses the help of a parser to extract the pairs which are then assigned semantic relations semi-automatically using HAIKU. The external judges were asked to pick the relations completely manually. This study could help in providing further insight into the quality of pairs extracted from the parsers.

There is no circumventing the dependency of using a parser to extract pairs for the assignment of semantic relations unless the whole process is carried out manually. As, at this point in the work it is not a feasible option, a further evaluation is conducted using the common set. A snapshot in the phase of evaluation of the data set is shown in Table 4.5.

Sentence: But during all that time it was developing certain habits that it might survive more easily upon the inhospitable earth.			
Head	**Modifier**	**Rel_{oth}**	**Rel_{Auth}**
It	Developing	Agt	Agt
During	Time	Tthr	Tthr
Developing	Habits	Obj	Obj
Habits	Certain	Prop	Prop
Survive	Easily	Manr	Manr
Survive	Earth	Lat	Lat
Earth	Inhospitable	Prop	Prop
It	Survive	-	Agt

Table 4.5 Snapshot of the Evaluation data

The Rel_{oth} are the relations assigned by the external human judge and Rel_{Auth} are the relations assigned by the author.

The Kappa calculated for the two parsers as compared with the manual annotation with the seven judges is shown in Table 4.6. The total amount of common pairs represents around 10% of the data and an average Kappa over this set should provide us with an accurate measure of how the parser assignments matched up with the external human assignments.

The external judges who annotated the semantic relations have similar background in the subject matter. Even with their differences, they represent a common group who have a similar understanding for the task tackled here. In addition to this, the fact that the task was to annotate the output with one of the 47 possible options can allow for the consideration of the seven judges as one combined judge. The Kappa observed by assuming the seven judges as one combined judge is also shown in Table 4.6.

Annotator	**Kappa (Machinese Syntax)**	**Kappa (Xerox Incremental Parser)**
Ann1	0.83	0.87
Ann2	0.72	0.75
Ann3	0.74	0.78
Ann4	0.80	0.77
Ann5	0.60	0.59
Ann6	0.85	0.80
Ann7	0.66	0.60
Average Kappa	**0.74**	**0.73**
Combined Judge	**0.77**	**0.76**

Table 4.6 Kappa coefficient calculated against human judges

The average Kappa for both parsers is around 0.74 and Kappa for the data obtained from seven different judges as one combined judge is little higher. These values signify high agreement between the annotator and the human experts. The level of agreement observed confirms the quality of annotations, and is an important result as this is a hard task and open to different viewpoints.

Chapter 5: Experiments and Results

This chapter presents the different features and the machine learning experiments used to evaluate the classification performance of the semantic relation assignment. The information extracted, from different sources, is converted into a feature vector and then input into the machine learner. The feature vector consists of a set of binary or bounded real values. The testing was done using two Machine learning software systems: Decision tree (C5.0) and SVM - LibSVM. In addition to this, the decision tree implementation in Weka (j48) has also been used to get a detailed insight into the optimal classifier selected in this work.

5.1 Note on Baseline, Measure and Machine Learning Implementations

As there is no previous work to compare with, there is a lack of adequate baseline to compare these results. Comparison with other systems or even with the performance of noun modifier relation assignments is not a credible option. The objective of this task is to study the feasibility of achieving automation and thus will result in constructing a realistic baseline. This leaves the comparison of the work herein with the most naïve baseline possible: the random probability of guessing a class while assuming uniform distribution or as in the case of imbalance the size of the biggest class.

Type	Baseline
Random guessing	2.1%
Biggest class	17%

Table 5.1 Naïve Baselines

The better option of the two possible baselines, as given in table 5.1, is the size of the biggest class (relation – Agent). *Agent* has a size of 18% in the Machinese Syntax dataset

and 16% in Xerox data set. For all of the following experiments the baseline used is approximately 17%.

All machine learning tests were run using a ten-fold cross-validation method. The choice of cross-validation versus exclusively training and testing on separate sets of data is dictated by the lack of adequate quantity of data. The use of cross-validation with multi-class problem leaves us with accuracy as output. All of the machine learning system implementations used here give only the accuracy as the output measure.

The reason for picking two different implementations of decision trees resulted from the fact that for some of the experiments, running Weka was not possible as the machine used lacked adequate resources. This led to the use of the c5.0 implementation of Decision Trees. The shortcoming of using SVM and C5.0 was that neither outputs a confusion matrix. This lack of confusion matrices results in absence of detail about individual class performance. In order to get an idea of individual classes involved, the overall best classifier was rerun using Weka implementation. The Weka classifier performs a little lower than both SVM and C5.0 but was used regardless of this fact to get insight into individual relations.

5.2 The Syntactic Features

The literature reveals that the syntactic features have been primarily used in this classification task [Gildea and Jurafsky 2002, Surdeanu *et al.* 2003, Thompson *et al.* 2003, Pradhan *et al.* 2005]. In agreement with this finding, all of the syntactic information given by the two parsers is used in these experiments. In addition to this, the dependency tags output by the parser were also used as features. The performance purely based on the information obtained from the parser will help us decide between the two parsers. The next two sections contain the information on feature vectors generated from the output of the two parsers and the corresponding results obtained by the machine learning algorithms.

5.2.1 Machinese Syntax-based Features

The feature vector derived from the information obtained from the parses of Machinese Syntax parser is presented in Table 5.2:

Variable	Explanation and values
POS	Part of speech tag (Noun, verb, Adverb, Adjective, Pronoun and preposition)
Case	The case can be Accusative, Nominative, Genitive
Number	Number can be Singular, Singular first person, Singular third person, Plural, Plural first person, Plural third person
Tense	Tense can be past or present.
Person	Singular first or singular third person.
Adjective –Comparison	Absolute, Comparative and Superlative
Wh-	Pronouns and Adverbs beginning with Wh and 'how'
Pronoun type	Values: Personal, Demonstrative and Reciprocal
Adverb features	Other adverb features defined by the parser. E.g. existential there.
Verb Features	Other verb features defined by the parser output. E.g. Noun-verb combination.
Verb type	Modal Auxiliary, Infinitive, Imperative
Dependency tag	Dependency tag assigned by the parser

Table 5.2 Feature derived from Machinese Syntax

5.2.2 Xerox Incremental Parser-based Features

The feature vector for Xerox parser contains the information as presented in Table 5.3:

Variable	Explanation and values
POS	Part of speech tag (noun, verb, auxiliary, adjective, noun-adjective, preposition, adverb, conjunction, numeral, digit, determiner, pronoun, quantifier, possessive particle, measure unit, interjection, the infinitive "to")
Pronouns	Relative, possessive, reflexive and personal pronouns.
Number	Number can be Singular, Plural
Tense	Past participle, past tense, progressive and infinitive.
Person	First, second or third person.
Coordination	Coordination and subordination
Wh-	Pronouns and Adverbs beginning with Wh and 'how'
Proper Nouns	Acronyms and Proper Nouns
Dependency tag	Dependency tag assigned by the parser

Table 5.3 Feature derived from Xerox parser

In addition to the features extracted from the parsers, two additional features were added to the vector:

Phrase Type:

The Phrase type feature records whether the unit in the pair is a part of a noun phrase or a verb phrase in the sentence. This information is obtained by shallow parsing the text with ltchunk[7]. Ltchunk marks the noun and verb phrases in the text. Using simple pattern-matching algorithm the values were obtained for the head and modifier units in the relation pair database.

The feature vector length for this is two; one each for the head and the modifier. The binary values are used to mark which type of phrase is present (Table 5.4).

[7] http://www.ltg.ed.ac.uk/software/chunk/index.html

Phrase Type	Value
Verb Phrase	0
Noun Phrase	1

Table 5.4 Values for the phrase type attribute

Keyword:

The keyword feature is generated from the information derived from the TANKA's marker dictionary. This dictionary is present as a static source in the HAIKU system. The dictionary consists of semantic cues (prepositional and adverbial senses) which point to a specific relation. The feature vector length for this feature is 47, in other words equal to the size of the relation set; each semantic relation has a preset position in this vector. The corresponding value of the bit in the vector is set to 1 when that relation is observed based on the elements of the marker dictionary. In this way the name of the class is used as a feature in this category, and the Marker Dictionary provides the cues as to which of the semantic relations the corresponding keyword may signify.

A table with number of features in each of the feature sets used in the following experiments can be seen in Appendix G.

Table 5.5 presents the results obtained for the specific tests for the data from Machinese Syntax parser. All of the tests were run using ten-fold cross-validation.

Test	**Accuracy(in percent)**	
	Decision trees	SVM
MsSyn1: Parser, Keyword and Phrase type	62.80	62.14
MsSyn2: Excluding Keyword feature	58.86	57.59
MsSyn3: Excluding the phrase type	62.58	61.88

MsSyn4: Excluding the phrase type and Keyword	58.38	57.49

Table 5.5: ML Accuracy obtained with various tests for Machinese Syntax

The following table (Table 5.6) presents the results obtained for the specific tests for the data from Xerox parser. As before all of the tests were run using ten-fold cross-validation.

Test	**Accuracy(in percent)**	
	Decision trees	SVM
XipSyn1: Parser, Keyword and phrase Type	52.68	50.17
XipSyn2: Excluding Keyword feature	49.60	47.29
XipSyn3: Excluding the phrase features	52.51	49.72
XipSyn4: Excluding the phrase features and Keyword	49.42	47.09

Table 5.6: ML Accuracy obtained with various tests for Xerox Parser

The Machinese Syntax parser dataset outperforms the Xerox parser dataset when using the syntactic features output by the parser. The difference in accuracy is almost ten percent. This result is indicative of the fact that the Machinese Syntax parser outputs a richer detail which translates into a better accuracy in the learning process. The other fact noticed here is that the Keyword feature gives a good boost whereas the phrase type feature also increases the accuracy but not by much. More proof is needed to fully ascertain that the difference in accuracy is solely because of the feature set. The next section describes experiments with the datasets using features other than the ones obtained from the parser's output. The results from these are able to confirm if the difference in accuracy is solely based on the parser-derived feature set.

5.3 Statistical Features using WordNet, LDOCE and Roget's

In the next set of sections, LDOCE, Roget's and WordNet are mined for any learning that can be achieved using these resources for the purpose of classifying the set of semantic relations. The focus of this thesis is limited to finding out if any of the semantic resources analyzed here serve as a useful source of features for the task of assigning semantic relations automatically.

5.3.1 Longman Dictionary of Contemporary English (LDOCE)

Longman dictionary is an excellent source of information. The information extracted from LDOCE for each unit in the relation pair includes:

1) Part of Speech (POS)
    - This is recorded solely for the use of classification process; as it is this information is available prior to the mining of LDOCE.
2) Subject Field Code
    - LDOCE encodes all the entries with one of 236 subject codes such as law, technology, arts, medicine, etc.
3) Grammar pattern
    - LDOCE contains grammar pattern for all of the senses of 500 of the most frequent verbs in English.
4) Activator code
    - The activator code represents the semantic categorization of the entries in LDOCE. Around 20,000 entries are represented by one of the 1,052 activator codes. These summarize the similar concepts under one heading such as greedy, happy, healthy, etc.
5) Definition field is extracted in three different formats:
    - The most common definition matched with POS; LDOCE formats the output in such a way that the most common definition is the first one provided. In the experiments performed here, by default, the most

common definition is selected without performing any sense disambiguation.

- All definitions matched with POS.
- Using no definitions.

6) The pair itself

The part that requires some experimentation in LDOCE is the definition field. The dictionary provides more than one definition for most of the cases and this requires a deeper look. The choice of using one definition versus all definitions versus not using this field at all is important as the statistical synthesis is sensitive to providing more but irrelevant data. Figure 5.1 shows the information obtained from LDOCE.

```
WORD1:--------------- rock

<POS>n</POS>
<POINTER>STONE</POINTER>
<GRAM>U</GRAM>
<FIELD>HEG</FIELD>
<DEF>stone, or a type of stone that forms part of the Earth's surface</DEF>
<DEF>a piece of stone, especially a large one</DEF>
<DEF>a line of rock under or next to the sea</DEF>
<DEF>a type of popular modern music with a strong loud beat, played using</DEF>

WORD2:--------------- stand

<POS>v</POS>
<GRAM>I</GRAM>
<ACTIV>STAND</ACTIV>
<DEF>to support yourself on your feet in an upright position</DEF>
<DEF>to take a particular position or do something in particular while standing</DEF>
<DEF>to rise to an upright position or to make someone do this</DEF>
<DEF>to stay upright on a base or on an object, or to put something there</DEF>
```

Figure 5.1 LDOCE's output

This information is converted to a feature vector with the following format:

[HeadWord, ModifierWord, POS, Subj_Code, Grammar_pattern, Activator_code, Definition]

The first six fields in the feature vector are extracted directly from the dictionary's output. The definition field is represented as a bag-of-words. A stop list (Appendix F) was used while creating this bag-of-words. An example of this process is shown next:

For example, the definition of rock is

<DEF>stone, or a type of stone that forms part of the Earth's surface</DEF>

This is represented as a list of words: *[Stone, type, stone, forms, part, Earth, surface]*

In bag-of-words approach each word becomes a feature. The bag-of-words input is analyzed by the machine learner and the best features are automatically selected. In the case of multiple definitions, the list is concatenated end to end as the order of the features is not important.

The percentage of Heads and Modifiers found in the dictionary are shown in the table 5.7. This gives us the theoretical upper limit on the accuracy that can be achieved using LDOCE.

Parser	Head	Modifier
Machinese Syntax	93.91%	90.26%
Xerox Incremental	89.15%	85.11%

Table 5.7 Coverage of Heads and Modifiers observed in LDOCE

Three different sets are constructed by varying the definitions field. The next table (Table 5.8) shows the results as achieved via LDOCE.

	Machinese Syntax		Xerox	
	SVM	Decision Trees	SVM	Decision Trees
All Definition	56.18	56.00	50.16	51.39
Most Common Definition	56.20	58.60	52.34	52.40
No Definition	53.12	54.66	42.46	44.54

Table 5.8 Accuracy obtained via LDOCE

The baseline here is the same – 17%, as established at the start of this chapter. The results are higher than the naïve baseline but lower than the accuracy achieved using the parser-based features. The results clearly show that using the most common definition in this case gives us the highest accuracy. The difference is not very large in bare accuracy numbers, but when considering the fact that using all definitions increases the size of the training file by an enormous factor, thus making the learning process computationally expensive, this result is an important one.

The other thing to notice is the performance difference between Machinese Syntax and Xerox parser is not as big as it was using syntactic features. This confirms the reasons for higher accuracy observed with syntactic features as the detail level of the output of the parser and hence making Machinese Syntax a better choice for this task.

5.3.2 Roget's Thesaurus

The information extracted from Roget's is in the shape of a path between the Head and Modifier in the thesaurus. This path carries the semantic knowledge on how the two words are connected in the thesaurus. These paths, described in chapter 3, are sorted according to a distance metric. Any path at the nth position is guaranteed to be smaller or equal in length to the path at the $(n+1)^{th}$ position. As seen in the example presented next, the total number of paths generated is quite high and therefore the first step is to

determine the optimal number of paths that will give the best accuracy. Including all of the paths in the learning process is not a cost effective option if there is no gain in the accuracy obtained from the paths. This reduction in the number of paths raises the issue of which paths to include and which paths to drop in addition to the quantity to keep. In order to find an answer to these questions different experiments using only SVM are run. The paths are selected in two formats: The first using the distance metric of API, selecting n sequential paths from the top (the smallest path onwards), the second is randomly selecting n paths from the output of the API. The number n is then varied linearly to find the optimal value for the number of paths. The total paths selected at any point are converted into machine learner input by creating a list of words by appending the paths end to end with duplicate words removed. This list is then sorted and a bag-of-words feature is constructed for the word vector.

The percentage of paths found in Roget's governs the absolute maximum accuracy that can be achieved using this process. This is shown in Table 5.9.

Parser	Coverage
Machinese Syntax	86.95%
Xerox Incremental	88.01%

Table 5.9 Coverage of Pairs in Roget's

The process of extraction is explained with the help of the following example. Looking up the paths between **Rock** and **Stand** in Roget's the information obtained is:

Path between rock and stand (3978 paths in total)

Path between base 214 N. and base 214 N. (length = 2)

rock --> base <-- stand

Path between make inactive 679 VB. and make inactive 679 VB. (length = 2)

rock --> make inactive <-- stand

Path between basis 218 N. and prop 218 N. (length = 4)

rock --> basis --> N. <-- prop <-- stand

Path between basis 218 N. and stand 218 N. (length = 4)

rock --> basis --> N. <-- stand <-- stand

Path between bring to rest 266 VB. and be quiescent 266 VB. (length = 4)

rock --> bring to rest --> VB. <-- be quiescent <-- stand

Path between make inactive 679 VB. and be inactive 679 VB. (length = 4)

rock --> make inactive --> VB. <-- be inactive <-- stand

Path between tranquillize 823 VB. and be patient 823 VB. (length = 4)

rock --> tranquillize --> VB. <-- be patient <-- stand

All the paths are considered as a sequential list of words or phrases. Any re-occurring word or phrase in its entirety is omitted and considered only once. The list constructed for the above example is:

[Rock, stand, base, make inactive, basis, stand rock, bring rest, be quiescent, be inactive, tranquillize, be patient]

Again for the similar reasons as described earlier, the paths are concatenated end to end to form one vector of words. This is then converted into a numeric bag-of-words feature and fed as input to the two Machine Learner systems.

The first objective here was to find the optimal number of paths. The next two figures show the accuracy observed using SVM by gradually increasing number of paths. Any point x_i on the x-axis represents

1) the first *i* sequential paths
2) Randomly selected *i* paths out of the maximum available.

If $i <$ maximum available paths then all the paths are selected.

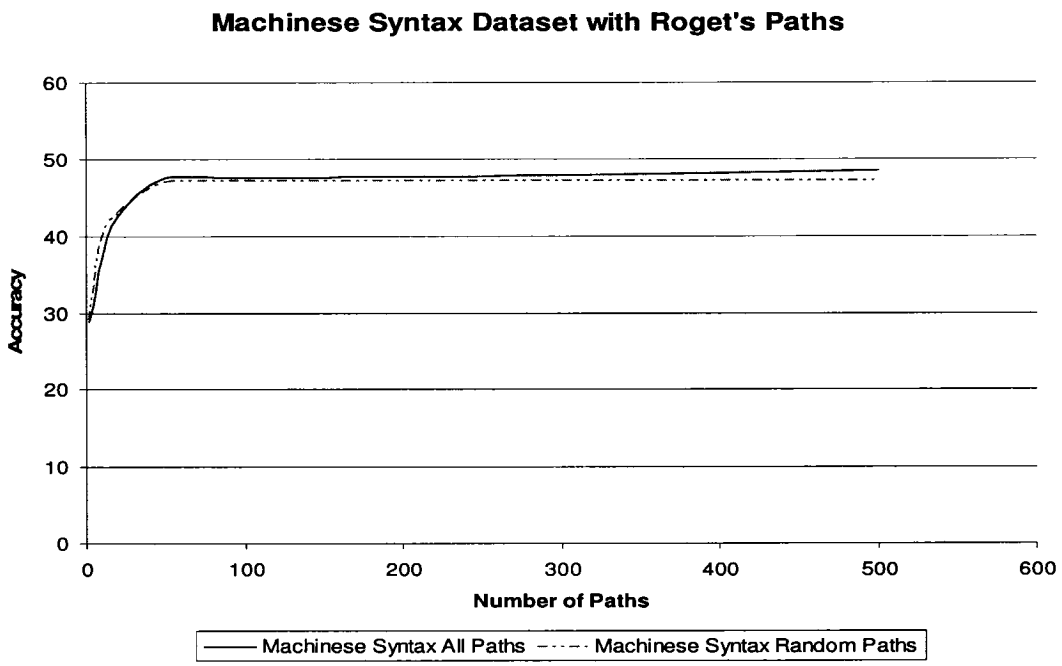


Figure 5.2 Accuracy for Machinese Syntax Dataset vs. Number of paths

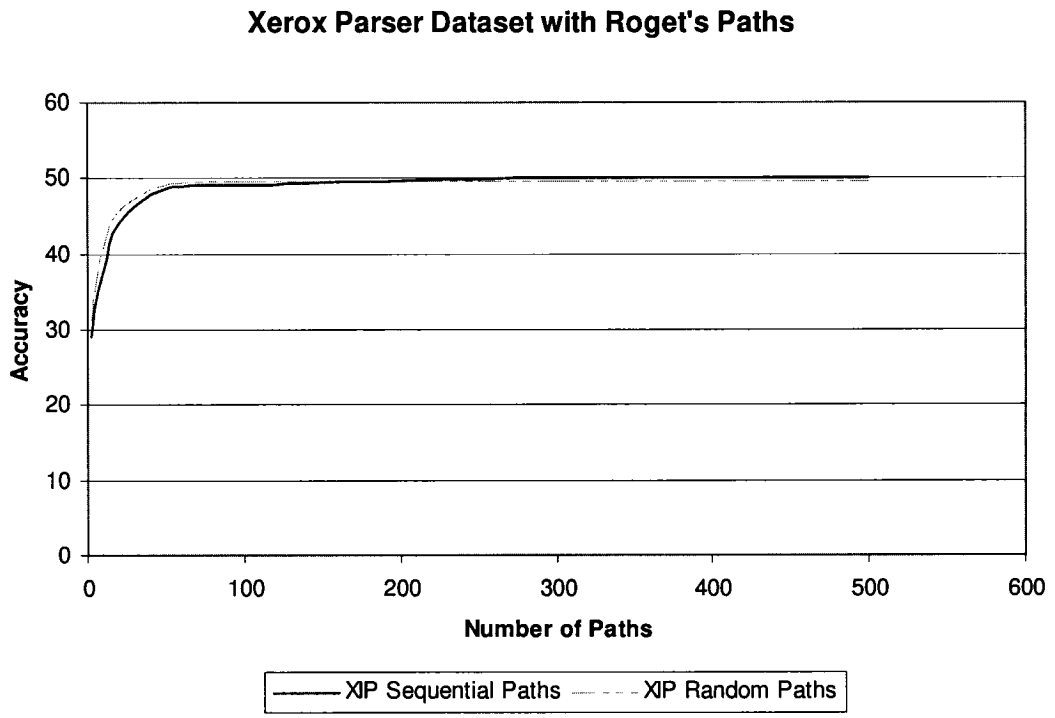


Figure 5.3 Accuracy for XIP Dataset vs. Number of paths

Experimenting on different number of paths by picking them randomly or sequentially it is observed that the curve flattens at about x= 55 for both Machinese Syntax and Xerox parser data sets.

The results obtained by using 60 paths, selected sequentially using both SVM and DT are shown in the following table (Table 5.10).

	SVM	Decision trees
Machinese Syntax	47.67	46.9
XIP	49.73	52.1

Table 5.10 Accuracy with Roget's

Here the results indicate that the difference in accuracy between Xerox parser dataset and the Machinese Syntax dataset is not as big as the one observed while using only syntactic features obtained from the parsers. Also in Roget's case the Xerox parser outperforms Machinese syntax. The difference is not much in case of SVM, but is substantial with Decision Trees. The accuracy observed however is lower than LDOCE and also lower than the accuracy observed with syntactic features.

5.3.3 WordNet

The version of WordNet used in these experiments is WordNet 2.0. A newer version of WordNet was released after the start of this work but it did not offer any improvements which would affect the work in this thesis. WordNet 2.0 provides information under 37 different link types. The work here attempts a thorough scan of the information WordNet contains to achieve the classification of the relations. The information from all of the thirty seven link types was extracted (the process is explained in the next few pages with the help of an example) and out of the thirty seven link types only 10 had information for

more than 30 percent of the units in the pair dataset. The experiments were conducted with these ten link types plus four more link types picked randomly; each of these four were underrepresented in the data extracted from WordNet and hence were selected as to understand the consequence of having limited data. The coverage of various links with the relation pairs is shown in Table 5.11.

No.	Link Type	Percentage coverage of Head Word	Percentage Coverage of Modifier
1.	antonyms	20.63	31.74
2.	hypernyms	**74.37**	**83.20**
3.	hyponyms	**63.74**	**53.75**
4.	entailment	12.04	12.16
5.	similar	**74.37**	**83.20**
6.	member meronym	9.38	5.39
7.	substance meronym	3.79	4.99
8.	part meronym	11.8	11.9
9.	member holonym	5.13	3.41
10.	substance holonym	1.87	1.43
11.	part holonym	15.76	13.40
12.	meronym	18.59	14.88
13.	holonym	20.05	15.92
14.	cause	6.59	4.67
15.	participle	0	0.01
16.	also see	14.30	15.78
17.	pertainym	1.23	3.9
18.	attribute	5.27	11.28
19.	verb group	29.92	11.28
20.	derivation	**48.30**	**42.66**
21.	classification	23.91	22.69
22.	class	4.81	3.37

23.	synonyms	**74.37**	**83.20**
24.	polysemy	34.37	33.20
25.	frames	**74.37**	**83.20**
26.	noun coordinates	**67.63**	**58.50**
27.	group related senses	29.92	27.20
28.	meronym hierarchy	**67.63**	**58.50**
29.	holonym hierarchy	20.05	15.92
30.	substring	**74.31**	**83.59**
31.	overview	**74.37**	**83.20**
32.	classif-category	19.91	17.17
33.	classif-usage	4.55	5.85
34.	classif-regional	1.59	0.79
35.	class-category	3.89	2.37
36.	class-usage	0.17	0.09
37.	class-regional	0.93	1.15

Table 5.11 Coverage of relation pairs (from Machinese Syntax) in different WordNet link types

The ten link types from which the information was extracted are: *Hypernyms, Hyponyms, Similar, Synonyms, Derivation, Frames, Noun coordinates, Meronym hierarchy, Substring and Overview.* Out of these ten, *Similar* and *Synonym* had the exactly the same output for the dataset used here. As there was no difference whatsoever in the outputs, these two are considered as one in the experiments. The four other link types, randomly selected to be included in the analyses are *Meronym, Group Related Sense, Verb Group and Part holonym.*

The pre-processing of text extracted from WordNet is detailed with the help of the following example. Looking up **Rock** under the Hypernym link in WordNet gives the following seven entries:

7 senses of rock

Sense 1

rock, stone -- (a lump or mass of hard consolidated mineral matter)

Sense 2

rock, stone -- (material consisting of the aggregate of minerals)

Sense 3

Rock, John Rock -- (United States gynecologist)

Sense 4

rock – (figurative) someone who is strong and stable and dependable

Sense 5

rock candy, rock -- (hard stick bright-colored stick candy)

Sense 6

rock 'n' roll, rock'n'roll, rock-and-roll, rock and roll, rock, rock music

Sense 7

rock, careen, sway, tilt -- (pitching dangerously to one side)

The output from WordNet is converted into a bag-of-words feature vector. The same stop list (Appendix F) was used in this process. Four different types of feature vectors are created: Two based on the inclusion/omission of the example sentence; two based on the selection of all senses versus the first sense outputted by WordNet. The example is considered to be the information contained in the parenthesis of the gloss. In the case where all of the senses were selected, the words from all of the senses were appended together into one list.

The example translates to the following two feature vectors depending on whether the example part of the gloss was taken into account or not:

With the example removed

[rock, stone, John Rock, rock candy, rock 'n' roll, rock'n'roll, rock-and-roll, rock and roll, rock, rock music, careen, sway, tilt]

With the examples:

> *[rock, stone, lump, mass, hard, consolidated, mineral, matter, material, consisting, aggregate, John Rock, United, States, gynecologist, figurative, someone, strong, stable, dependable, rock candy, stick, bright-colored, candy, rock 'n' roll, rock'n'roll, rock-and-roll, rock and roll, rock, rock music, careen, sway, tilt, pitching, dangerously, one, side]*

In this manner four different types of test sets are constructed for the 13 link types. The results obtained for Machinese Syntax dataset by running ten-fold cross-validation tests using SVM are shown in the table 5.12 below. Table 5.13 shows the similar results for the Xerox parser dataset. The baseline for these results is same as noted at the start of the chapter and is equal to 17%.

	Link Type	Examples removed		With Examples	
		All Senses	One Sense	All Senses	One Sense
1.	Hypernyms	39.88	36.74	42.41	42.13
2.	Hyponyms	26.95	23.83	29.45	25.33
3.	Meronym hierarchy	31.62	31.66	28.45	28.63
4.	noun coordinates	32.42	32.12	33.72	33.74
5.	group related senses	25.17	18.08	28.11	25.98
6.	verb group	25.23	18.10	28.07	25.99
7.	Frames	38.22	38.48	41.85	42.09
8.	part holonym	18.08	18.76	20.33	20.25
9.	Meronym	18.05	18.78	19.88	19.80
10.	Synonyms	23.21	18.56	34.26	22.89
11.	Overview	31.06	32.40	35.46	35.42
12.	Derivation	18.12	18.10	28.37	27.29
13.	Substring	26.07	26.65	24.69	25.83

Table 5.12 Accuracy for Machinese Syntax

	Link Type	Examples removed		With Examples	
		All Senses	One Sense	All Senses	One Sense
1.	Hypernyms	33.63	33.69	34.22	34.32
2.	Hyponyms	26.57	21.89	27.63	25.46
3.	Meronym hierarchy	28.65	28.69	27.32	27.24
4.	noun coordinates	29.81	29.32	32.56	32.06
5.	group related senses	25.87	26.10	25.56	26.28
6.	verb group	25.87	26.04	25.53	26.28
7.	Frames	32.36	32.36	31.95	31.95
8.	part holonym	20.97	21.06	21.34	21.48
9.	Meronym	21.14	21.10	21.73	21.51
10.	Synonyms	29.95	27.87	30.89	31.75
11.	Overview	32.83	32.71	33.77	33.28
12.	Derivation	24.36	24.22	28.00	27.87
13.	Substring	26.55	26.38	26.51	26.38

Table 5.13 Accuracy for Xerox Parser

For both the Xerox parser and Machinese Syntax dataset the best results are seen for the Hypernym category. A close second is the Frames category. The results also show that using the examples with all senses gives the best accuracy. Overall the results are lower than what is observed for either LDOCE, Roget's or by using the syntactic features. The difference in accuracy observed for Hypernym and Frames is quite small and insignificant. A closer inspection of the data extracted under these two link types revealed that the information under these has a large intersection set with only very small difference. Also, hypernyms formed a superset and fully contained the Frames link type.

5.4 Comparison of the Accuracy Observed using LDOCE, Roget's and WordNet.

This section provides a review of the best results obtained from Roget's, LDOCE, WordNet and from pure syntactic features. All of the individual resources perform quite well when compared with the naïve baseline established at the beginning of this chapter.

A quick view of the results shows that the Xerox parser dataset has the best performance while using LDOCE. This is surprising as it beats the accuracy obtained when using the same dataset with syntactic features. On the other hand the results for Machinese Syntax are as expected; the syntactic features give the best accuracy.

	Accuracy(in percent)	
	Xerox Parser	Machinese Syntax
WordNet	34.32	42.41
Roget's	50.16	48.5
LDOCE	52.34	56.2
Syntactic	47.29	57.59
Baseline	16.42	18.07

Table 5.14 Comparison of accuracies

The most surprising results are from WordNet. WordNet is a "better" version of a dictionary and a thesaurus, and the results obtained here are worse than both of those individual resources. The question that arises from this situation is: Is there any other way of using WordNet which may give better results? One of the important results obtained from the experimentation with WordNet is that the most information available is found under the hypernym link type. This allows limiting any further investigation into WordNet to just the hypernyms.

5.5 WordNet Hypernyms

The bag-of-words synthesis of WordNet reveals that the best results are achieved by using the hypernym definitions. In order to achieve a better classifying performance out of WordNet, hypernyms are treated specially.

Nastase [2003b] looked at the use of Roget's and WordNet to classify the noun modifier relations which form a subsection of the list of semantic relations [Appendix A]. For a complete description of which semantic relations occur as noun modifiers see Nastase [2003]. She looks at hypernym definitions and creates a vector based on the complete hypernym output of WordNet. She leaves the generalization to be achieved via the learner. Girju [2003] used 9 noun-hierarchies in WordNet: *entity, psychological feature, abstraction, state, event, act, group, possession and phenomenon* as features to classify noun phrases to automatically recognize the causal relations. She uses the causal relations to develop a question-answering system. This thesis looks at the hypernym definitions and tries to come up with generalization rules that can help classify semantic relations. The working principle involves finding generalization in the hypernym definition for all the HEAD units and among all of the MODIFIER units of a pair in the dataset under a particular relation type. The generalization, if observed over the entire spectrum and if found sufficiently prevalent is accepted as a rule. This technique does not provide rules for all the relations but does provide concrete cues for some of the relations. This process is explained with the help of the following examples.

Consider the following two pairs:-

> [Cigarette, Burn] Relation EFFECT (Head is the result of the modifier)
> [Snow, Blindness] Relation EFFECT (Head is the result of the modifier)

Looking up the hypernym glosses for *cigarette* and *Snow,* the following common section in the glosses of the two is observed:

----------*snow ∩ cigarette*----------

=> *entity -- (that which is perceived or known or inferred to have its own distinct existence*

=> ***causal agent, cause, causal agency*** *-- (any entity that causes events to happen)*

Similarly for *Burn* and *blindness*:

----------*blindness ∩ burn*----------

=> *ill health, unhealthiness, health problem -- (a state in which you are unable to function normally and without pain)*

=> *pathological state -- (a condition that is not normal)*

=> *condition, status -- (a state at a particular time; "a condition (or state) of disrepair"; "the current status of the arms negotiations")*

=> ***state*** *-- (the way something is with respect to its main attributes; "the current state of knowledge"; "his state of health"; "in a weak financial state")*

This common section is obtained by running a generalization-mapping heuristic (Figure 5.4) among the glosses of all the Heads and Modifiers in the dataset. The final output of the heuristic is the most prevalent common section found and two lists: the first list is of the pairs whose glosses are a part of the common section, and the second list of pairs whose glosses did not have that common section in them.

- *Populate an array A with* ${}^{n}C_2$ = n!/(2!(n-2)!) *elements where n = number of Heads/Modifiers, each array element contains the common gloss section found by pair-wise comparison*
- *For each element in the array perform an intersection with the other n-1 glosses.*
  - *Record the pairs that have a common section in i-listCommon and the ones that do not have a common section in i-listNotPresent. (i is the position in array A)*

- *The final intersection results for all i-listCommon lists were manually examined to deduce a possibility of suitable generalization.*

Figure 5.4 Pattern Matching Heuristic for finding Hypernym Generalizations

In some cases a generalization is only found for either the Head or the Modifier part.

The following generalization level is achieved when an intersection operation is run over the entire set and the common sections observed for the two pairs mentioned before:

Modifier: Head:: [Causal Agent] : [State] ::
Snow : Blindness :: Cigarette Burn

Another example can be seen with the help of the following pairs:-

[weapon, iron] Relation : MATERIAL (Head is made of modifier)
[mask, clay] Relation : MATERIAL
[bank,mud], Relation : MATERIAL
[sword, metal] Relation : MATERIAL

Generalizations found for the MATERIAL relation using WordNet give the following:

Modifier: Head:: [Substance, Matter] : [Artifact] ::
Iron: Weapon :: Clay : Mask :: Mud : Bank :: Metal : Sword

An important observation here is that *bank* in the example [bank, mud] is generalized as an *Artifact* using WordNet and works with the classification method defined here, even though the sense of *bank* that is used here is not the right sense according to the text. As no word-sense resolution was performed in this process, this was a predictable shortcoming. This has both negative and positive effects and is left as future work.

For semantic relations LOCATION AT, LOCATED the presence of the patterns *"Location, Region, Geographical formation"* is observed in the Modifier fields.

As mentioned above, this generalization technique gives strong cues but only for a subset of the relations. This process was done once for all the instances and the rules thus generated are shown in Table 5.15.

Relation	Cue in the WordNet gloss
Cause	Head word: "causal agent, cause, causal agency" "phenomenon -- any state or process known through" "act, human action, human activity -" Modifier: "state -- the way something is with" "condition, status -- a state at a particular time"
Effect	Head word: "state -- the way something is with" "condition, status -- a state at a particular time" Modifier: "causal agent, cause, causal agency" "phenomenon -- any state or process " "act, human action, human activity -"
Purpose	Head word or Modifier: reason, ground ,a rational" "intent, intention, aim"
Frequency	"many times at" "at all times" "times
Time from	Head : "have a beginning, of a temporal event"
Time to, time at	Modifier : "have a beginning, of a temporal event"
Direction	Modifier: "direction -- the spatial relation between something"
Located, location at	Modifier: "location -- a point or extent in space" "region -- a large indefinite" "geological formation, formation"
Location from	Head: "location -- a point or extent in space" "region -- a large indefinite" "geological formation, formation"

Agent	Head: "act, human action, human activity"
Beneficiary	Modifier: "act, human action, human activity"
Stative	Head: "substance, matter " "living thing, animate thing" Modifier: "state -- the way something is with" "act, move -- perform an action" "position, place "
Object property	Head: "thing -- a separate and self" Modifier: "change -- undergo a change"
Container	Head: "container -- any object that can be used to"
Content	Modifier: "container -- any object that can be used to"
Material	Head : "artifact, artifact" Modifier: "substance, matter --"
Measure	Head word or Modifier: "measure, quantity, amount -- how much there"
Manner	Head word or Modifier: "in what way or manner or by what means"

Table 5.15 Cues found using Hypernym Definitions in WordNet

5.6 Comparing the Performance of Different WordNet-Based Feature Sets

The hypernym-based feature developed in the last section has one shortcoming as noted earlier; it is not available for all relations. This shortcoming prevents it from being tested directly against the bag-of-words approach. One of the solutions that has been adopted here is to compare the performance by combining the hypernym-based feature with the syntactic features.

The Table 5.16 below gives the comparison of the accuracy for the different WordNet-based features.

	Machinese Syntax		Xerox	
	SVM	Decision Trees	SVM	Decision Trees
Bag-of-words	42.41	29.34	34.22	25.56
Bag-of-words+ Syntactic	61.65	54.10	46.52	42.80
Hypernym + Syntactic	**64.46**	**65.36**	53.27	56.27
Syntactic	62.14	62.80	50.17	52.68
Baseline	16.42	16.42	18.07	18.07

Table 5.16 Accuracy for different WordNet-based features

This table shows the difference in accuracy achieved by employing the hypernym feature. Parser-based features combined with the hypernym derived feature outperform the combination of parser features with bag-of-words feature. The best results are achieved for Machinese Syntax dataset. The Machinese Syntax-based dataset combined with the hypernym feature is better than the similar dataset from Xerox Parser by more than 10% for SVM and 9% with DT. The hypernym feature increases the accuracy observed here by 2.32% with SVM and 2.56% for DT. The hypernym-based feature not only performs better but it is also computationally quite efficient as compared to the bag-of-words. To give an idea of the computation involved, it takes more than 2 hours to run a ten-fold cross-validation using SVMs on the bag-of-words based features versus only a few minutes for the syntactic and the condensed hypernym-based feature.

The next section extends this approach of combining the syntactic features with LDOCE and Roget's and presents a comparison of the results achieved.

5.7 A Hybrid Classifier

This section reports the results obtained from a hybrid classifier. The statistical features are combined with the syntactic features to form a hybrid solution. The results are presented in table 5.17 below:

	Accuracy(in percent) using SVM	
	Xerox Parser	Machinese Syntax
Syntactic + Roget's	56.12	65.73
Syntactic + LDOCE	60.37	65.71
Syntactic + WordNet	46.52	61.65
All	58.33	64.41
Syntactic + WordNet Hypernym	53.27	64.46
Baseline	16.42	18.07

Table 5.17 Accuracy with a Hybrid classifier

The best performance is seen with a combination of Machinese Syntax-based features and Roget's or with Machinese Syntax and LDOCE. The condensed feature set consisting of syntactic features and the hypernym-based feature comes as a close second in performance. The combination of Xerox parser derived-features with bag-of-words features gives varied results. However the Machinese Syntax-based dataset gives overall stable results and is better than Xerox in all cases.

5.8 The Optimal Classifier

The first choice is to select the best performing classifier, however if optimality is brought into the picture, some loss in pure accuracy is acceptable to have a computationally inexpensive classifier. The difference in the training files between the bag-of-words classifier derived from WordNet and the condensed features based classifier is 9:1 (456 MB file versus 52 MB file).

Table 5.18 shows the results achieved using feature vector which consists of condensed hypernym, phrase type, keyword and parser features for Machinese Syntax and Xerox Incremental parser-based datasets.

Test	Accuracy	
	Decision trees	SVM
MsSem1: Parser and Keyword, Phrase type and Hypernym	**65.35**	**64.46**
XIPSem1: Parser and Keyword, Phrase type and Hypernym	**56.27**	**53.74**

Table 5.18 ML accuracy with the combined test of hypernyms and parser features.

The next two tables (Table 5.19 and Table 5.20) list all of the relations with non-zero accuracy. The baseline is calculated according to the distribution of the relations and represents the size of that class in the dataset. The first table gives the results for Machinese Syntax as obtained by running ten-fold cross-validation using J48 implementation in Weka.

Class	Precision	Recall	F Measure	Baseline
Caus	22.4	21.8	22.1	1.55
Effe	26.5	12.3	16.8	1.45
Purp	33.0	29.8	31.3	2.41
Entl	9.50	5.0	6.6	0.79
Freq	57.7	76.5	65.8	4.59
Prec	16.0	31.9	21.3	0.93
Tthr	47.2	51.5	49.3	0.65
Dir	41.7	20.0	27.0	0.49
Lfrm	72.7	59.3	65.3	1.07
lto	44.7	53.1	48.6	1.27
lthr	45.0	40.9	42.9	0.87
lat	45.6	62.6	52.8	4.27
Ltd	9.1	1.5	2.6	1.29
Conj	68.6	68.6	68.6	3.69

Class	Precision	Recall	F Measure	Baseline
Agt	90.5	96.6	93.4	18.08
ben	25.0	4.8	8.0	0.41
Prop	53.2	75.7	62.5	10.36
Poss	91.4	88.2	89.8	4.07
Inst	13.0	13.0	13.0	0.45
obj	67.8	80.1	73.5	13.94
Recp	50.8	44.3	47.3	1.39
Part	43.6	41.1	42.3	2.13
Src	52.9	18.0	26.9	0.99
Top	40.0	40.0	40.0	0.099
Manr	45.0	48.1	46.5	2.07
matr	66.7	26.7	38.1	0.299
Meas	70.6	40.3	51.3	7.99
equa	56.5	38.5	45.8	1.81
Type	43.0	27.4	33.4	3.79

Table 5.19 Individual classes for Machinese Syntax (Using Decision trees in WEKA)

The next table (Table 5.20) gives the results for Xerox Incremental parser under similar settings.

Class	Precision	Recall	F Measure	Baseline
Effe	36.4	15.4	21.6	1.06
Purp	22.0	8.70	12.4	2.12
Freq	77.9	78.2	78.1	4.40
Tat	51.4	51.0	51.2	2.91
Tthr	32.7	43.2	37.2	0.75
Dir	9.10	5.60	6.90	0.36

Class	Precision	Recall	F Measure	Baseline
Lfrm	50.0	35.5	41.5	1.55
Lto	63.8	34.1	44.4	1.79
Lthr	37.5	15.8	22.2	0.77
Lat	38.6	38.2	38.4	4.32
Ltd	15.2	8.90	11.2	2.28
Conj	98.4	98.8	98.6	5.01
Disj	100	100	100	0.55
Agt	71.9	94.3	81.6	16.42
Prop	54.8	69.1	61.1	10.16
Poss	68.7	77.4	72.8	5.14
Inst	100	3.10	6.10	0.65
Obj	44.1	70.8	54.3	14.11
Recp	12.5	1.40	2.60	1.42
Part	34.4	32.1	33.2	3.36
Src	38.9	12.5	18.9	1.14
Top	33.3	13.0	18.8	0.46
Manr	46.7	9.20	15.4	1.55
Matr	6.70	2.90	4.10	0.69
Meas	60.2	43.9	50.7	6.32
Equa	19.1	13.3	15.7	2.61
Type	43.6	31.2	36.4	2.87

Table 5.20 Individual classes for Xerox Incremental Parser (Weka)

Some of the classes with high level of accuracy are: *Frequency, Time at, cause, Conjunction, disjunction, agent, object, manner and measure.* The individual relations are discussed in detail in the next chapter. Appendix H presents a small section of the decision tree generated by Weka using the Machinese Syntax dataset.

5.9 Testing with Six Super-Classes

Learning on the 47 relations using a small dataset does not give a true picture of the potential that can be achieved. One way to analyze this potential is to have more annotated data, which is quite a lengthy process. The other approach to analyze a theoretical upper limit for the learning task is to combine the relations into six main super-classes and run tests on them. This approach was first taken for this relation set by Turney [2003]. A similar approach was also undertaken by Gildea and Jurafsky [2002] where they reported results for 18 condensed thematic roles from the original set of 67 Frame types. Following the same approach, the 47 relations are reduced to six classes, namely: Causality, Temporality, Spatiality, Conjunctive, Participant and Quality. The test was only done using the features derived from the syntactic and the hypernym features. Table 5.21 gives the combined accuracy observed using Decision Trees and SVM

Test	**Accuracy**	
	Decision trees	SVM
6clsMsSem1	**73.47**	**73.34**

Table 5.21 Accuracy with Machinese Syntax for 6 super classes

These results indicate that the maximum accuracy that can be achieved with the full set of relations with the current feature set is 74%. The next table (Table 5.22) shows the performance of individual classes. These results are obtained using ten-fold cross-validation in Weka J48 classifier.

The best accuracy is achieved for the *Participant* class which is also the biggest class in the data set. The two classes at the lower end are *Causality* and *Quality*. These results indicate *Quality* is the hardest class to learn as it is nearly three times the size of causality and still has the same F-measure. A complete discussion on the performance of individual relations follows in the next chapter.

Class	Precision	Recall	F Measure	Class-size/Baseline
Causality	50.7	55.7	53.1	6.63
Temporality	70.9	64.2	67.4	10.48
Spatiality	64.7	59.0	61.7	9.30
Conjunctive	80.6	69.8	74.8	4.29
Participant	79.1	88.8	83.7	53.13
Quality	65.0	45.7	53.6	16.14

Table 5.22 Accuracy for the 6 super classes

Chapter 6: Results

This section provides a review of the individual performance of the relations for the Machinese Syntax parser dataset. The results are obtained by using the syntactic features from Machinese Syntax parser and the WordNet hypernym-based feature with the decision tree implementation in Weka (J48). This discussion helps clearly mark which relations are easily classifiable and which ones are difficult to learn. The results are discussed in groups of six main classes.

A) The Causality class

	CAUSALITY			
Relation	**Precision**	**Recall**	**F-measure**	**Class size/Baseline**
Caus	22.4	21.8	22.1	1.55
Effe	26.5	12.3	16.8	1.45
Purp	33.0	29.8	31.3	2.41
Entl	9.50	5.0	6.6	0.79
Enbl	*0*	*0*	*0*	*0.11*
Detr	*0*	*0*	*0*	*0.23*
Prev	*0*	*0*	*0*	*0.039*

Table 6.1 Accuracy of Individual Relations in Causality class

The relations under this heading are generally challenging to pick and this can be seen in the results as well. The best score is achieved by the relation *purpose* which is also the biggest relation in terms of size in this class. The three relations with zero F-measure are *enablement, detraction and prevention.* The respective class sizes/baselines for these relations are also quite small. One of the reasons for zero F-measure can be attributed to their small size. The other reason this may happen is that there are usually no direct markers that would help in picking these relations.

For example in the following sentence

> *Who was he, himself, a strange little creature surrounded on all sides by [death and sickness]$_H$ and yet [happy and full of laughter]$_M$.*

Here H opposes M but does not prevent it and hence marks the detraction relation. There is not enough semantic or syntactic information which can help point out the relation.

B) The Temporality class

	TEMPORALITY			
Relation	**Prec.**	**Recall**	**F-meas**	**Class Size/Baseline**
Cooc	*0*	*0*	*0*	*0.25*
Freq	57.7	76.5	65.8	4.59
Prec	16.0	31.9	21.3	0.93
Tat	64.8	63.4	65.1	3.65
Tfr	*0*	*0*	*0*	*0.199*
Tthr	47.2	51.5	49.3	0.65
tto	*0*	*0*	*0*	*0.17*

Table 6.2 Accuracy of Individual Relations in Temporality class

The set of relations in the temporality class are very closely packed. The results quite clearly point at the fact that the under-represented relations take a hit in performance due to this. The lack of adequate examples for *co-occurrence*, *time from* and *time to* make these relations hard to classify and this also affects the performance of other relations. The other thing to note is that there are relations that are efficiently classified irrespective of a small class size. This is due to the fact that the Head or the Modifier in the pair usually carries some marker which indicates time. *Frequency* and *Time at* perform well in this class. *Time from* and *time to* are very close to *time through* and *time at* and hence need better mechanisms to differentiate them. Also, to further improve performance in this category additional examples are needed. As an immediate solution some of the relations can be clubbed together and if incorporated into the heuristics, the system can

give a refined choice to the user in a form of a smaller list instead of presenting a complete list of relations during the semi-automatic assignment process.

C) The Spatiality class

	SPATIALITY			
Relation	**Precision**	**Recall**	**F-measure**	**Class Size/Baseline**
Dir	41.7	20.0	27.0	0.49
Lfrm	72.7	59.3	65.3	1.07
lto	44.7	53.1	48.6	1.27
lthr	45.0	40.9	42.9	0.87
lat	45.6	62.6	52.8	4.27
Ltd	*9.1*	*1.5*	*2.6*	*1.29*
Orn	*0*	*0*	*0*	*0*

Table 6.3 Accuracy of Individual Relations in Spatiality class

The relations in this class are similar to the temporality class in their property of being closely related. Overall the relations perform well in this class, but a higher performance can be achieved in this class and the temporality class by determining some distinguishing features amongst the tightly related relations. The relation *Orientation* has no instances in the text and hence the zero F-measure. The other relation that performs badly in this class is *located.* This relation is close to *location at* and *location to.* This may be the cause of the low performance.

For example, in the sentence

... see your less familiar [cousins]$_M$ [behind the bars]$_H$ of the zoological garden.

Here H is *located* M.

... they are able to [live]$_H$ with equal ease [on the land]$_M$ and in the water.

In the above sentence H is at the *location at* M.

These relations are fairly complex for humans to distinguish as well, so it is not surprising that the results are at the lower end.

D) The Conjunctive class

	CONJUNCTIVE			
Relation	**Precision**	**Recall**	**F-measure**	**Class Size/Baseline**
Conjunction	68.6	68.6	68.6	3.69
Disjunction	*0*	*0*	*0*	*0.59*

Table 6.4 Accuracy of Individual Relations in Conjunctive class

The results for disjunction are very surprising but then again it has a very small presence in the data set which could account for the zero F-measure.

E) The Participant class

	PARTICIPANT			
Relation	**Precision**	**Recall**	**F-measure**	**Class Size/Baseline**
Acmp	*0*	*0*	*0*	*0.059*
Agt	90.5	96.6	93.4	18.08
ben	25.0	4.8	8.0	0.41
Excl	*0*	*0*	*0*	*0.059*
St	*0*	*0*	*0*	*0.75*
Prop	53.2	75.7	62.5	10.36
Posr	*0*	*0*	*0*	*0*
Poss	91.4	88.2	89.8	4.07
Inst	13.0	13.0	13.0	0.45
obj	67.8	80.1	73.5	13.94
Obj_prop	*0*	*0*	*0*	*0.17*
Recp	50.8	44.3	47.3	1.39
Part	43.6	41.1	42.3	2.13
Wh	*0*	*0*	*0*	*0.015*
Prod	*0*	*0*	*0*	*0.039*
Src	52.9	18.0	26.9	0.99

Table 6.5 Accuracy of Individual Relations in Participant class

The participant class is the largest class in the data set and is also the one with the highest performance. This class has two of the biggest relations *agent* and *object*, which are not only abundant in the text but also easily classifiable. Hence the high F-measure for *agent* and *object* relations is not a surprise. The other relations that perform well in this class are *property* and *possession*. The relations that perform badly in this class are again the ones which are very scarce in the data set. These relations are *accompaniment, exclusion, stative, object property, whole, product and instrument.*

E) The Quality class

	QUALITY			
Relation	**Precision**	**Recall**	**F-measure**	**Class Size/Baseline**
Cont	*0*	*0*	*0*	*0.059*
Top	40.0	40.0	40.0	0.099
Cntr	*0*	*0*	*0*	*0*
Manr	45.0	48.1	46.5	2.07
Matr	66.7	26.7	38.1	0.299
Meas	70.6	40.3	51.3	7.99
Ord	*0*	*0*	*0*	*0*
Eq	56.5	38.5	45.8	1.81
Type	43.0	27.4	33.4	3.79

Table 6.6 Accuracy of Individual Relations in Quality class

The quality class is the second biggest class. Overall the relations performance is mediocre. *Manner, measure,* and *equative* are very difficult relations to automatically classify but the overall performance is much higher than the baseline. The relations that do not perform well are *container, content, and order.* This again can be attributed to the small presence of these relations in the data set.

In a nutshell there are two main reasons why some of the relations were not easily classifiable:

1) The relations were under-represented in the data.
2) The relation was semantically too close to another relation which made it harder to classify. One relation suffers at the expense of the other.

The quantity of relations in the data set is representative of the text, so if a relation is not observed frequently than it is a rare relation. This however needs more proof and requires work using multiple texts.

There is a distinct need to have more features which are able to classify closely related relations. One realistic solution was provided by the hypernym-based feature. Both Roget's and LDOCE offer in them a vast resource which can be exploited in more ways than what has been attempted in this thesis. Further study of these semantic resources should unveil more clues which shall help in improving the accuracy further.

Chapter 7: Conclusion and Future Work

7.1 Conclusion

The work here presents the first stage in the process of achieving full automation in assigning semantic relations. The contribution of this work is manifold and provides a strong base for any future work:

1) It provides a parser replacement for the system.
2) It provides a new data set for future work.
3) It gives an insight into performance of individual relations.
4) It gives a comparison of the performance, in assigning semantic relations automatically, obtained by using syntactic features versus the features derived from semantic resources.
5) It develops a realistic baseline and proves the feasibility of the task.

The shortcomings of the previous parser DIPETT outlined in this thesis necessitated finding a suitable replacement that will allow the project to move to the next stage. This has been achieved in this thesis and a suitable replacement in Machinese Syntax has been found. The parser gives highly accurate parses and also provides a rich detail of syntactic features. The advantage of this is evident in the results that were obtained using purely syntactic features. The difference in the accuracy achieved between the two parsers tested here (XIP and Machinese Syntax) is approximately 10 percent.

The previous data set employed in the development and testing of the system was not sufficiently large enough to reasonably experiment and automatically classify all the relations. The large set of semantic relations (47) being classified in this thesis warranted the need for a bigger dataset. This thesis provides two sets of 5000 pairs annotated with semantic relations. Each of these sets of data is two and a half times larger than the previous data set. This data, although bigger and richer in its coverage of relations, is still short of what would be needed to achieve full automation. The requirement for a larger

data set is driven by the large set of relations to be classified. Nevertheless, the data set created here helps to achieve a greater understanding of the requirements of the automation process.

Another important finding is the insight into the performance of individual relations. The results clearly mark which relations are hard to classify. These results give a good ground on which further work can be based to intelligently find new features which will help to classify the harder relations. Also the results hint that having a hierarchical classifier can give better results. Clubbing the relations into 6 super classes gives a higher performance, but this may not be the most practical approach as there is a need to classify all the relations. The in-between solution is to club the closely related relations especially in the temporal and spatial classes.

This thesis also provides a review of what can be achieved from different semantic resources as opposed to sticking with only syntactic features. The results are interesting as the overall combined performance gives a considerable boost over the individual resource. The use of the hypernym-based feature shows the kind of difference that can be achieved over the bag-of-words approach. This insight suggests that a better mining of these resources should help in classifying some of the harder relations which are not distinguishable using syntactic features.

Lastly, the most important result is the proof of the feasibility of this task. This completes the main aim of this work. The thesis provides very good results for a task which is relatively hard for humans as well. The results are about twice the baseline of PropBank-based semantic role labeling. The accuracy observed here is proposed as baseline for further work in this direction. The results are encouraging and clearly provide a detailed insight into further work needed to achieve the task of fully automating the system.

7.2 Future work

The future work for this project involves using the data set and conclusions presented above to further increase the accuracy of the task and to build it into the system as heuristics.

Though the desirable final output is a fully automatic system, implementing the classifier features into the system at any stage would result in the improvement of the semi-automatic process. This has a dual advantage: one being the improved semi-automatic learning process which will enrich the system itself and the second is that the partially improved system will help in creating more annotated data at a quicker pace. The data thus created can be used to further increase the quality of automatic learning. The heuristics used in the current system are weak and implementing the features examined here would help in bolstering up the process.

The dataset created here, though adequate for the first stage, is still a bottleneck in the learning process. There is a definite need to have a larger data set. The essential requirement that needs to be met here is to have substantial examples for all the relations that are being classified. The experiments here indicate at having a minimum of 200 examples per relation. The process of creating more data is a required dedicated process which is necessary before an efficient automation can be achieved.

The other issue not addressed in this thesis is the comparison of performance across different type of texts. It will be interesting to see how it performs across different texts. Comparison of performance taking purely literary texts versus technical and semi-technical domain-specific texts should reveal details of how the relations are distributed under these different circumstances and also what types of relations are classifiable.

The other key issue that can be addressed in the future is the dependence on the output of parser. A hint at this future work was seen while evaluating the dataset created here. For this work the dependency pairs were used as output by the parser. This limits the ability

of capturing the semantic relations on the parser's ability to pick the semantic relations. A refined approach would be able to pick up relations which are not directly linked by the parser. This step would require creating the dataset manually in a way that ensures that all relations in a sentence are correctly annotated. It would be very interesting to see the impact on the performance in such a system while automatically classifying relations.

The other thing in this context is the process of rejection of any relations assigned by the parser which do not form a part of the set of relations analyzed in this system. For now this was done manually, but on the whole this process should also be a part of the bigger task of assigning semantic relations automatically, and in this case assigning a null relation.

References

1.) Banerjee, Satanjeev and Pedersen, Ted. 2003. "Extended gloss overlaps as a measure of semantic relatedness", in Proceedings of the Eighteenth International Joint Conference on Artificial Intelligence, pp. 805-810, Acapulco, Mexico.

2.) Barker, Ken. 1998. "Semi-Automatic Recognition of Semantic Relationships in English Technical texts." Ph.D. Thesis, University of Ottawa, Department of Computer Science. http://www.cs.utexas.edu/users/kbarker/thesis/

3.) Barker, Ken and Delisle, Sylvain. 1996. Experimental Validation of a Semi-Automatic Text Analyzer. Technical Report TR-96-01, Department of Computer Science, University of Ottawa.

4.) Baker, Collin F., Fillmore, Charles J., and Lowe, John B. 1998. "The Berkeley FrameNet project." In Proceedings of the COLING-ACL, pp. 86-90, Montreal, Canada.

5.) Bhartrhari. 1971. "The Vakyapadiya. Critical texts of Cantos I and II with English Translation, Summary of Ideas and Notes by K. Raghavan Pillai." Delhi: Motilal Banarasidass.

6.) Boas, H. C. 2002. "Bilingual FrameNet dictionaries for machine translation." In Proceedings of LREC 2002, pp. 1364–1371, Las Palmas, Canary Islands.

7.) Bréal, Michel. 1883. "Les lois intellectuelles du langage." L'annuaire de l'association pour l'encouragement des etudes grecques en France.

8.) Byrd, Roy J., Heidorn, George E., Chodorow, M. S. 1985. "Extracting semantic hierarchies from a large on-line dictionary." Proceedings of the 23rd Annual Meeting of the Association for Computational Linguistics, pp. 299- 304, Chicago.

9.) Carletta, J. 1996. "Assessing Agreement on Classification Tasks: The Kappa Statistic", Computational Linguistics, 22(2), 249-254.

10.) Carnap R. 1937. "The Logical Syntax of Language." London: Kegan Paul.

11.) Carreras, X. and Màrquez, L. 2005. "Introduction to the CoNLL-2005 Shared Task: Semantic Role Labeling" 9th Conference on Computational Natural Language Learning (CoNLL-2005), pp. 152-164, Ann Arbor, Michigan, US.

12.) Carreras, X. and Màrquez, L. 2004. "Introduction to the CoNLL-2004 Shared Task: Semantic Role Labeling." In Proceedings of Eighth Conference on Computational Natural Language Learning (CoNLL-2004), pp. 89-97, Boston, USA.

13.) Chanod, J.P., Ait-Mokhtar, S., Roux, C. 2004 "Xerox Incremental Parser", ongoing research Xerox Research Centre Europe.

14.) Chen, J. and Rambow, O. 2003. "Use of deep linguistics features for the recognition and labeling of semantic arguments." EMNLP-03, pp. 41 - 48 .

15.) Chih-Chung Chang and Chih-Jen Lin, "LIBSVM. 2001. A library for support vector machines", Software available at http://www.csie.ntu.edu.tw/~cjlin/libsvm

16.) Cohen, J. 1960. "A coefficient of agreement for nominal scales." Educational and Psychological Measurement 20, pp. 37–46.

17.) Cohen, W. and Singer, Y. 1999. "A simple, fast and effective rule learner." In Proc. of Annual Conference of American Association for Artificial Intelligence, pp. 335-342.

18.) Copeck, T., Delisle, S. and Szpakowicz, S. "Parsing and Case Analysis in TANKA." Proc. of COLING-92 (Nantes, France), 1992, 1008-1012.

19.) Delisle, S. 1994. "Text Processing Without a-priori Domain Knowledge: Semi - Automatic Linguistic Analysis for Incremental Knowledge Acquisition." Ph.D. Thesis, University of Ottawa, Department of Computer Science. TR-94-02.

20.) Delisle, S., Copeck, T., Szpakowicz, S. & Barker, K. 1993. "Pattern Matching for Case Analysis: A Computational Definition of Closeness", Proceedings of the 5th International Conference on Computing and Information — ICCI-93, pg. 27-28. Sudbury, Ontario, Canada.

21.) Eugenio, Barbara Di, Glass, Michael. 2004. "The kappa statistic: a second look", Computational Linguistics, v.30 n.1, pp. 95-101.

22.) Fillmore, Charles J. 1968. "The case for case." In Emmond Bach and Robert T. Harms, editors, Universals in Linguistic Theory, pp. 1-88. Holt, Rinehart and Winston, Inc.

23.) Fillmore, Charles J. 1976. "Frame semantics and the nature of language." In Annals of the New York Academy of Sciences: Conference on the Origin and Development of Language and Speech, volume 280, pp. 20–32.

24.) Fillmore, Charles J. 1977. "The Case for Case Reopened." Studies in Syntax and Semantics, pp. 59-81.

25.) Fellbaum, C. 1998 (ed.). "WordNet: An Electronic Lexical Database." The MIT Press.

26.) Frege, G. 1952. "On sense and reference." Translations from the Philosophical Writings of Gottlob Frege, ed. Peter Geach and Max Black, Oxford Blackwell.

27.) Gildea, D. and D. Jurafsky: 2002, "Automatic Labeling of Semantic Roles." Computational Linguistics 28(3), pp. 245–288.

28.) Girju R. 2003. Automatic detection of causal relations for question answering. In Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics (ACL-2003), Workshop on Multilingual Summarization and Question Answering. pp. 76 – 83.

29.) Grinberg, D., Lafferty, J., Sleator D. "A robust parsing algorithm for link grammars" Proceedings of the Fourth International Workshop on Parsing Technologies, pp. 111-125.Prague and Karlovy Vary, September 1995.

30.) Gruber, J.S. 1967. "Look and see." Language, 43(4), pp. 937-947.

31.) Gruber, Jeffrey. 1965." Studies in Lexical Relations." MIT PhD.

32.) Harabagiu, S. M., Miller, G. A., Moldovan, D. I. 1999. "WordNet 2 - A Morphologically and Semantically Enhanced Resource." The Proceedings of the ACL SIGLEX Workshop: Standardizing Lexical Resources, pp. 1-8, Maryland.

33.) Hacioglu, K., Pradhan, S., Ward, W.., Martin, J. H. and Jurafsky, D. 2004. "Semantic Role Labeling by Tagging Syntactic Chunks." In Proceedings of Eighth Conference on Computational Natural Language Learning (CoNLL-2004), pp. 110-113, Boston, USA.

34.) Hearst, M. 1999. "Untangling text data mining." In Proceedings of the ACL'99: the 37th Annual Meeting of the Association for Computational Linguistics, pp. 3-10. University of Maryland.

35.) Jackendoff, R. 1972. "Semantic Interpretation in Generative Grammar." MIT Press, Cambridge,Massachusetts.

36.) Jackendoff. R. 1976. "Toward an Explanatory Semantic Representation." Ling. Inq., 7 89-150.

37.) Jing, Hongyan. 1998. "Usage of WordNet in natural language generation", in COLING-ACL '98 Workshop on Usage of WordNet in Natural Language Processing Systems ,pp. 128-134, New Jersey.

38.) Jones, S. 1964. "Synonymy and semantic classification." Edinburgh. Edinburgh University Press, 1986.

39.) Jurafsky, D. and James H. Martin, 2000. "Speech and Language processing, An introduction to Natural Language Processing, Computational Linguistics and Speech Recognition." Prentice Hall, New Jersey.

40.) Karlsson, Fred, Atro Voutilainen, Juha Heikkilä, and Arto Anttila, editors. 1995. "Constraint Grammar: a language-independent system for parsing unrestricted text." Volume 4 of Natural Language Processing. Mouton de Gruyter, Berlin and New York.

41.) Kirkpatrick, B. (1998). "Roget's Thesaurus of English Words and Phrases." Harmondsworth, Middlesex, England: Penguin.

42.) Krippendorff, K. 1980. "Content Analysis: An Introduction to Its Methodology." Newbury Park, CA: Sage.

43.) Larrick, Nancy. 1961. "Junior Science Book of Rain, Hail, Sleet and Snow." Garrard Publishing Company.

44.) Lloyd, Susan. 1984. Roget's Thesaurus by P. M. Roget. Penguin Books, Ltd. England.

45.) Lenat, D. B. 1995. "CYC: A large-scale investment in knowledge infrastructure." Comm. ACM, 38:11,pp. 33-38.

46.) LDOCE. 1997. "Longman Dictionary of Contemporary English", 3rd Edition.

47.) Levin, B. and M. Rappaport Hovav. 1996. "Lexical Semantics and Syntactic Structure", in S. Lappin, ed., The Handbook of Contemporary Semantic Theory, Blackwell, Oxford, pp. 487-507.

48.) Light, M. 1996. "Morphological Cues for Lexical Semantics." PhD thesis, University of Rochester, Department of Computer Science, Rochester, NY.

49.) Lin, Dekang. 1998. "Dependency-based evaluation of MINIPAR." In Proceedings of the Workshop on the Evaluation of Parsing Systems, First International Conference on Language Resources and Evaluation.

50.) Lesk, M. 1986. "Automatic sense disambiguation using machine readable dictionaries: how to tell a pine cone from an ice cream cone." In Proceedings of the 5th Annual International Conference on Systems Documentation, pp. 24-26.

51.) Masterman, Margaret, 1957. "The thesaurus in syntax and semantics." Mechanical Translation, 4: pp. 1-2.

52.) Jarmasz, M 2003. "Roget's Thesaurus As A Lexical Resource For Natural Language Processing." M.Sc. Thesis, University of Ottawa, Department of Computer Science.

53.) Miller S., Bobrow, R., Ingria, R., and Schwartz, R.. 1996. "A Fully Statistical Approach to Natural Language Interfaces," In Proceedings of the 34th Annual Meeting of the Association for Computational Linguistics, pp. 55-61.

54.) Mishra, Vidya, Niwas, 1966, "The Descriptive Technique of Panini: An Introduction. Mouton & Co." The Hague, Paris.

55.) Michiels, A. and Noël, J. 1982. "Approaches to Thesaurus Production." In Proceedings of the 9th International Conference on Computational Linguistics (COLING-82), pp. 227-232, Prague, Czechoslovakia.

56.) Michiels, A. Mullenders, J. , Noël, J. 1980 "Exploiting a large data base by Longman", Proceedings of the 8th conference on Computational linguistics, pp. 3724-382, Tokyo, Japan

57.) Moschitti, A., Morarescu, P., and Harabagiu, S. 2003. "Open-domain information extraction via automatic semantic labeling." In Proceedings of the 14th Florida AI Conference, pp. 397–401, St. Augustine, FL.

58.) Nakamura, Jun-ichi and Nagao, Makoto. 1988. "Extraction of Semantic Information from an Ordinary English Dictionary and its Evaluation." In Proceedings of the 12 th International Conference on Computational Linguistics, Budapest, pp. 459-464.

59.) Nastase, Viviana. 2003. "Semantic Relations across Syntactic Levels." Ph.D. Thesis. University of Ottawa, School of Information Technology and Engineering.

60.) Nastase, Viviana and Szpakowicz, S. 2003. "Exploring Noun-Modifier Semantic Relations." International Workshop on Computational Semantics, pp. 285-301 Tillburg, Netherlands.

61.) Narayanan, S. and Harabagiu, S. 2004. "Question answering based on semantic structures." In Proceedings of COLING 2004, pp. 693–701, Geneva, Switzerland.

62.) Ogden, CK, Richards, IA. 1946. "The Meaning of Meaning", New York, Harcourt, Brace and World, 8th Edition.

63.) O'Hara, Tom , Mahesh, Kavi, and Nirenburg, Sergei. 1998. "Lexical acquisition with wordnet and the mikrokosmos ontology." COLING-ACL '98 Workshop on the Usage of WordNet in NLP Systems. pp. 94-101.

64.) Pedersen T., Patwardhan S., and Michelizzi J.. 2004. "Wordnet::similarity - measuring the relatedness of concepts." In the Proceedings of the Nineteenth National Conference on Artificial Intelligence (AAAI-04). pp. 1024-1025.

65.) Pradhan, S., Hacioglu, K., Krugler, V., Ward, W, Martin J. H., and Jurafsky, D.. 2005. "Support vector learning for semantic argument classification." Machine Learning Journals special issue on Speech and Natural Language Processing.

66.) Punyakanok, V., Koomen, P., Roth, D. and Wen-tau Yih. 2005. "Generalized Inference with Multiple Semantic Role Labeling Systems." 9th Conference on Computational Natural Language Learning (CoNLL-2005), pp. 181-184. Ann Arbor, Michigan, US.

67.) Quirk, R., Greenbaum S., Leech, G., and Svartvik, J. 1985. "A Comprehensive Grammar of the English Language." Longman.

68.) Quinlan, J. R. 1996. "Improved use of continuous attributes in c4.5," J. Artificial Intelligence. Res., vol. 4, pp. 77—90.

69.) Ratnaparkhi, A. (1998). "Maximum Entropy Models for Natural Language Ambiguity Resolution." Ph.D. thesis, University of Pennsylvania.

70.) Riloff, Ellen. 1993. "Automatically constructing a dictionary for information extraction tasks." In Proceedings of the Eleventh National Conference on Artificial Intelligence (AAAI),pages 811–816, Washington, D.C.

71.) Riloff, Ellen and Mark Schmelzenbach. 1998. "An empirical approach to conceptual case frame acquisition." In Proceedings of the Sixth Workshop on Very Large Corpora, pages 49–56, Montreal, Canada.

72.) Roget, P. (1852). "Roget's Thesaurus of English Words and Phrases." Harlow, Essex, England: Longman Group Ltd.

73.) Scott, S. and Matwin, S. 1998. "Text Classification using WordNet Hypernyms." In S. Harabagiu, editor, Use of WordNet in Natural Language Processing Systems: Proceedings of the Conference, pp. 38-44. Association for Computational Linguistics, Somerset, New Jersey.

74.) Sekine, S. 1996. "Apple Pie Parser: home page". http://cs.nyu.edu/cs/projects/proteus/app

75.) Siegel, S. Castellan N. J. 1988. "Nonparametric statistics for the behavioral sciences." 2nd edition. London: McGraw-Hill.

76.) Sim, Julius, Wright, Chris C. 2005. "The Kappa Statistic in Reliability Studies: Use, Interpretation, and Sample Size Requirements", Physical Therapy Volume 85, Number 3. pp 257-268.

77.) Singh, P. et al. 2002. "Open Mind Common Sense: Knowledge acquisition from the general public." In Proceedings of ODBASE'02. pp. 1223 – 1237.

78.) Somers, HL. 1987. "Valency and Case in Computational Linguistics" (Edinburgh Information Technology Series. Edinburgh: Edinburgh University Press. ISBN 0852245181.

79.) Surdeanu, M., S. Harabagiu, J. Williams, and P. Aarseth: 2003, "Using Predicate-Argument Structures for Information Extraction." In: Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics. Pp. 8-15. Sapporo, Japan.

80.) Tapanainen, Pasi and Järvinen, Timo. 1997. "A non-projective dependency parser." In the proceedings of the 5th Conference on Applied Natural Language Processing (ANLP'97). pages 64-71. ACL, Washington, D.C.

81.) Tjong Kim Sang, Erik F. and Sabine Buchholz. 2000. " Introduction to the CoNLL shared task: Chunking, in Proceedings of CoNLL and LLL 2000", Lisbon, Portugal, pp. 127 – 132.

82.) Temperley, D., Sleator, D., Lafferty, J. 1998. "The LINK parser" http://www.link.cs.cmu.edu/link.

83.) Tesniere, L. 1959. "Elements de syntaxe structurale." Klincksieck, Paris.

84.) "The British National Corpus, Version 2 (BNC World)." 2001. Distributed by Oxford University Computing Services on behalf of the BNC Consortium. URL: http://www.natcorp.ox.ac.uk/

85.) Thompson, C. A., R. Levy, and C. D. Manning: 2003, "A Generative Model for Semantic Role Labeling." In: Proceedings of the European Conference on Machine Learning (ECML).pp. 397-408

86.) Turney, P.D., and Littman, M.L. 2003. "Learning Analogies and Semantic Relations, National Research Council," Institute for Information Technology, Technical Report ERB-1103. (NRC #46488)[8]

87.) Turney, P.D. 2004. "Human-Level Performance on Word Analogy Questions by Latent Relational Analysis." National Research Council, Institute for Information Technology, Technical Report ERB-1118. (NRC #47422)[9]

88.) Wallis, S., Nelson, G. "Knowledge Discovery in Grammatically Analysed Corpora." Data Min. Knowl. Discov. 5(4): 305-335 (2001)

89.) Wilks Y. 1998. "Language processing and the thesaurus." In Proceedings National language Research Institute, Tokyo, Japan, 1998.

90.) Witten, Ian H. and Frank, E. 2005. "Data Mining: Practical machine learning tools and techniques", 2nd Edition, Morgan Kaufmann, San Francisco, 2005.

91.) Wittgenstein, L. 1953. "Philosophical investigations." (Anscombe, G.E.M., trans.). Oxford: Basil Blackwell.

92.) Xue, N. and Palmer, M. 2004. "Calibrating features for semantic role labeling." In Proceedings of the Conference on Empirical Methods in Natural Language Processing, Barcelona, Spain.

93.) Yarowsky, D. 1992. "Word-Sense Disambiguation Using Statistical Models of Roget's Categories Trained on Large Corpora." In Proceedings, COLING-92. Nantes, pp. 454-460.

[8] Turney, P.D., and Littman, M.L. (2005), Corpus-based learning of analogies and semantic relations, *Machine Learning*, 60 (1-3), 251-278.

[9] Turney, P.D. (2005), Measuring semantic similarity by latent relational analysis, *Proceedings of the Nineteenth International Joint Conference on Artificial Intelligence (IJCAI-05)*, Edinburgh, Scotland, pp. 1136-1141.

Appendix A: List of Relations

Relation	Abbrv.	Example	Paraphrase
CAUSALITY			
Cause	caus	flu virus	H makes M occur or exist
Effect	effe	exam anxiety	M makes H occur or exist
Purpose	purp	concert hall	H is for V-ing M
Entailment	Entl	Action in case of fire	H makes M occur or to exist
Enablement	Enbl	car runs if there is gas in the tank	H enables M to occur
Prevention	prev	Service breakdown in case of crash	H opposes M
Detraction	detr	headache pill	H opposes M
TEMPORALITY			
Frequency	freq	daily exercise	H occurs every time M occurs
Time at	tat	morning exercise	H occurs when M occurs
Co-occurrence	cooc	writing novels while listening to music	H & M occurs at the same time
Precedence	prec	Watering the plants before leaving for holidays	H occurs before M
Time to	tto	Party until dawm	H existed until M started to exist
Time from	tfr	Playing well since January	H began to occur when M became true
Time through	tthr	six-hour meeting	H existed while M existed
SPATIAL			
Direction	dir	outgoing mail	H is directed towards M
Location to	Lto	Home town	H is the location of M
Location at	lat	desert storm	H is located at M
Location through	Lthr		H occurred through M
Location	ltd	Home town	H is the location of M
Orientation	Ornt		H is oriented as M
Location from	lfrm	foreign capital	capital H originates at M
CONJUNCTIVE			
Conjunction	conj	Running and swimming	Both H and M exist
Disjunction	Disj	Painting or singing	Either one or both H and M exist
PARTICIPANT			
Agent	ag	student protest	M performs H
Beneficiary	ben	student discount	M benefits from H
Instrument	Inst	laser printer	H uses M
Object	obj	metal separator	M is acted upon by H
Part	part	printer tray	H is part of M
Possessor	posr	national debt	M has H
Possession	poss	Bearded man	M is possessed by H
Product	prod	plum tree	H produces M
Exclusion	excl	Rice instead of potatoes	M is excluded from H
Stative	st	Sleeping dog	H is in a state of M
Accompaniment	acmp	Supper with my family	M is accompanied by H

Object property	obj_prp	Sunken Ship	H underwent M
Source	src	Olive Oil	M is source of H
Whole	Wh	Daisy Chain	M is part of H
Recipient	recp	Wrote a letter to employers for Smilla	H is the recipient of M
Property	prop	Blue book	H is M
QUALITY			
Container	cntr	film music	M contains H
Content	cont	apple cake	M is contained in H
Equative	eq	player coach	H is also M
Order	ord	Flowers in front of window	H is before M in physical space
Material	matr	brick house	H is made of M
Measure	meas	expensive book	M is a measure of H
Manner	manr	Stylish writting	H occurs in a way indicated by M
Type	type	Oak tree	M is a type of tree
Topic	top	weather report	H is concerned with M

Appendix B: List of Dependency Function Assigned by the Parser (Machinese Syntax)

Tag	Explanation	Tag	Explanation
main	Main element	tmp	Time
qtag	Question	ins	Instruememt
v-ch	Verb chain	com	Comitative
pm	Preposed marker	copred	Co predicative
pcomp	Prepositional complement	comp	Subject complement
phr	Verb particle	dat	Indirect object
subj	Subject	oc	Object complement
agt	Agent	obj	Object
dur	duration	frq	Frequency
qua	Quantity	man	Manner
loc	Location	sou	source
goa	Goal	pth	path
cnt	contingency	cnd	Condition
meta	Clause adverbial	cla	Clause initial adverbial
ha	Heuristic	qn	Quantifier
det	Determiner	neg	Negator
attr	Attribute Nominal	mod	Other post modifier
ad	Attribute adverbial	cc	Coordination

Appendix C: Frequency of Occurrence of Semantic Relations in the Dataset

Machinese Syntax:

Relation	*Percentage*	*Relation*	*Percentage*	*Relation*	*Percentage*
caus	1.55%	lto	1.27%	obj_prop	0.17%
effe	1.45%	lthr	0.87%	recp	1.39%
purp	2.41%	lat	4.27%	part	2.13%
entl	0.79%	ltd	1.29%	whole	0.15%
enbl	0.11%	ornt	0%	prod	0.039%
detr	0.23%	conj	3.69%	src	0.99%
prev	0.039%	disj	0.59%	cont	0.059%
cooc	0.25%	acmp	0.059%	top	0.099%
freq	4.59%	agt	18.07%	cntr	0%
prec	0.93%	ben	0.41%	manr	2.07%
tat	3.65%	excl	0.059%	matr	0.29%
tfr	0.19%	st	0.75%	meas	7.99%
tthr	0.65%	prop	10.36%	ord	0%
tto	0.17%	posr	4.07%	equa	1.81%
dir	0.49%	inst	0.45%	type	3.79%
lfrm	1.07%	obj	13.94%		

Xerox Parser:

Relation	*Count*	*Relation*	*Count*	*Relation*	*Count*
caus	1.22%	lto	1.79%	obj_prop	0.30%
effe	1.06%	lthr	0.77%	recp	1.42%
purp	2.12%	lat	4.32%	part	3.36%
entl	0.53%	ltd	2.20%	whole	0.12%
enbl	0.12%	ornt	0.00%	prod	0.14%
detr	0.18%	conj	5.02%	src	1.14%
prev	0.08%	disj	0.55%	cont	0.18%
cooc	0.04%	acmp	0.16%	top	0.46%
freq	4.40%	agt	16.42%	cntr	0.00%
prec	0.34%	ben	0.44%	manr	1.55%
tat	2.91%	excl	0.10%	matr	0.69%
tfr	0.02%	st	0.83%	meas	6.32%
tthr	0.75%	prop	10.16%	ord	0.00%
tto	0.18%	posr	5.14%	equa	2.61%
dir	0.36%	inst	0.65%	type	2.87%
lfrm	1.55%	obj	14.12%		

Appendix D: List of Function Tags (Machinese syntax)

Tag	Explanation	Tag	Explanation
@+FAUXV	Finite Auxiliary predicator	@SUBJ	Subject
@-FAUXV	Non Finite Auxiliary predicator	@F-SUBJ	Formal Subject
@+FMAINV	Finite Main predicator	@OBJ	Object
@-FMAINV	Non Finite Main predicator	@I-OBJ	Indirect object
@PCOMPL-S	Subject complement	@ADVL	adverbial
@PCOMPL-O	Object complement	@O-ADVL	Object Adverb
@APP	Apposition	@VOC	Vocative
@NH	Stray Noun phrase	@DN>	Determiner
@A>	Pre modifier of a nominal	@QN>	Pre modifying quantifier
@AD-A>	intensifier	@<NOM-OF	Post modifying prepositional phrase beginning with of
@<AD-A	Post modifying intensifier	@<NOM	Post modifier of a nominal
@INFMARK>	Infinitive marker to	@<P-FMAINV	Non finite clause as preposition complement
@<P	Other Prepositional complement	@CC	Coordinating Conjunction
@CS	Subordinate conjunction		

Appendix E: List of Surface Tags (Machinese syntax)

Tag	Explanation
%>N	Determiner or Pre modifier of a nominal
%NH	Nominal head
%N<	Post modifier of a nominal
%E>	Pre modifying adverb
%EH	Adverbial head
%<E	Post modifying adverb
%AUX	Auxiliary verb or particle
%VP	Main verb in passive chain
%VA	Main verb in active chain
%EH	Introducer of coordination
%CC	Coordinating conjunction
%CS	Subordinating conjunction

Appendix F: Table of Stop-List Units

Stop-List	Stop-List	Stop-List
$	A	0 – 99 (all numbers)
%	And	Single Alphabets
&	Be	Two character long words
*	For	We'll
+	From	I'll
-	Has	I'd
=	I	You'll
(	In	I'm
)	Is	I've
@	It	Don't
$	Of	
^	On	
,	To	
=>	Not	
--	Your	
;	By	
.	The	
!	with	
:	He	
_	His	
~	She	
[	There	
]	Here	
>	That	
<	This	
{	Then	
}	When	
/	As	
?	Are	
#	You	
*>	My	

Appendix G: Table of Number of Features

Source / Feature type	Number of Features
Xerox Incremental Parser	9 parser based + Phrase type + Keyword
Machinese Syntax	12 parser based + Phrase type + Keyword
LDOCE	6
Keyword Feature	47 (equal to the number of classes)
WordNet Hypernym Feature	47 (equal to the number of classes)
Bag of Words – LDOCE	Approx. 7000
Bag of Words – Roget's	Approx. 7000
Bag of Words – WordNet	Ranging from 7000 – 30,000 for the different WordNet link types

Appendix H: Decision Tree Rules for Machinese Syntax Dataset

Nomenclature of the Features	
Feature	**Explanation**
Head_*	Parser based features for the Head word
Mod_*	Parser based features for the Modifier word
KW_*	The keyword feature
WN_*	The WordNet Hypernym based feature

The top level features in the decision tree are Modifier word and Head word Function tags (*_FT) and Surface tags (*_ST). The Part of speech feature helps pick up the AGENT relation versus the other relations. The Keyword based features especially for spatial and temporal relations. WordNet Feature helps distinguish the harder relations like material, measure relations etc.

A cut section of the Rule Tree generated by Decision Trees in Weka

```
| Mod_FT > 3
| | Head_ST <= 2
| | | KW_cont <= 0
| | | | Mod_CASE <= 0
| | | | | Mod_POS <= 1
| | | | | | Head__PT <= 0
| | | | | | | WN_ltd <= 0
| | | | | | | | Head_FT <= 18
| | | | | | | | | WN_agt <= 0: poss (2.0)
| | | | | | | | | WN_agt > 0: freq (3.0/1.0)
| | | | | | | | Head_FT > 18: disj (2.0/1.0)
| | | | | | | WN_ltd > 0: meas (2.0)
| | | | | | Head__PT > 0
| | | | | | | Head_PRON <= 0
| | | | | | | | WN_meas <= 0
| | | | | | | | | Mod_PT <= 0: prop (5.0/2.0)
| | | | | | | | | Mod_PT > 0
| | | | | | | | | | Head_COMP <= 0
| | | | | | | | | | | WN_lat <= 0
| | | | | | | | | | | | WN_st <= 0
| | | | | | | | | | | | | WN_cont <= 0
| | | | | | | | | | | | | | Head_ST <= 1: meas (3.0/1.0)
| | | | | | | | | | | | | | Head_ST > 1: freq (76.0/36.0)
| | | | | | | | | | | | | WN_cont > 0: prop (2.0)
| | | | | | | | | | | | WN_st > 0: prop (2.0)
| | | | | | | | | | | WN_lat > 0
| | | | | | | | | | | | WN_matr <= 0: prop (9.0/5.0)
| | | | | | | | | | | | WN_matr > 0: freq (8.0/4.0)
| | | | | | | | | | | Head_COMP > 0: freq (5.0/2.0)
| | | | | | | | | | WN_meas > 0: freq (133.0/30.0)
```

```
| | | | | | | Head_PRON > 0: agt (2.0)
| | | | | Mod_POS > 1
| | | | | | Mod_POS <= 3
| | | | | | | Mod_PT <= 0
| | | | | | | | Mod_ST <= 1
| | | | | | | | | Head__PT <= 1: meas (6.0/2.0)
| | | | | | | | | Head__PT > 1: lto (2.0/1.0)
| | | | | | | | Mod_ST > 1
| | | | | | | | | WN_matr <= 0: conj (8.0/1.0)
| | | | | | | | | WN_matr > 0: ltd (2.0/1.0)
| | | | | | | Mod_PT > 0
| | | | | | | | KW_freq <= 0
| | | | | | | | | Mod_FT <= 20
| | | | | | | | | | Head_PRON <= 0
| | | | | | | | | | | KW_lat <= 0
| | | | | | | | | | | | KW_tat <= 0
| | | | | | | | | | | | | Mod_COMP <= 2
| | | | | | | | | | | | | | WN_effe <= 0
| | | | | | | | | | | | | | | KW_tthr <= 0
| | | | | | | | | | | | | | | | WN_agt <= 0
| | | | | | | | | | | | | | | | | WN_lat <= 0: prop (483.0/206.0)
| | | | | | | | | | | | | | | | | WN_lat > 0
| | | | | | | | | | | | | | | | | | Head_FT <= 5: type (10.0/5.0)
| | | | | | | | | | | | | | | | | | Head_FT > 5
| | | | | | | | | | | | | | | | | | | Head_NUM <= 1: prop (48.0/15.0)
| | | | | | | | | | | | | | | | | | | Head_NUM > 1
| | | | | | | | | | | | | | | | | | | | WN_meas <= 0
| | | | | | | | | | | | | | | | | | | | | WN_matr <= 0
| | | | | | | | | | | | | | | | | | | | | | Head_FT <= 7: prop (4.0/1.0)
| | | | | | | | | | | | | | | | | | | | | | Head_FT > 7: meas (13.0/6.0)
| | | | | | | | | | | | | | | | | | | | | WN_matr > 0
| | | | | | | | | | | | | | | | | | | | | | Head_FT <= 7: meas (3.0/1.0)
| | | | | | | | | | | | | | | | | | | | | | Head_FT > 7: prop (3.0/1.0)
| | | | | | | | | | | | | | | | | | | | | WN_meas > 0: part (4.0/2.0)
| | | | | | | | | | | | | | | | | | | WN_agt > 0
| | | | | | | | | | | | | | | | | | | | Mod_FT <= 17
| | | | | | | | | | | | | | | | | | | | | Mod_COMP <= 1
| | | | | | | | | | | | | | | | | | | | | | Head_NUM <= 0: meas (3.0/1.0)
| | | | | | | | | | | | | | | | | | | | | | Head_NUM > 0: prop (121.0/54.0)
| | | | | | | | | | | | | | | | | | | | | Mod_COMP > 1: meas (6.0/3.0)
| | | | | | | | | | | | | | | | | | | | Mod_FT > 17: meas (3.0)
| | | | | | | | | | | | | | | | | | | KW_tthr > 0: meas (25.0/12.0)
| | | | | | | | | | | | | | | | | | WN_effe > 0: type (3.0)
| | | | | | | | | | | | | | | | | Mod_COMP > 2: meas (19.0/2.0)
| | | | | | | | | | | | | | | | KW_tat > 0
| | | | | | | | | | | | | | | | | WN_meas <= 0: prop (12.0/1.0)
| | | | | | | | | | | | | | | | | WN_meas > 0
| | | | | | | | | | | | | | | | | | Head_NUM <= 1: prop (6.0/1.0)
| | | | | | | | | | | | | | | | | | Head_NUM > 1: meas (2.0)
| | | | | | | | | | | | | | | | KW_lat > 0: meas (7.0/1.0)
| | | | | | | | | | | | | | | Head_PRON > 0: poss (2.0)
| | | | | | | | | | | | | | Mod_FT > 20: conj (5.0/2.0)
| | | | | | | | | | | | | KW_freq > 0: freq (4.0)
```