Analysis and Construction of Noun Hypernym Hierarchies to Enhance Roget's Thesaurus

by

Alistair Kennedy

Thesis submitted to the
Faculty of Graduate and Postdoctoral Studies
In partial fulfillment of the requirements
For the degree of Master of Computer Science (MCS)

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

Abstract

Lexical resources are machine-readable dictionaries or lists of words, where semantic relationships between the terms are somehow expressed. These lexical resources have been used for many tasks such as word sense disambiguation and determining semantic similarity between terms. In recent years some research has been put into automatically building lexical resources from large corpora. In this thesis I examine methods of constructing a lexical resource, not from scratch, but rather expanding existing ones. Roget's Thesaurus is a lexical resource that groups terms together based on degrees of semantic relatedness. One of Rogets Thesaurus' weaknesses is that it does not specify the nature of the relationships between terms, it only indicates that there is a relationship. I attempt to label the relationships between terms in the thesaurus. These relationships could include: synonymy, hyponymy/hypernymy and meronymy/holonymy. I examine the Thesaurus for all of these relationships. Sources of these relationships include other lexical resources such as WordNet, and also large corpora and specialized texts such as dictionaries. Roget's Thesaurus has other weaknesses including a somewhat outdated lexicon. Our version of Roget's Thesaurus was created in 1987 and so does not contain words/phrases related to the Internet and other advances since 1987.

I examine methods of creating a hypernym hierarchy of nouns. A hierarchy is constructed automatically and evaluated manually by several annotators who are fluent in English. These hypernyms are intended to be used in a system where a human annotator is given a set of hypernyms and indicates which are correct and which are incorrect. This is done to facilitate the process of constructing a lexical resource, a process which was previously done manually.

I import over 50,000 hypernym relationships to Roget's Thesaurus. An estimated overall accuracy of 73% is achieved across the entire hypernym set. As a final test the new relationships imported to the Thesaurus are used to improve Roget's Thesaurus capacity of calculating semantic similarity between terms/phrases. The improved similarity function is tested on several applications that make use of semantic similarity. The relationships are also used to improve Roget's Thesaurus' capacity for solving SAT style analogy questions.

Acknowledgements

I would like to thank Dr. Stan Szpakowicz for supervising me. Mario Jarmasz for providing the Java interface for, and insight into, Roget's Thesaurus and also for many of the data sets. Peter Turney and his colleagues at NRC/IIT for giving us access to their copy of the Waterloo MultiText System with the terabyte corpus of web data as well as the SAT analogy problems. I would also like to thank Dr. Stan Szpakowicz, Darren Kipp, Dr. Vivi Nastase and Ramanjot Singh Bhatia for graciously taking the time to annotate my data. The Natural Sciences and Engineering Research Council of Canada (NSERC) and the University of Ottawa support my research.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

Lexical Resources are valuable tools for many Natural Language Processing (NLP) applications. They provide semantic information about terms and concepts and their relationships with each other. For a lexical resource to be truly useful for NLP applications, it must track relationships between as many words and phrases as possible for the language it is intended. This is a problem, since the only way to accurately construct a lexical resource is to manually construct it. In recent years work has been done on ways to automatically construct these lexical resources, mostly by creating hypernym hierarchies (Hearst, 1992; Sombatsrisomboon et al., 2003; Snow et al., 2005; Rydin, 2002; Shinzato and Torisawa, 2004; Cederberg and Widdows, 2003; Caraballo and Charniak, 1999). In this thesis I will examine methods of not constructing a lexical resource from scratch, but rather expanding an existing lexical resource in the hopes of increasing its usefulness for NLP.

One of the best known lexical resources used in NLP is WordNet (Fellbaum, 1998b). WordNet was constructed manually at Princeton University. In this thesis WordNet 2.0 is used unless specifically stated otherwise. It contains many English words and phrases as well as relationships between these words and phrases. The words and phrases are mapped to concepts called "synsets". The nouns and verbs in this lexical resource are contained in hierarchies based on hypernym/hyponym relationships. Other hierarchies, such as a meronym/holonym hierarchy for nouns also exist in WordNet. Although WordNet is popular, it is not yet complete (Fellbaum, 1998b). There are many words and phrases that exist in Roget's Thesaurus that cannot be found in WordNet. Roget's Thesaurus is a lexical resource that arranges terms and phrases into a hierarchy of groups. At a high level words are broken down into various topics, and several levels of sub-topics.

At a lower level words are grouped together based on some sort of close semantic relatedness. Often the terms in a semicolon group (the smallest grouping in Roget's Thesaurus) are near synonyms, however other relationships are possible. Although it is not always clear what the criteria for placing two terms in the same group are, it is clearly observable that words that are in the same group, or groups close together, are semantically closer related than terms that appear in groups farther apart (Jarmasz and Szpakowicz, 2003b). One of the principal hindrances from using Roget's Thesaurus for NLP purposes is the lack of explicit relationships between its terms and phrases. Although it is easy for people to deduce that terms are related, it is not clear in what way. I will examine the feasibility of importing relationships, especially hypernym/hyponym relationships into the thesaurus. I also examine the possibility of introducing other relationships like meronymy, however importing such relationships does not appear to be productive.

The goal of this thesis is to incorporate explicit relationships (particularly hypernyms) into Roget's Thesaurus and to evaluate those relationships. Several methods of automatically extracting hypernyms are tested and evaluated. The best hypernym relationships are then incorporated into Roget's Thesaurus.

1.1 Roget's Thesaurus

Dr. Peter Mark Roget first developed Rogets Thesaurus over 150 years ago. It was originally compiled from lists of synonyms that Roget had created during his career as a physician (Jarmasz, 2003). Over time many different versions of the thesaurus have been created. The version of Roget's Thesaurus that will be used in this thesis is the 1987 *Penguin's Roget's Thesaurus* (Kirkpatrick, 1987).

The 1987 *Penguin's Roget's Thesaurus* version of Roget's Thesaurus has been built into an Electronic Lexical Knowledge Base (ELKB) (Jarmasz and Szpakowicz, 2001a). This ELKB is a machine tractable version of the thesaurus. It allows Java applications to have access to the terms, phrases and structure of the thesaurus. A search for a single term will result in a list of locations in the thesaurus in which that term appears.

The usefulness of Roget's Thesaurus for Natural Language Processing has also been demonstrated. Roget's Thesaurus has been shown to be an excellent resource for determining semantic relatedness of words (Jarmasz and Szpakowicz, 2003b). It has also been used for creating lexical chains, which could be used for such applications as text summarization (Jarmasz and Szpakowicz, 2003a).

1.2 Getting Relationships

New relationships for Roget's Thesaurus will be retrieved from several sources. The first source to be examined is WordNet (Fellbaum, 1998b). Synonyms, Hypernyms and Hyponyms can be extracted from WordNet. Other lexical resources and Ontologies like Cyc (Lenat, 1995) are also examined as potential sources of relationships between terms. Relationships from WordNet and Open Cyc are incorporated into Roget's Theaurus.

The second source of relationships is large corpora. By identifying patterns that indicate hypernym/hyponym relationships between terms, and then searching for them, it is possible to identify new hypernym/hyponym relationships. Relationships that are discovered using this method are added to Rogets Thesaurus. These relationships are taken from two different sources, the British National Corpus (BNC), (Burnard, 2000) and the Waterloo MultiText (WMT) System (Clarke and Terra, 2003).

A third source is specialized text. Specialized texts are dictionaries and other texts that are written in a regulated style for a specific purpose. Unlike mining information from general text one can take advantage of a standardized format of text in order to extract more accurate relationships. LDOCE (Procter, 1978) and Wiktionary (Wiktionary, 2006) are mined for relationships. Different dictionaries will have different degrees of standardization. The style and format in which LDOCE is written is much more strictly enforced than that of Wiktionary. The relationships are incorporated into Roget's Thesaurus.

It should be noted that although the relationships mined from these resources are used to enhance Roget's Thesaurus, this is in fact, only one possible application of these relationships. These relationships could also be used to enhance other lexical resources, or they could be useful for solving some specific NLP application.

1.3 This Thesis

This thesis is divided into 7 chapters. This Chapter 1 introduces the problem being addressed, and the goals of this thesis. In Chapter 2 the lexical resources and Ontologies used in this thesis are described, as well as some work done on constructing and using them. Chapter 3 examines previous work done on the problem of mining relationships from regular text and dictionaries. Chapter 4 contains an analysis of Roget's Thesaurus where the kinds of relationships that exist in Roget's Thesaurus are examined. In Chapter 5 I explain the methodology that is used to import hypernym relationships to Roget's

Thesaurus. Chapter 6 contains the results from my experiments constructing and evaluating the resource. This includes a discussion of human evaluation and kappa scores. It also contains an analysis of the resource through testing it on several application domains. The final chapter is Chapter 7. It offers a conclusion and proposes future work that can be done to further expand and test the Thesaurus. This includes a discussion of other resources from which relationships can be extracted, other types of relationships that can be added, and other NLP applications that could be used to test then enhanced Roget's Thesaurus.

Chapter 2

Lexical Resources and Ontologies

Lexical resources have become indispensable for many NLP applications. This chapter describes the lexical resources used in this thesis, particularly Roget's Thesaurus and WordNet. It also discusses some of the applications that these lexical resources have been used for, and work done combining these lexical resources.

2.1 Roget's Thesaurus

Roget's Thesaurus divides its terms into a variety of semantic *Classes*, *Sections*, *Su-Sections* etc. There are a total of 8 different levels of semantic relatedness represented in the Thesaurus (Jarmasz, 2003). The *Classes* are broken down into *Sections* as shown in Figure 2.1. The hierarchy of the Thesaurus is as follows:

- The whole Thesaurus is divided into *Classes*
- *Classes* are divided into *Sections*
- *Sections* are divided into *Sub-Sections*
- *Sub-Sections* are divided into *Head Groups*
- *Head Groups* are divided into *Heads*
- *Heads* are divided into *Part of Speech (POSs)*
- *POSs* are divided into *Paragraphs*
- *Paragraphs* are divided into *Semicolon Groups*

1 Abstract Relations
- Existence, Relation, Quantity, Order, Number, Time, Change, Causation
2 Space
- Space in general, Dimensions, Form, Motion
3 Matter
- Matter in general, Inorganic matter, Organic matter
4 Intellect: formation of ideas
- Intellectual operation in general, Precursory conditions and operations, Materials for reasoning, Reasoning processes, Results of reasoning, Extension of thought, Creative thought
5 Intellect: communication of ideas
- Nature of ideas communicated, Modes of communication, Means of communicating ideas
6 Volition: individual volition
- Volition in general, Prospective volition, Voluntary action, Antagonism, Results of action
7 Volition: social volition
- General social volition, Special social volition, Conditional social volition, Possessive relations
8 Emotion, religion and morality
- General, Personal emotion, Interpersonal emotion, Morality, Religion

Figure 2.1: The Classes and Sections of Roget's Thesaurus

- *Semicolon Groups* are divided into *Words and Phrases*

There are a total of 990 heads. Each head contains up to 5 parts of speech, those being nouns, verbs, adjectives, adverbs and interjections (Jarmasz and Szpakowicz, 2001a). Not every part of speech needs to appear in each head. Most contain nouns, however few contain interjections, and many do not contain verbs, adjectives and adverbs. An example of a head can be seen in Figure 2.2. Roget's Thesaurus can sometimes be confusing, it is not always clear what the relationships between the words in the same semicolon group, or even paragraph are (Jarmasz and Szpakowicz, 2001b).

2.1.1 Applications of Roget's Thesaurus

Roget's Thesaurus has been shown to be a useful resource for a variety of NLP applications. Roget's Thesaurus has been used to measure semantic distance between words/phrases (Jarmasz and Szpakowicz, 2003b). Distance between terms is measured by finding the shortest distance through the thesaurus between these two terms. There are 9 different levels of granularity in the thesaurus and a score is assigned based on which level of granularity these pair of terms/phrases appear.

- Length 0 : same semicolon group
- Length 2 : same paragraph
- Length 4 : same part of speech
- Length 6 : same head
- Length 8 : same head group
- Length 10 : same sub-section
- Length 12 : same section
- Length 14 : same class
- Length 16 : in the thesaurus or not found

Using this system of assigning lengths based on granularity in the thesaurus a fairly good estimate of semantic similarity between two words can be found. On the Miller and Charles (1991) list of 30 pairs it had a correlation of .878, and for the Rubenstein

Class 5: Intellect: communication of ideas
Section 3: Means of communicating ideas
Sub-Section: Means of communicating ideas
Head Group: 557 Language
Head: 557 Language

N. *language,* tongue, speech, idiom, parlance, talk, dialect,; langue, parole,; spoken language, living living,; patter, lingo, 560 *dialect*,; personal language, idiolect,; mother tongue, native tongue,; vernacular, common speech, demotic speech, vulgar tongue,; colloquial speech, English as she is spoken, 579 *speech*,; Queen's English,; correct speech, idiomatic speech, slang, jargon, vulgarism,; lingua franca, koine, Swahili, creole, pidgin, pidgin English, b<E8>che-de-mer,; sign language, semiology, 547 *gesture*,; diplomatic language, international language, International Scientific Vocabulary, Basic English,; pasigraphy,; artificial language, Esperanto, Ido, Volapuk,; official language, Mandarin, Hindi,; Received Pronunciation, Standard English, BBC English,; officialese, translatorese, 560 *neology*,; machine language, 86 *data processing*,; learned language, dead language, Latin, Greek, Sanskrit,; metalanguage,; confusion of tongues, polyglot medley, Babel, babble, 61 *confusion*,.
...
ADJ. *linguistic,* lingual, philological, etymological, grammatical, morphological,; diachronic, synchronic,; lexicographical, lexicological, onomasiological, semasiological, semantic,; analytic,; agglutinative,; monosyllabic,; tonal, inflected,; holophrastic,; correct, pure,; written, literary, standard,; spoken, living, idiomatic,; vulgar, colloquial, vernacular, slangy, jargonistic, 560 *dialectal*,; local, enchorial, familial,; current, common, demotic,; bilingual, diglot,; multilingual, polyglot,.
...

Figure 2.2: Sample of Head 557: Language from Roget's Thesaurus

and Goodenough (1965) list of 65 pairs it had a .818 correlation, as reported in (Jarmasz and Szpakowicz, 2003b). Roget's Thesaurus was also put to use in solving synonym problems, such as those in Test Of English as a Foreign Language (TOEFL)(Landauer and Dumais, 1997), English as a Second Language (ESL) (Turney, 2001) and Reader's Digest Word Power Game (RDWP) (Lewis, 2001), where a correct synonym must be selected from a group of four. Roget's Thesaurus was found to be about 79% accurate on 80 TOEFL questions, 82% on 50 ESL questions and 74% on 300 Reader's Digest questions. The Reader's Digest questions come from Reader's Digest Word Power Game. Although it is not standard in Computational Linguistics it can still be used to compare the different similarity metrics. These scores were compared against a variety of other systems, including WordNet edge counting. In most cases Roget's Thesaurus outperforms the other systems (Jarmasz and Szpakowicz, 2003b).

Six methods for semantic similarity using WordNet (Hirst and St-Onge, 1998; Leacock and Chodorow, 1998; Resnik, 1995; Jiang and Conrath, 1997; Lin, 1998b) including edge counting were compared against Roget's Thesaurus. The method used by Hirst and St-Onge (1998) starts with basic edge counting in WordNet but modifies it to take advantage of the changes in direction, i.e. changes in relationship types along the path. Changing relationships less often make the terms appear to be more similar.

$$rel_{HS}(c_1, c_2) = C - path\ length - k * d$$

C and k are constants while d is the number of times the direction changes. In Leacock and Chodorow (1998) the length of the path *len* as well as its depth in the hierarchy are used to determine semantic similarity.

$$sim_{LC}(c_1, c_2) = -log\frac{len(c_1, c_2)}{2D}$$

D is the overall depth in the hierarchy. In Resnik (1995) a method that uses the lowest common subsumer, or lowest super-oriented *lso* and its probability of appearing in a specific corpus are used to determine the semantic relatedness of two terms.

$$sim_R(c_1, c_2) = -log(p(lso(c_1, c_2)))$$

The probability *p(x)* can be thought of as *information content* of the lowest common subsumer and is determined using the frequency counts in WordNet. In Jiang and Conrath (1997) another method taking advantage of frequency counts in WordNet. This one considers frequency counts for several of the individual words as well as their subsumer.

$$sim_R(c_1, c_2) = 2 * log(p(lso(c_1, c_2))) - (log(p(c_1)) + log(p(c_2)))$$

The final method, proposed in Lin (1998b) is another variation on information content and the lowest common subsumer.

$$sim_R(c_1, c_2) = \frac{2 * log(p(lso(c_1, c_2)))}{log(p(c_1)) + log(p(c_2))}$$

Of these systems Lin (1998b) performed the best of both the Miller and Charles (1991), and Rubenstein and Goodenough (1965) data sets. For the TOEFL, ESL and RDWP tests Hirst and St-Onge (1998) had higher scores than the other WordNet based measures (Jarmasz and Szpakowicz, 2003b).

In another experiment the 5 measures of semantic relatedness in WordNet (Hirst and St-Onge, 1998; Leacock and Chodorow, 1998; Resnik, 1995; Jiang and Conrath, 1997; Lin, 1998b) were used for the problem of spell checking (Budanitsky and Hirst, 2006). The test is done under the assumption that if a word is semantically different from many of the other words surrounding it then it is likely to be a spelling error of some sort. A corpus was created where words are replaced by similarly spelled, but semantically unrelated words. These similarity functions were used to try to detect which words were incorrect. Different window sizes were used when detecting incorrect words. It was found that Jiang and Conrath (1997) worked the best at all window sizes. Hirst and St-Onge (1998) was quite good at low window sizes, however as the window increase it rapidly fell below the other metrics. This shows that for three different tests, three different WordNet based similarity functions all appeared to be best, one for each test.

Another application of Roget's Thesaurus is the construction of Lexical Chains (Jarmasz and Szpakowicz, 2003a). A lexical chain is a sequence of words extracted from a text where all the words can represent the same topic (Morris and Hirst, 1991). Building lexical chains can be divided into 4 steps. The first step is to collect a set of candidate words. Removing stop words, and other words that appear too frequently in the text accomplishes this. The second step is to find an appropriate chain for each candidate word. A term can be included in a chain if its morphological base already exists (i.e. the word is repeated) or if the term exists in the same head as another term in the chain. If the term cannot be put into an existing chain then a new one is created. The third step is to insert the word into the chain. Words are added to a chain, however no word sense disambiguation takes place. If no words are added after five sentences then they stop adding to the chain. The final step is to merge lexical chains and only keep the strongest ones. This step is left open and relies on an evaluation scheme for determining if a lexical chain is useful or not.

2.1.2 Factotum

The framework for an Ontology called FACTOTUM has already been designed out of a much older version of Roget's Thesaurus (Cassidy, 2000). In this case the 1911 version of the thesaurus was used. In FACTOTUM concepts can inherit properties from multiple other concepts. In addition to this each concept can have explicit semantic links to other concepts. This plan has not been fully implemented. FACTOTUM is meant as more of a framework that can be adapted to be used for various specific applications. A total of 150 different kinds of relations are accepted in FACTOTUM. These relations were discovered by starting with the 30 relations from The Unified Medical Language System (UMLS) meta thesaurus and then breaking them down to a total of 150. For each relationship there is a corresponding inverse relation (e.g. hypernym is inverse of hyponym) with the exception of symmetric relationships (e.g. synonyms and antonyms). The end result of this work is a semantic resource that is quite different in design than WordNet.

A method of adding functional relationships to FACTOTUM was shown in O'Hara and Wiebe (2003). A functional relationship is any non-hierarchical relationships with the exception of attributes. An example of a functional relationship could be: drying *is-function-of* dryer. A corpus of disambiguated text from the SENSEVAL competition was used. Collocations from FACTOTUM were taken (as well as hypernyms of the terms in the collocation) and found in the corpus. Prepositions were then extracted from the collocations. The prepositions were used as features, with the known relationship type in order to train a decision tree to identify the functional relationship type. 21 different functional relationships from FACTOTUM were used for this experiment. Hierarchical relationships such as *has-subtype* and *has-part* were not used. O'Hara and Wiebe (2003) achieved 71% accuracy across all 21 different relationship types. This task can be thought of as disambiguating prepositions into FACTOTUM functional relationships.

2.2 WordNet

WordNet is a very popular lexical resource for many NLP applications. There are a variety of reasons behind this, some of its initial success came because it is freely available, and frequently updated. It has also been proven to be a valuable tool for many NLP applications. Terms are not ordered alphabetically, but rather are grouped together as concepts in a network of concepts (Fellbaum, 1998b). Psycholinguistics techniques, such as word association tests were employed in order to discover the best way to build such a

lexical resource. The goal with WordNet was to organize word senses by meaning, rather than alphabetically. This poses a problem since a word can have multiple meanings, and multiple words can have the same meaning. To resolve this there is a many-to-many relationship between words senses and the concepts they represent. Words are organized into synsets, which represent concepts. Each word from a synset is a synonym of another word. Polysemous words can have their different senses exist in different synsets. Four different parts of speech are recognized in WordNet: nouns, verbs, adjectives and adverbs. Interjections are not included in WordNet, however they are included in Roget's Thesaurus.

The synsets of WordNet are connected through a variety of relationships. The main relationships in WordNet are synonymy, antonymy, hyponymy/hypernymy and meronomy/holonymy. Morphology is also accounted for in WordNet. For example *trees* is known to be the plural of *tree* (Miller, 1998a). The nouns in WordNet are organized into one of nine hierarchies. This way a term can be described by other terms without the risk of a circular definition. This hierarchies are hypernym/hyponym hierarchy. An example from a noun hierarchy can be seen in Figure 2.3. These hierarchies were constructed under the supposition that people stored information in their lexical memory similar to the hierarchy. Although this supposition is hard to prove it is a factor in deciding to order terms into a hierarchy (Miller, 1998a). Each hierarchy has its own top element, they are:

- entity
- psychological feature
- abstraction
- state
- event
- act
- group
- possession
- phenomenon

```
Synonyms/Hypernyms (Ordered by Estimated Frequency) of noun thesis

2 senses of thesis

Sense 1
thesis
   => premise, premiss, assumption
     => postulate, posit
       => proposition
         => statement
           => message, content, subject matter, substance
             => communication
               => social relation
                 => relation
                   => abstraction

Sense 2
dissertation, thesis
   => treatise
     => writing, written material, piece of writing
       => written communication, written language
         => communication
           => social relation
             => relation
               => abstraction
```

Figure 2.3: WordNet hypernym chains from the two senses of the word “Thesis”

There is also a separate hierarchy for verbs. Other relationships, such as meronymy, exist between words in WordNet. Some relationships also go between parts of speech.

Meronymy information goes between nouns in WordNet. The hyponym trees can be used to inherit meronym information from a concepts parent. For example *bird* has *beak* and *wing* as meronyms. This means that hyponyms of the word: *bird* like *parrot* and *hen* inherit these meronyms (Miller, 1998a).

WordNet also contains adjectives. These adjectives can be broken down into two main groups, descriptive adjectives such as *big, interesting* and *possible* and relational adjectives such as *presidential* and *nuclear* (Miller, 1998b). There is also a smaller number of reference modifying adjectives such as *former* and *alleged*. For verbs there are two main relationships: hyponymy and entailment (Fellbaum, 1998a). In logical entailment, if A entails B then this means that if A is true, then B cannot be false. For lexical entailment this principle is applied to verbs. For example if a sentence *Someone V1* logically entails *Someone V2*, then *V1* entails *V2*. The relationship is not commutative. The entailment relationship with verbs is similar to the meronym relationship with nouns. Verbs also have hypernym relationships in WordNet. These verb hypernym relationships are a bit different from the noun hypernyms in WordNet since if one says "An x *is a* y" works for nouns but not for verbs. A hypernym hierarchy is built for verbs. An example from the verb hierarchy would be if one takes the verb *move* then all the different ways of moving would be hyponyms. For example *slide* is a manner of moving, and *pull* is a cause of moving, and so these can be hyponyms of *move*. A similar kind of relationship is *Troponymy*. Two verbs V1 and V2 are troponyms if they work in the following sentence *to V1 is to V2 in some particular manner*. Not all the the verb hypernyms in WordNet are actually troponyms, however many are (Fellbaum, 1998a).

2.2.1 Merging Roget's Thesaurus with WordNet

In the past some people have attempted to map terms and phrases from Roget's Thesaurus onto the words and phrases from WordNet. This task can be thought of as word sense disambiguation, where one is disambiguating senses of words in one lexical resource into senses in the other lexical resource. This is a particularly difficult task as the mapping will be many to many (Kwong, 1998a).

In Kwong (1998a), LDOCE was used as an intermediary for mapping between Roget's Thesaurus and WordNet. The first step was to take all the definitions for every sense of a word from LDOCE and put them in a matrix holding similarity scores between every

sense of the word in LDOCE and every sense of the word in WordNet. To compare the two resources, the number of words overlapping between the LDOCE definition, and the hypernyms, synonyms and gloss from WordNet was computed. The definitions for LDOCE were then compared against senses from Roget's Thesaurus, but this time counting the overlap between the LDOCE definition and the words/phrases in the Roget's paragraph where the word appears. For each sense from LDOCE, she found the maximum scoring sense from Roget's Thesaurus, and the maximum scoring sense from WordNet. The Roget's and WordNet senses were then mapped to the LDOCE senses.

This technique was tested on a set of 36 nouns that were randomly selected. The words were divided into groups based on how polysemous they were. Three degrees of polysemy, (low, medium and high) were examined. Low polysemy words are those words with 5 senses or less, medium polysemy is 5 to 10 senses, and high polysemy is 11 senses or more. Mappings from WordNet to Roget's Thesaurus were examined. It was found that the mapping was about 79% correct with low polysemy words while medium and high polysemy words had closer to 70% accuracy. For the low polysemy words, most of the unmapped words, did not have corresponding senses in Roget's Thesaurus (Kwong, 1998a).

A similar experiment was carried out in Kwong (1998b). Only this time LDOCE senses were mapped to Roget's Thesaurus and WordNet is used as an intermediary. The same technique as used in Kwong (1998a) was used to map senses from one resource to another. This method was tested on 12 different nouns and was found to be 55.4% accurate at assigning LDOCE definitions to terms in Roget's Thesaurus.

Words from Roget's Thesaurus were mapped to WordNet word senses to aid in the task of Word Sense Disambiguation (Kwong, 2001). Different metrics were used to determine the sense of a word. Semantic similarity to near by words calculated using WordNet was the first method. Counting of neighboring words from the Heads of Roget's Thesaurus was second. Counting co-occurrences of near by words with the WordNet definitions was a third. The last method was to determine if a word sense was to pick the Head from Roget's Thesaurus that had the most overlap within the whole document. It was found that combining the first three scores had the most accurate results (Kwong, 2001).

Another similar experiment was done in Nastase and Szpakowicz (2001). This time LDOCE was not used as an intermediary. Synonyms, hypernyms, hyponyms, meronyms and holonyms were extracted from senses in WordNet and co-occurrences were counted in Roget's Thesaurus paragraphs. This was done for all parts of speech where the listed

relationships are available. For a set of 719 nouns a precision of 55% was achieved. One reason for the lower score is that some Roget's paragraphs are very similar and get similar scores.

2.3 Ontologies

There are a variety of Ontologies available for use today. Ontologies generally do not focus on building lexical hierarchies, but rather deal more with concepts and facts, and what can be inferred from these facts. Some Ontologies are domain specific, where they deal well with a particular topic, but are not good for general use.

2.3.1 Cyc

Cyc was designed to contain common knowledge about the real world. Currently over 900 person years have been put towards building this database of common knowledge (Matuszek et al., 2006). Cyc contains 2.2 million assertions (facts) and describes over 250,000 terms in its full version. A partial version called OpenCyc has been released freely to the public. It contains only a small subset of the assertions and terms from Cyc. A language called CycL is used to query the Cyc database. CycL employs first order logic as well as many higher order extensions (Matuszek et al., 2006). The system contains a quoting mechanism that allows it to distinguish knowledge concerning a topic from knowledge concerning a term that represents a concept. Through this query language it is possible to find information about a particular concept. Information about a concept, can be its relationship to other concepts, as well as what kinds of relationships are possible with this concept. Cyc allows for the representation of both individuals as well as collections and collections of collections. This is important in an ontology because while history often talks about individual people and their accomplishments, science tends to talk about properties of an entire class (Matuszek et al., 2006).

The Cyc ontology is broken down into three different parts, the upper, middle and lower Ontologies. The upper ontology is primarily concerned with regulating the number of aspects in the ontology itself. This would be where the ontology defines what kind of things can be represented. The middle ontology contains abstractions that are widely used, for example geospatial relationships and knowledge of human interaction (Matuszek et al., 2006). The lowest level of the ontology contains precise information that is mostly useful to specific domains. It is the largest part of the ontology, but is the least broadly

applicable (Matuszek et al., 2006).

2.3.2 The Unified Medical Language System (UMLS)

Another ontology, designed for a specific domain, is the Unified Medical Language System (UMLS). This system was designed as a system for categorizing medical research documents (Humphreys and Lindberg, 1993). It contains a great deal of information about topics relevant to medical research and medical conditions.

UMLS was designed to aid in connecting users to relevant machine readable information (Humphreys and Lindberg, 1993). UMLS can be broken down into 3 different parts: The Metathesaurus, the Semantic Network and the Information Source Map. The Metathesaurus is a large multi-purpose and multi-lingual database of terms. The terms are organized by concept rather than alphabetically. The Metathesaurus is constructed from a variety of sources with information useful to medical document classification, "It is built from the electronic versions of many different thesauri, classifications, code sets, and lists of controlled terms used in patient care, health services billing, public health statistics, indexing and cataloging biomedical literature, and /or basic, clinical, and health services research" (United States National Library of Medicine: National Institute of Health, 2004).

The semantic network contains definitions for the semantic types, and kinds of relationships available in the Metathesaurus. Currently there are a total of 135 semantic types and 54 relationships in the Semantic Network. The semantic types can include objects, events, concepts and many others (United States National Library of Medicine: National Institute of Health, 2004).

The third part is the Special Lexicon and Lexical Programs. This is a controlled lexicon of biomedical terms that are used by the SPECIALIST NLP system. It includes morphological, syntactic and orthographic information about its terms (United States National Library of Medicine: National Institute of Health, 2004).

From UMLS even more domain specific Ontologies can, and have, been built. UMLS has been expanded to deal specifically with blood transfusion data (Achour et al., 1999). First new, useful, medical terms were extracted from text. This was done by interviews with medical experts. Corresponding terms and concepts were then extracted from UMLS. The new terms were then mapped to the correct concepts, and finally new relationships were added. Medical experts also added these relationships. This expanded subset of UMLS was designed to be used in a blood transfusion decision support appli-

cation which was to be integrated into the Henri Mondor Hospital Information system (Achour et al., 1999).

2.4 Conclusion

A great deal of work has been put into designing Lexical Resources and Ontologies. Roget's Thesaurus has been proven to be a valuable resource in the past (Jarmasz and Szpakowicz, 2003b,a). Some work has been put into merging lexical resources, as well as dictionaries (Kwong, 1998a,b, 2001).

Chapter 3

Literature Review

This chapter describes past work done on mining relationships from text. It focuses on mining hypernym relationship, as those are the relationships extracted in this Thesis. Techniques for mining dictionaries as well as regular text are discussed.

3.1 Acquiring Relationships from Specialized Text

Specialized texts such as dictionaries have potential to be excellent sources of hypernyms. I define *Specialized Texts* to be any text that is written and organized in a specific format, for a specific purpose. These kinds of texts will be easier to mine relationships from than general text since they have structure that can be exploited.

In recent years there has not been so much work done on mining relationships from dictionaries, although if one looks further back some work has been done on mining LDOCE. In Nakamura and Nagao (1988) two patterns were used to extract relationships from LDOCE. These patterns are:

- {determiner} {adjective}* *key noun* {adjective phrase}*
- {determiner} {adjective}* *function noun* of *key noun* {adjective phrase}*

An adjective phrase can be thought of as any phrase that modifies a noun or pronoun. A function noun is any noun that indicates a semantic relationship between the word being defined and the key noun. For example “kind”, “type”, “part”, or “set” are examples of function nouns (Nakamura and Nagao, 1988). The key noun is related to the word being defined, by way of the function noun. In cases where there is no

function noun, the relationships is usually hypernymy. Accuracies for the different kinds of relationships are not shown. Some examples of definitions in LDOCE are:

- Adagio: A piece of music to be played or sung slowly.
- Thesaurus: A book in which words are put into groups with other words that have similar meanings.

Similar work was done in Alshawi (1987). Patterns were used to extract information from definitions other than key nouns. In this case the entire definition was examined to discover relationships with several different words. This method is definitely not transportable to other dictionaries since it relies on a strict set of rules being followed when writing the definitions. A sample of 500 definitions was taken. From these samples the head word from the definition was correctly identified in 387 definitions. Of those definitions where the head was identified, additional information was discovered in 236 of those definitions. This additional information was judged to be 88% accurate.

In Guthrie et al. (1990) a method for automatically constructing a taxonomy from a dictionary was explored. Several heuristics were developed to determine the word sense of the hypernym (key/head noun) taken from the definition. Each definition in LDOCE has a semantic code. The sense of the key/head noun that has a semantic code closest to that of the original definition was taken to be the correct sense.

Work in this area was criticized in Ide and Véronis (1993). One of the problems pointed out is that often the hypernyms are not nearest hypernyms. That is one might get "cat *is an* animal" when perhaps "cat *is a* mammal *is a* vertebrate *is an* animal" would be more appropriate path between "cat" and "animal". Another problem is that sometimes the hypernym is far too general, for example: "a cigar *is a* thing" could be discovered. Sometimes there are multiple hypernyms for a single term. Their example is that a saucepan is both a pot and a pan, however pot and pan are not synonyms. Another problem is that sometimes, circular hypernym chains will be found. This is where a word is its own hypernym further down the hyponym chain. They show that using several different dictionaries and merging the results together can improve the results. Putting relationships from several resources together decreases the frequency of the above described errors. All of these issues raised will be problems with any resource used for hypernym mining.

Another method for mining relationships from dictionaries was proposed in Senellart and Blondel (2003). This method focused on automatically finding synonyms in a dictionary. A graph was made up where each word in the dictionary is a word, and an edge

appears between terms if one term appears in the definition of the other. A query word w was then chosen and a sub-graph is created where all vertices either have an edge pointing to w or from w. All terms that have both an incoming edge and an outgoing edge in the sub graph were chosen as synonyms of w. To evaluate this method 21 people ranked several synonym pairs with scores between 0 and 10. For the words *disappear* and *sugar* an average of over 6 was achieved, however for other worse *parallelogram* and *science* the results were lower, around 4. No single score is given to evaluate the whole system. Some similar experiment was done in Siniakov (2006), except this time it was used on words in German text, and general text rather than a dictionary is used. This way graphs were made for the words in every sentence of a large corpus.

3.2 Automatic Acquisition of Hyponyms and Synonyms from a Large Corpus

Although many hypernym relationships can be found in other lexical resources, Ontologies and dictionaries these may not always cover less frequent, domain specific, hypernym relationships. Mining relationships from general text is most difficult since less is known about the structure of the text. Relationships between terms can be extracted from text through use of various patterns, and co-occurring terms that can often indicate a certain kind of relationship between terms.

In the past research on discovering hyponyms and synonyms from a large corpus has followed two main methods. The first is to discover synonyms, or near synonyms by finding terms that appear in similar contexts throughout text. The second method is to identify and then search for patterns in text that are known to indicate particular relationships. This method is usually applied to a large corpus or on the Web.

3.2.1 Determining Synonymy Through Similar Contexts

Detecting near-synonymy through discovering words with similar statistical distributions has been examined in (Turney, 2001). In this paper PMI-IR (Pointwise Mutual Information – Information Retrieval) was tested as a method of determining the semantic similarity of two terms. PMI-IR is a method of determining the semantic relatedness between two terms using a large corpus. This is done with the assumption that terms that frequently co-occur in text tend to be semantically related. PMI-IR works by taking a term and a number of candidates for synonymy with that term, and then generates a

score for each candidate. The candidate for synonymy with the highest score was selected as the synonym. Several different variations on PMI-IR were tested. One of the simplest is:

$$score(choice) = hits(problem NEAR choice)/hits(choice)$$

problem is the term for which one is trying to find synonyms and *choice* is a candidate for synonymy. The AltaVista search engine provided the NEAR operator. NEAR returns documents in which the two terms are within a range of 20 terms of each other. This method was tested on the TOEFL (Landauer and Dumais, 1997) (Test of English as a Foreign Language) synonym questions, as well as a collection of tests for ESL (Turney, 2001) (English as a Second Language). The results of using PMI-IR on this dataset were evaluated against using Latent Semantic Analysis (LSA) (Deerwester et al., 1990). PMI-IR received 74% on the dataset, while LSA received only 64%. Part of the reason that PMI-IR scored better than LSA is that it allows for much more text to be used than LSA can, due to its high computational time. Still, this shows the advantage of using PMI-IR on discovering near synonyms. The problem with this method of synonym discovery is that it does not distinguish between synonyms and close hypernyms/hyponyms, coordinate terms and other related words. One still does not know what sort of relationships exist between the near-synonyms. Another problem is that one must already have candidates for synonymy.

In Senellart and Blondel (2003) another method of discovering synonyms (or at least near synonyms) from a large corpora is described. This method described is based on a document vector space model. For every term a vector is created where each document in a collection is one dimension in the vector. The cosine similarity function is used to determine the distance between vectors. Other similarity measures can also be employed to discover the distance between two vectors. If two vectors are very close together then the two terms are considered to be semantically similar. Accuracies for this technique are not reported.

A method for merging several web-based methods of determining synonyms was attempted in Turney et al. (2003). Four different modules are used. The first one was LSA, done using a web interface developed at the University of Colorado. The second was PMI-IR, as described in Turney (2001). The third method was to use the online thesaurus Wordsmith as a source from which synonyms are extracted. Each module gives its own similarity score. The fourth and last module used Googles summaries and measures how often the pair of terms appears beside each other, or separated by one of a list of symbols and stop words. These four modules were combined in three different

ways. The first way was called the mixture rule. The mixture rule basically takes the weighted sum of all four modules. The second rule was the logarithmic rule, which works by combining the logarithm of all four modules. The last method was called the product rule. The product rule takes the weighted product of the modules. The product rule scored particularly high on TOEFL data, getting a total of 97.5%. This method of discovering synonyms does not really introduce anything new but rather shows that mixing several techniques for synonym identification can work better than any technique on its own.

These methods of discovering synonyms seems to be quite effective if one has a list of candidate synonyms, from which it is known that only one is the correct synonym. This cannot be directly applied to our problem of identifying relationships to be imported to Roget's Thesaurus. Within Roget's thesaurus it is possible that a term will have several synonyms, or even no synonyms. Also these techniques for discovering synonyms actually discover semantically similar terms, some of which will not necessarily be synonyms. For example it is possible that these techniques will identify hypernyms or hyponyms of a given term, and there would be no way to tell that they are not synonyms.

3.2.2 Using Patterns to Mine Relationships from a Corpus

Patterns found in text can be used to extract information from that text. Using patterns to extract hypernyms has been fairly well covered in literature. Identifying patterns in text has also been used for mining Meronyms, and determining semantic classes for words.

Hypernymy

Using known patterns of words to discover hyponyms in text was proposed in Hearst (1992). Hearst shows that patterns like: "NP_0 *such as* $\{NP_1$*,* NP_2*, ...,* $(and \mid or)\}$ NP_n", can be used to discover relationships where NP_0 is a noun and NP_1, NP_2, and, NP_n are hyponyms of NP_0. This relationship is reflexive and transitive but not symmetric. Six different patterns were used in this experiment, three being manually created upon observation, and another three being found by using bootstrapping methods, where phrases are retrieved where two terms, known to be semantically related are found. Some manual effort is put into writing the rules in a way that generalizes them properly. As such this process is only semi-automatic. In addition to the pattern shown above these patterns are also used:

- *such NP as {NP, }* {(and | or)} NP*
- *NP {NP, }* or other NP*
- *NP {NP, }* and other NP*
- *NP {,} including {NP, }* {(and | or)} NP*
- *NP {,} especially {NP, }* {(and | or)} NP*

Text from the Grolier encyclopedia was used as the source of data. Using the "such as" phrase 152 relationship were found in Grolier encyclopedia (Grolier, 1990). Of the 226 words found in the relationships 180 were also present in WordNet. A total of 106 relationships had both terms present in WordNet, with 61 of those relationships already present in WordNet (Hearst, 1992). This gives an accuracy of about 57%. Since construction of WordNet is an ever changing resource, it cannot be expected that the early version used by Hearst (1992) contains all the correct hypernyms for the English language. As such it is quite possible that some of the relationships extracted from Grolier were correct but were not, at the time, present in WordNet. This means that 57% of the relationships found are confirmed hypernyms while the others are still potential hypernyms. As such 57% can be thought of as a lower bound on the possible accuracy of the pattern.

Experiments have also been done in searching the web for specific hypernyms and hyponyms of a given term in (Sombatsrisomboon et al., 2003). Hyponyms of a term were found by searching the web for "** is a/an term*". Hypernyms were found by searching for "*term is a/an **". Google's web API was used to find these phrases on the web. Results are examined for several sample query terms. The possible set of hypernyms is ranked by frequency of occurrence. For example, "Java" has the hypernyms "Programming language" and "language" as the most frequent hypernyms, while things like "trademark" and "interpreted language" were much less frequent. A total accuracy for this method is not shown, however the authors do admit that this method fails for more general nouns, such as "student" or "animal". Also since each term would have to be queried individually this could be a very slow process, especially given that Google restricts the number of queries per day executed through its API. It is also pointed out that the patterns in Hearst (1992) could be used here, however since the web is so enormous that having just one query will often return a large number of relationships (Sombatsrisomboon et al., 2003).

In Caraballo (1999) an attempt was made to create a complete noun hierarchy. A bottom-up clustering method was used to create an initial hierarchy, where similar terms are found in the leaf nodes. Every noun was represented by a vector of terms that appear around the noun. These vectors were used to measure the similarity between two nouns. The closest two nouns were clustered together and form a new group. This was repeated until every noun was linked to every other noun through a tree-cluster. The method described in Hearst (1992) was used to extract hypernyms. Hypernyms were assigned to internal nodes in the tree, based on how often the term occurs as a hypernym of the leaf nodes under that internal node. Some compression was done, where if an internal node did not have any hypernym or has the same hypernyms as its parent nodes it is removed and its children become children of its parent node. This method successfully constructed a single hypernym tree, however it was found that when evaluated with strict criteria it was only about 33% accurate. When evaluated with loose criteria it was around 60% accurate at assigning hyponyms.

There are methods of automatically discovering the patterns that indicate hypernyms or hyponyms (Morin and Jacquemin, 1999). This method basically works by taking pairs of terms with a hypernym relationships and searching for those terms in order to discover patterns of text that they appear in.

An attempt has been made to apply supervised Machine Learning to the task of Hypernym extraction (Snow et al., 2005). My description of this method in this chapter will be brief as it is described in Chapter 5 where I experiment with several modified versions of it. The method starts by building a labeled corpus of positive and negative hypernym examples. This was done by first parsing 6 million sentences of news wire articles using Minipar (Lin, 1998a). From each parsed sentence pairs of nouns were selected. WordNet was then used to label the pair of nouns as a known hypernym pair, a known non-hypernym pair, or unknown. From the dependency tree created by Minipar, dependency paths between the nouns for all known hypernym pairs and known non-hypernym pairs were constructed. These dependency paths are similar to the ones used in Lin and Pantel (2001), where they were used for identifying relationships for a question answering system. Snow et al. (2005) then used these dependency paths as features for a machine learning algorithm. Under-sampling of the non-hypernym class was done so there was a 50:1 ratio of non-hypernym to hypernym pairs. Features were binary though they could also be placed into one of 14 redundant threshold buckets spaced at exponentially increasing intervals. For redundant threshold buckets, a dependency path has 14 features, each of which indicates that the number of times that dependency path

was found is past a particular threshold. A bucket for a pattern p at threshold t is set to 1 if the pattern is observed more than t times (Snow et al., 2005). Three Machine Learning algorithms were tested, Multinomial Naïve Bayes, Complement Naïve Bayes and Logistic Regression. The best system was found to be Logistic Regression, where buckets are used. An F-measure of 0.348 was achieved. Although this is not extremely high, the system could be designed to accept high precision at the cost of low recall. Since recall cannot be measured in any other hypernym mining system it is hard to tell what level of recall can be expected from other methods. It may not be fair to use buckets to count the number of times a feature appears since it is dependent on the size of the corpus. In different corpora the number of times a feature appears will not be consistent.

Hypernym relationships have been mined in languages other than English. In (Rydin, 2002) a hypernym hierarchy was constructed in Swedish using newspaper text as a corpus. Five patterns based on variations on Hearst (1992) patterns are used to mine relationships from the corpus. New relationships are found through a process where frequently occurring hypernym/hyponym pairs were taken and used in a new query to find new hypernyms. The query is as follows:

$$NP_h \text{ (funcword)+ } NP_2 \text{ (}NP_n\text{,)* (and | or) } NP_1$$

(funcword)+ is one or more function words, NP_h is the hypernym term from the original pair, and NP_2 *(*NP_n*,)* (and | or)* NP_1 is a conjunction of noun phrases, one of which is a known hyponym of NP_h. The total number of ordered hypernym pairs extracted was 28,133. 1000 pairs were selected and tested by 4 judges who rated their accuracies anywhere between 52.2% and 76.6% accurate. Kappa was calculated to be 0.51 for these judges.

A slightly different method of mining hypernym relationships from Japanese text was explored in Shinzato and Torisawa (2004). In this paper hypernyms were mined from lists on the web. This method works with three assumptions.

1. Expressions in the same itemization, or listings in HTML will likely have a common hypernym.

2. A hypernym and its hyponym will often appear in the same documents.

3. A hyponym and its hypernym are semantically similar.

Lists were extracted from web pages and their contents are assumed to be hyponyms. A large and random set of web pages G was downloaded. The set of hyponyms was used

to select a subset *LD* of the web pages in *G*, in which they appear. From *LD* a formula was applied to select terms that appear frequently in the subset, but infrequently in other documents in *G*. The list of potential hypernyms will have a lot of errors in it, since any words closely associated with the hyponyms from the list will appear. For example, if one has a list containing *Toyota, Ford, Chrysler* then words like *price* and *door* as well as *car* will appear. All potential hypernym candidates were then ranked for semantic similarity to the hyponyms. This was done by parsing all the documents in *LD* and then figuring out which potential hypernyms frequently occur in the same context as the words in the hyponym set. Several other heuristic rules were then applied. This resulted in a ranked set of hypernym/hyponym pairs. The results were shown where different thresholds were used for accepting the hypernym/hyponym pairs. When 500 pairs were accepted this system was over 80% accurate. When 1500 pairs were accepted it was closer to 60% accurate (Shinzato and Torisawa, 2004). This is a relatively successful system when there is a high threshold, however since it was done entirely with Japanese documents it is not clear if this will work as well for English.

Synonymy and Coordinate Terms

A method of obtaining synonyms from a large corpus or the web is examined in Ruiz-Casado et al. (2005). A method known as context overlapping is described for determining semantic similarity between two terms. Context Overlapping is done by collecting a set *S1* of all sentences that contain the word *w1*, and then collecting the set of sentences *S2* that contain the word *w2*. Next the percentage of sentences from *S1* in which replacing *w1* with *w2* will create a sentence in *S2* is calculated. For this method to work properly it will be necessary to limit the overlap to only several surrounding terms rather than the entire sentence, also it will be necessary to use an extremely large corpus such as the Internet. This can be used to determine if two terms are synonymous. If two terms are actually synonyms then they will be interchangeable in a lot of sentences. This method did well on TOEFLs synonym test, getting 82.5%. This method may not always detect just synonyms. There is the possibility that Context Overlapping will score highly for a term and its hyponym or another related term.

In Snow et al. (2005) coordinate terms were used to expand existing hypernyms. Two methods were examined for discovering coordinate term relationships in text. In the first method a distributed vector space model is created for the most frequent 163,198 occurring nouns using Minipar dependency links as features. A normalized similarity score is used to determine which nouns were coordinate terms. A second technique for

finding coordinate terms is to simply find conjunctions in text. For example "X, Y and Z" are all connected through conjunctions that can be detected using Minipar. The vector space model and the conjunction pattern methods scored an F-measure of .33 and .29 respectively. It shows the potential for patterns to be used for identifying coordinate terms, although it shows that a distributed vector space model works a bit better (Snow et al., 2005).

Semantic Word Classes

Similar work to Shinzato and Torisawa (2004) was proposed in Shinzato and Torisawa (2005) to discover semantic word classes. Words and phrases from itemizations in HTML were extracted. The words from each list were then paired with another word from the list, and pairwise mutual information is computed for each pair, using a search engine. From the set of pairs, 21 features were then extracted including the sum and average of all Pairwise Mutual Information scores, as well as other features related to the number of documents retrieved by the search engine to compute the Pairwise Mutual Information scores. These features were then used to train Support Vector Machines (SVM) and the results were ranked based on the output of the SVM decision function. A training set of 400 samples was created and tested on a test set of 800 samples. Since the SVM ranks the results, rather than assigning boolean yes/no values, it is possible to rank precision for different numbers of samples. A group was ranked as correct if 70% of its members had a common hypernym that does not cover an extremely wide range of objects (e.g. *object* or *thing* would not be acceptable hypernyms). This method was found to be fairly good for a small number of classes. It had 88% accuracy for the top 100 classes, and close to 80% for the top 200 classes (Shinzato and Torisawa, 2005). Although this work is also done in Japanese, it does show a potential method for discovering new classes of objects.

Some work was done on labeling semantic classes in Pantel and Ravichandran (2004). It was assumed that a semantic class has already been created. Feature vectors represented words, where each feature corresponds to a count of the context in which the word appears. The context was another term and the relationship to that term as determined by Minipar. A mutual information vector was then computed for each word, between itself and each context. Cosine similarity was used to determine similarities between terms. Different instances of each word were clustered together using group-average clustering. Scores for each words similarity to the other words in the class was computed and those words with the highest scores are selected as members of a committee. The

feature vectors of the members of the committee were averaged together and the word from the feature vector that has the highest score was chosen as the title for the semantic word class. To evaluate the work three judges were selected and classified 125 samples as either *correct*, *partially correct* or *incorrect*. The mean reciprocal rank (MRR) was then computed. Overall an accuracy of 77.1% was achieved. Kappa was also computed for this task and was found to be 0.72 (Pantel and Ravichandran, 2004). This work is similar to extracting hypernym/hyponym pairs from text, however it requires that a semantic class already be constructed.

Meronymy

In addition to hypernymy, there are methods of extracting meronym relationships from text. Some of these methods use similar methods to those of Hearst (1992).

A method similar to that in Hearst (1992), was used for obtaining meronym and holonym relationships in Berland and Charniak (1999). Patterns for finding meronym relationships were found by searching for pairs of terms that are related by meronymy. Once the set of patterns was identified they are searched for in a 100,000,000-word corpus to retrieve the meronym relationships. A series of metrics were used to rank the results. Although I will not be searching for meronyms or holonyms, it is interesting to note that the methods similar to Hearst (1992) can be used to discover relationships other than noun hypernyms and hyponyms.

Other work on mining meronyms is done in Girju et al. (2003) and Girju et al. (2006). In these papers a method of retrieving part-whole relationships from text was explored. In Girju et al. (2003) three patterns from text are used for meronym extraction. Two work at the phrase level:

- NP_x of NP_y
- NP_y's NP_x

One works at the sentence level:

- NP_1 Verb NP_2

In Girju et al. (2006) more patterns including noun compounds and prepositions are used as well as those above

- NP_{xy} (compound e.g. "door *knob*")

- NP_x PP_y
- NP_y have NP_x

These patterns can identify potential meronym/holonym pairs. Machine Learning was used to identify a series of semantic constraints that must be met for two noun phrases to be in a part-whole relationship. To build a training and testing corpus SemCor (Miller et al., 1993) and part of the TREC 9 (Voorhees and Harman, 2000) corpus made up of LA Times articles are used. A word sense disambiguation program was used to find WordNet senses of words in the LA Times articles. True and false hypernym pairs were extracted from the corpora and used as training data. For each pair of words the hypernym hierarchy was used to determine what "kinds" of objects can be in a part-whole relationship. For example if one finds that a *hand* is part of a *woman* then the system may come up with a rule where a *part* is part of a *causal agent*. Likewise if one finds that an *apartment* is not part of a *woman* then a rule *whole* is not part of a*causal agent*. Note that, *hand*, *apartment* and *woman* are all *entities*, however any rule saying an *entity* (is | is not) part of an *entity* would be ambiguous. These rules were refined so as to eliminate ambiguous examples as much as possible. The machine learning algorithm C4.5 (Quinlan, 1993) was used to create the constraints. On the Wall Street Journal articles a precision of 83% precision and recall of 79% were obtained in Girju et al. (2006).

It would be difficult to use the Girju et al. (2006) method of meronym discovery on on Roget's Thesaurus. The reason for this is that a good system for disambiguating words into Rogets Thesaurus word senses needs to be developed. Even if one is developed the lack of a corpus with Roget's Thesaurus sense tags make it difficult to verify that the method works. Also the proposed method will only work properly for terms that already exist in a hypernym hierarchy. Roget's Thesaurus has a partial hierarchy already constructed. New hypernyms will be added to the thesaurus from this thesis, though it is not clear that these imported hypernyms will work as well as those present in WordNet.

3.2.3 Bootstrapping for Patterns

Bootstrapping patterns is basically the process of taking a few positive examples of a relationship and then using those examples to discover patterns to represent the relationship. These patterns can then be used to find more positive examples, which in turn can be used to find more patterns. Bootstrapping is a recursive process. Techniques similar

to this have been used quite frequently in Information Extraction (Geleijnse and Korst, 2006; McLernon and Kushmerich, 2006; Surdeanu et al., 2006; Tomita et al., 2006; Pantel and Pennacchiotti, 2006; Chen et al., 2006; Suchanek et al., 2006; Turney, 2006b). These systems have been used for many kinds of relationships, including sometimes hypernymy.

In Geleijnse and Korst (2006) pairs of words were used to find Hearst (1992) like patterns for recognizing different kinds of relationships. Words known to have a particular kind of relationship were searched for on the web using Google, the patterns that are found for these words are then used to find patterns. It was also shown how this can be used for question answering.

When using bootstrapping it is beneficial to find some method of determining which patterns are good and which are bad. In McLernon and Kushmerich (2006) a bootstrapping approach to discovering patterns is described. The algorithm finds patterns in an annotated text and then uses these patterns to find new training examples, which are in turn used to find new patterns. A function for rating the patterns was also devised. More research done on bootstrapping methods is proposed in Surdeanu et al. (2006). This paper focuses a bit more on techniques for determining which patterns are best at each iteration of the bootstrapping.

One of the main advantages of bootstrapping for new patters is that it can increase the recall of a system quite significantly. Tomita et al. (2006) tested two different bootstrapping systems for Information Extraction. The first was simply to find pairs of words in text and then use adjacent terms to determine patterns for identifying other word pairs with the same relationship. The second method used a bag of words approach using neighboring terms. These two methods were tested on a corpus for "Mayor of" relationships and were found to increase recall on a test set by 250%, while precision was 0.87. It was also tested for several other relationships, all of which shows improvements for recall.

A system called *LEILA* was proposed in Suchanek et al. (2006). It works by collecting samples from the text that are labeled either positively as an *example*. If some relationship is found to be contrary to a *example* it is a *counter example*. If a relationship is neither an *example* or *counter example* it is a *candidate*. The pair can also be labeled as none of the above. *LEILA* was tested on three different kinds of relationships: birth-dates, synonymy, and instance-of relationships. Next they use statistical learning to discover positive patterns for each relationship and test these words on text. Other systems like *Espresso*, developed by Pantel and Pennacchiotti (2006) have been created to learn patterns for recognizing relationships in text. *Espresso* works by taking only a

few positive examples of a relationship and a large text, and finds patterns that can be used to discover new instances of the relationship.

Turney (2006b) proposes a method of representing relationships with text patterns. It works much the same as the other methods described in this section, in that it takes a pair of words and searches for patterns that appear between those words. Patterns which best represent a particular relationship are selected. This system is tested on a set of 374 SAT analogy questions, where a pair of related words are given, and from a set of four pairs of words the correct analogy must be selected. Some of the questions could not be answered because pairs of words were not found close together in the corpus. 55.7% accuracy was found on those questions answered and 53.5% accuracy was found on the whole data set.

Parse trees have been used in different ways for identifying a variety of semantic relations. Zhang et al. (2006) used features from parse trees to represent semantic relationships. Two words were taken, from a parsed sentence, and all the non-terminals between the words in a parse tree are taken as features. SVM's were trained and tested to find relationships in a corpus. A corpus from The Automatic Content Extraction Projects, called ACE was used (ACE, 2004). ACE contains 5 main kinds of relationships, with 24 different subtypes.

Some work has also been done on inferring new relationships from other relationships (Culotta et al., 2006). For example if X has a father Y, who has a wife Z then X has a mother Z. This means that the relationship *mother* is the same as following the two relationships $father -> wife$. This was done using a graph of people and organizations, where these people/organizations are linked together by relationships with each other. These relationships can be found in places where there are two paths between the same people/organizations in the graph.

3.2.4 Relationships with Named Entities

A similar problem to identifying hypernyms in text is labeling named entities for Ontologies. In Tanev and Magnini (2006) a weakly supervised method of Ontology populating was described. Several different classes of objects such as mountains, cities, athletes and inventors were selected. This method extracted patterns from a parsed corpus. These patterns were used as features for training a Machine Learning algorithm. The supervised method was compared against two other unsupervised methods, where it was found that the supervised method worked better. This task is very similar to hypernym extraction

from text although this focuses on named entities rather than more abstract concepts that would appear closer to the root of a hypernym hierarchy.

In Bunescu and Pasça (2006) another method of disambiguating name entities in Encyclopedias was described. Name entities from Wikipedia were selected and SVM Light (Joachims, 1999) was used to create a classifier to identify which categories from Wikipedia the name entity belonged to. These name entities were limited to subcategories of people.

3.2.5 Refining the Relationships

The methods described above for finding hypernyms/hyponyms often are not extremely accurate. There has been quite a bit of work done on finding ways to improve precision of the hypernyms/hyponyms retrieved. One such method uses Latent Semantic Analysis (LSA) to help determine if two terms are semantically similar enough to be hypernyms/hyponyms (Cederberg and Widdows, 2003). If two words appear regularly enough in the same context then they are more likely to be related by hypernymy or hyponymy. If the terms do not frequently appear in the same context then they are most likely not related and can be removed from the set of relationships. This method will improve the precision of the relationships found. A method for improving recall was also proposed in Cederberg and Widdows (2003). Lists of words in the form of "y_1, y_2 and y_3" were found and assumed to be coordinate terms. If one of those terms has a known hypernym, then all of the coordinate terms are assigned the same hypernym.

Another method was proposed by Caraballo and Charniak (1999) where the specificity of a noun is determined to decide, from a pair of nouns, which one is more likely to be the hypernym and which the hyponym. Specificity is a measure of how general a term is. This was done under the assumption that a term is more specific than its hypernyms. The specificity of a common noun can be determined by how often it is preceded by a modifier. An entropy-based function was used to decide if a term is more specific or less specific. This method can help to verify if a term is actually a hypernym or hyponym of another term depending on how specific each term is.

Methods of ranking meronym relationships were examined in Berland and Charniak (1999). Although the methods were designed for ranking meronyms it is likely that they could also be used for ranking hypernym/hyponym. Two systems were tested for ranking. In the first method log likelihood was used to rank the pair by the likelihood of a *whole* and its *part* appearing together versus the *whole* appearing on its own. This

is done using the log-likelihood metric proposed by Dunning (1993), which builds on the likelihood ratio. The likelihood ratio is the maximum likelihood of a particular hypothesis, divided by the maximum likelihood of the entire parameter space:

$$\lambda = \frac{max_{\omega \in \Omega_0} H(\omega, k)}{max_{\omega \in \Omega} H(\omega, k)}$$

Where Ω is the entire parameter space and Ω_0 is the hypothesis space being tested. ω is a point in the parameter space, and k is a point in the observation space. $H(\omega, k)$ is the likelihood function, giving the probability of an experimental outcome k given a parameter ω. The second measure ranked the pairs based on how far apart the distributions of the *whole* with its *part* are versus just the *whole* at different significance levels.

A technique of automatically inferring taxonomies has also been discussed in Snow et al. (2006). This technique works by mining hypernyms and coordinate terms for a word X in a corpus. The method of mining hypernyms is the same as in Snow et al. (2005). Each hypernym of X was considered as a possible parent to X. The correct parent was chosen by finding out how many hypernyms and how many coordinate terms found for X in text are also found in a given location in WordNet. A threshold is set to determine where a word should be placed. WordNet without these newly added hypernym relationships were then tested against WordNet without the new links at identifying hypernym pairs from a manually labeled corpus. After adding 30,000 new relationships it was found that the F-measure was increased by 23% over WordNet.

The idea of enhancing WordNet with new relationships was also explored in Pennacchiotti and Pantel (2006). Two words, X and Y were given, along with a relationship R. Two disambiguation methods are proposed. The first method took X and then found all words related to X by R and found which instance of X was most closely related to those words in WordNet (by counting the edges in the shortest path between the words). The sense of X which was related to the most words by relationship R was the chosen word sense. It then did the same thing for Y. The second approach was a clustering approach, it starts by generalizing the words in a set of relationships. For all relationships R, the hypernyms of the X's and Y's are found. These hypernyms are the clusters and a distance function is used to determine which cluster each term was assigned to. *Espresso* (Pantel and Pennacchiotti, 2006) was used to mine several different kinds of relationships. For both systems the relationship can be assigned to more than one sense of each word. Rather than taking only the best sense, all senses above some threshold were chosen. Tests were done on *part-of* relationships and *causation* relationships. For *part-of* the clustering approach worked best, for *causation* the first method worked

better, though only narrowly.

3.3 Conclusions

Mining hypernyms from dictionaries as well as raw text has been accomplished with varying degrees of accuracy. Many of the most accurate methods rely on dictionaries or work best on specific kinds of terms. Other methods use patterns to extract relationships from general text, though are not quite as accurate.

Two main methods of evaluation can be seen too. One method is to get humans to evaluate samples of the relationships. The other is to use the new resource to solve some applications and use the results to evaluate it against other resources like in Jarmasz and Szpakowicz (2003b,a) described in Chapter 2.

Chapter 4

Analysis of Roget's Thesaurus

In this chapter I examine Roget's Thesaurus for the types of relationships contained in the thesaurus and where these relationships can be found. This will include an examination of the hierarchy present in Roget's Thesaurus. An examination has been done using both WordNet and manual evaluation. I found that most semicolon groups, while containing semantically closely related terms did not contain synonyms. The relationships between terms in the semicolon group are likely to be closer to the WordNet coordinate terms relationship. WordNet is used to determine which relationships were most frequent in Roget's Thesaurus. Hypernymy/hyponymy are found to be the most frequent relationship in the POS, Paragraph and Semicolon group levels of granularity.

4.1 Manual Evaluation of Roget's Thesaurus

I examined a random selection of about 1% of all noun and verb paragraphs from Roget's Thesaurus and labeled the kinds of relationships found in each semicolon group as one of 6 relationships and unknown/other. The relationships are: synonymy, hyponymy, meronymy, antonymy, coordination (the equivalent of WordNet's coordinate terms) and causality. Synonym, hyponym, meronym and antonym are all fairly obvious. Coordination for verbs and nouns, are any terms that probably share a hypernym but are not synonyms. They can be thought of as near synonyms other than hypernyms, hypernyms and strict synonyms. The causal relationship means that one thing creates/produces/enables the other. Some semicolon groups were labeled as unknown if the words in the semicolon group either had nothing in common, or had a relationship other than those listed above. Obviously there can be many different relationships between

Synonym	35.2%
Hyponym	55.8%
Coordinate Terms	57.3%
Meronym	1.9%
Antonym	0.4%
Causal	1.9%
Unknown	3.3%

Table 4.1: Percentage of semicolon groups in Roget's Thesaurus that contain a particular relationship (Nouns and Verbs)

Synonym	29.9%
Hyponym	55.5%
Coordinate Terms	56.0%
Meronym	2.7%
Antonym	0.3%
Causal	1.6%
Unknown	4.7%

Table 4.2: Percentage of semicolon groups in Roget's Thesaurus that contain a particular relationship (only Nouns)

the terms in a semicolon group. Because of this often semicolon groups will receive more than one relationship label. An interested reader can find the samples of Semicolon Groups with their assigned labels at `http://www.site.uottawa.ca/~akennedy/mastersThesis/supplement_A.pdf`.

In all coordinate terms are the most common kind of relationship found. Hyponyms and Synonyms are second and third most common, small numbers of the rest were found. Their percentages are listed for nouns and verbs in Table 4.1, and for just nouns in Table 4.2. Since more than one kind of relationship can be found in each semicolon group the total percentage will add up to more than 100% in each table. Often a single semicolon groups will have synonym, hyponym and coordinate term relationships.

Clearly Synonyms, Hyponyms and Coordinate Terms are by far the most frequent relationships. Although antonyms and meronyms do exist they are not very frequent. The causal relationship is also fairly rare. An example of a causal relationship can be

Relationship	Precision	Recall
Nouns and Verbs		
Synonymy	63%	37%
Hyponymy	79%	27%
Coordination	85%	29%
Nouns only		
Synonymy	62%	53%
Hyponymy	81%	36%
Coordination	85%	35%

Table 4.3: WordNet's agreement with my classification of Tables 4.1 and 4.2.

seen in the following semicolon group:

barristership, advocacy, pleading;

In this case *barristership* is the ability to practice law as an advocate, particularly in a higher court. Advocacy and pleading are two of the jobs of a barrister. Thus one's barristership enables one to perform *advocacy* and *pleading* in a higher court of law.

Since a single person was used to identify these relationships a method is needed to verify that the relationships identified in the semicolon groups are actually correct. To do this I used WordNet to verify the three most common relationships, those being synonymy, hypernymy and coordination. WordNet does not find a relationship in every semicolon group, and because of this a low recall score can be expected. The precision and recall of WordNet measured against my classification from Tables 4.1 and 4.2 can be found in Table 4.3. From this table it can be seen that WordNet does not find either synonymy, hyponymy or coordination relationships in many of the semicolon groups I did, indicated by low recall. However, when WordNet does find a relationship in the semicolon groups I usually also found that relationship in the semicolon group, indicated by relatively high precision.

One reason why the precision is not higher is because sometimes it is not completely clear if two terms should have a synonym or a hyponym relationship. For example the following semicolon group appears in Rogets Thesaurus:

prevention, veto, ban, bar, embargo, prohibition ;

In WordNet *ban* is both a hyponym and a synonym of *prohibition* depending on the WordNet word sense. Another example is:

prisoner at the bar, defendant, accused, accused person;

In WordNet *accused* is a hyponym of *defendant* however these could easily be synonyms.

4.2 Identifying Types of Relationships using WordNet

In Section 4.1 I wrote how relationships are manually evaluated, and WordNet is used to attempt to verify the results found. This section looks at what can be learned about Roget's Thesaurus by comparing it directly with WordNet.

4.2.1 WordNet Relationships at Varying Levels of Granularity

Knowing what kinds of relationships appear frequently in Roget's Thesaurus is very important to creating a more useful lexical resource. Part of this involves studying what relationships can be captured at different levels of granularity. To do this I use the relationships from WordNet, and found how many of them mapped to Roget's Thesaurus at the different levels of granularity. See Chapter 2.1 of this thesis for a description of Roget's Thesaurus' general organization.

Four levels of granularity are considered. The broadest level, is across the whole thesaurus. This means that relationships can exist between any pair of terms in the thesaurus. This is done largely for reference sake since it would be very difficult to determine which word senses in Roget's Thesaurus should be assigned to a relationship. The second level considered is the Part of Speech (POS) level. This level is all words of a particular part of speech within a given Head. The Head is not used since I focused on relationships that go between words of the same part of speech. The third level is the Paragraph, and the last level is the Semicolon Group. Other possible levels of granularity are Head Group, Sub-Section, Section and Class. These levels are not considered because in order to assign a relationship between two different Heads some sort of word sense disambiguation would have to be used. Relationships are tested across the entire thesaurus because it shows the total possible number of relationships that could be imported into Roget's Thesaurus. This is only for the sake of comparison since it suffers from the same problem that word sense disambiguation would be needed.

Within Roget's Thesaurus the words do not contain definitions so word sense disambiguation can be quite difficult, even for people. The same word never appears twice in the same semicolon group, and only on extremely rare occasions does it appear twice in

- `heathen, non-Christian, pagan, paynim`
- `heathen, pagan, infidel`

Figure 4.1: Two semicolon groups from two paragraphs in Head 974: Irreligion

- `rara avis, mythical beast, unicorn, phoenix, griffin, simurg, roc`
- `animal charge, lion, lion rampant, lion couchant, unicorn, griffin, cockatrice, eagle, falcon, martlet`

Figure 4.2: Two semicolon groups from two paragraphs in Head 84: Nonconformity and Head 547: Indication respectively

the same paragraph. Occasionally a word will appear twice in the same POS. Since the words do not contain any sort of definitions with them, it can be very difficult to distinguish what sense of a word is being used. In cases where the same word appears twice in the same Paragraph, or POS it may actually be impossible to distinguish a difference between the word senses. If a WordNet relationship is found between two words at a given level of granularity, then I assume that the word senses from WordNet, correspond to the word senses of the words in Roget's Thesaurus. For example, in Figure 4.1 are two semicolon groups from two different paragraphs in the same Head that both contain the word "heathen". In Figure 4.2 one can see two appearances of the word "griffin" from two different heads. In both cases the senses of the words "heathen" and "griffin" would appear to be very similar. It could be difficult to tell what differences, if any, there are between the senses.

My first experiment examines mapping of synonyms from WordNet into Roget's Thesaurus. The total number of relationships found, and the number of unique relationships found can be seen in Table 4.4. The counts for relationships for Antonyms, Hypernyms, Hyponyms, Holonyms and Meronyms can be found in Tables, 4.5, 4.6, 4.7, 4.8 and 4.9 respectively.

Relationships are counted only once at each level of granularity. That is, if all relationships must appear in semicolon groups then each semicolon group can contain any given relationship only once. The same goes for Paragraphs and POSs. For example, if a word appears twice in the same POS and is related to another word in the POS, then there are two relationships that are the same, but the relationship will only be counted

once, not twice. As such, the number of relationships in the "Total" row may seem low since the same relationship may appear more than once, but is counted only once. There is also a slight discrepancy in the number of hypernyms, and hyponyms, as well as the number of holonyms and meronyms. This occurs because how relationships are extracted from Wordnet. The process works as follows:

- Choose a word X from Roget's Thesaurus
- Pick a type of relationship *rel* to search for.
- Search for related terms Y in WordNet
  - Lemmatize X: *lem(X)*
  - Find terms Y related to *lem(X)* by relationships *rel*
  - Return Y
- Map words from Y to Roget's Thesaurus.

Lemmatizing changes a word to the format that it appears as in WordNet. Note that X is lemmatized, however the terms in Y are not. Because of this, if *lem(X)* is a hypernym of Y, it does not imply that *lem(Y)* is a hyponym of X. As a result, the number of relationships found for hypernyms and hyponyms will differ slightly. This also affects the number of holonyms and meronyms. Since the words in Y are from WordNet, they are not lemmatized.

From Table 4.5 it can be seen that Antonyms are not good relationships to map to Roget's Thesaurus as most of these relationships exist between different Heads. Synonyms, Hypernyms/Hyponyms, and Meronyms/Holonyms all have better coverage in the thesaurus. For Nouns about 90% of all synonyms found at the POS level were still found in the Paragraph level, and 53% were found at the Semicolon Group level as can be seen in Table 4.4. For hypernyms/hyponyms about 80% of all relationships found at the POS level were found at the Paragraph level, though only about 27% were found at the Semicolon Group level as can be seen in Tables 4.6 and 4.7. In most cases the coverage of the relationships found in the same Paragraph were fairly close to those found in the same POS. Holonym/Meronyms (seen in Tables 4.8 and 4.9) were the worst where about 70% of all relationships found at the POS level were found at the Paragraph level as well. This seems to indicate that relationships can mostly be limited to be within the same Paragraph.

Synonym	Noun		Verb		Adverb		Adjective	
	Unique	*Total*	*Unique*	*Total*	*Unique*	*Total*	*Unique*	*Total*
Total	23761		19161		1352		11418	
POS	16937	49861	12271	40124	907	1681	8102	26438
Paragraph	15100	44408	10922	36122	858	1574	7353	23937
SG	9066	24968	5555	17872	515	921	4143	12511

Table 4.4: Count of synonyms mapped from WordNet to Roget's Thesaurus

Antonym	Noun		Verb		Adverb		Adjective	
	Unique	*Total*	*Unique*	*Total*	*Unique*	*Total*	*Unique*	*Total*
Total	1335		1996		251		1603	
POS	200	615	174	540	20	82	184	607
Paragraph	170	536	127	418	16	62	165	544
SG	60	166	35	136	11	38	76	244

Table 4.5: Count of antonyms mapped from WordNet to Roget's Thesaurus

Hypernym	Noun		Verb	
	Unique	*Total*	*Unique*	*Total*
Total	72161		42334	
POS	38823	57478	20695	41976
Paragraph	30624	45481	17014	35245
SG	10576	15106	5377	12204

Table 4.6: Count of hypernyms mapped from WordNet to Roget's Thesaurus

Hyponyms	Noun		Verb	
	Unique	*Total*	*Unique*	*Total*
Total	72518		42360	
POS	39418	58105	20704	41988
Paragraph	31100	45987	17019	35251
SG	10752	15293	5377	12204

Table 4.7: Count of hyponyms mapped from WordNet to Roget's Thesaurus

Holonyms	NOUN	
	Unique	*Total*
Total	5630	
POS	2607	3701
Paragraph	1856	2773
SG	804	1435

Table 4.8: Count of holonyms mapped from WordNet to Roget's Thesaurus

Meronyms	NOUN	
	Unique	*Total*
Total	5590	
POS	2619	3724
Paragraph	1867	2796
SG	810	1454

Table 4.9: Count of meronyms mapped from WordNet to Roget's Thesaurus

4.2.2 Semicolon Groups and Their Contents

Semicolon groups are the finest level of granularity in the Thesaurus, after the words/phrases themselves. Only allowing relationships to be within the same semicolon group may prove extremely limiting since relationships between semicolon groups within the same paragraph are not always clear. Table 4.10 shows the number of semicolon groups that contain at least one of the indicated relationships and the number of total semicolon groups. I examine synonyms, direct hypernyms and indirect hypernyms. Direct hypernyms are the immediate parents in the hypernym tree, while indirect hypernyms are grandparents, great grandparents and so on, in the hypernym tree.

Table 4.10 shows that most semicolon groups do not contain any of the relationships found in WordNet. Even so, synonyms and direct hypernyms are both relatively common, however indirect hypernyms were much less frequent, particularly for nouns. Since direct hypernyms are more frequent than indirect hypernyms this suggests that most of the Semicolon Groups mostly contain words that are directly related to each other somehow. In many cases in WordNet, a term and its direct hypernym could be considered synonyms. For example "defendant" is an immediate hypernym of "accused", even

SG	Noun	Verb	Adverb	Adjective
Semicolon Groups	*31132*	*13958*	*1581*	*12893*
Synonyms	7234	3950	241	2449
Direct Hypernyms	7352	4264	-	-
Indirect Hypernyms	2268	1803	-	-

Table 4.10: Kinds of relationships in the Semicolon Group

though these terms are virtual synonyms. A more distant hypernym for "defendant" would be "person".

4.2.3 Relationships Between Semicolon Groups

In order to discover if there is some order to the semicolon groups I looked for various relationships within each paragraph and then measured the distance between the two terms in the relationship. If they appear in the same semicolon group then their distance is 0, if they are in neighboring semicolon groups then their distance is 1, and so on. Table 4.11 shows the number of synonym relationships found with distances ranging from 0 to 20. Since different paragraphs will have different numbers of semicolon groups, I attempt to normalize the counts of these relationships. To do this I take the number of relationships and divide it by the total possible number of relationships at that distance between the semicolon groups. The total possible number would be where a given word is related to every word in the other semicolon group. For synonyms this can be seen in Table 4.11, and for hypernyms and holonyms it can be seen in Table 4.12.

When the distance between the semicolon groups is 0 (i.e. the same semicolon group) more relationships are shown in Table 4.11. It can also be seen that as the distance between semicolon groups increases the number of relationships found decreases. This alone does not necessarily mean that more similar terms tend to appear in closer semicolon groups since paragraphs do not all have the same number of semicolon groups. To compensate for this I try to normalize the data, as described above. When this is done it can be seen that the proportion of relationships found decreases as the distance increases. For verbs the proportion of relationships found between distances of 1 to 4 is somewhat higher than those 5 apart or further, where the proportion tends to level out. This seems to suggest that closer semicolon groups are slightly more likely to contain synonyms than semicolon groups that are farther apart. The same tests are done

SG Distance	Noun		Verb		Adverb		Adjective	
	Total	*Ratio*	*Total*	*Ratio*	*Total*	*Ratio*	*Total*	*Ratio*
0	24750	0.0187	17688	0.0533	919	0.0252	12452	0.043
1	6278	0.00651	5210	0.0215	277	0.0135	3834	0.0182
2	3282	0.00376	3038	0.0142	158	0.0112	2122	0.0115
3	2417	0.00307	2144	0.0114	74	0.00746	1314	0.00822
4	1519	0.00218	1690	0.0105	68	0.00899	1009	0.00747
5	1252	0.00213	1156	0.00839	36	0.00646	760	0.0067
6	1072	0.00199	982	0.00839	14	0.00334	505	0.00537
7	850	0.00184	720	0.00728	6	0.00204	366	0.00474
8	746	0.00192	698	0.0084	6	0.00282	306	0.00478
9	537	0.00174	550	0.00794	4	0.00248	286	0.00553
10	348	0.00127	414	0.00718	0	0	204	0.00496
11	331	0.0014	362	0.00774	0	0	177	0.00544
12	206	0.00114	318	0.00824	0	0	160	0.00622
13	204	0.00126	222	0.00701	2	0.00315	136	0.00689
14	130	0.000986	152	0.006	8	0.0147	79	0.00531
15	124	0.0011	142	0.00716	0	0	48	0.00425
16	66	0.000787	94	0.00601	0	0	60	0.00728
17	60	0.000904	130	0.0105	0	0	35	0.00574
18	74	0.00135	70	0.00726	0	0	36	0.00769
19	38	0.000676	62	0.00821	2	0.00778	16	0.00481
20	32	0.00183	62	0.0107	0	0	6	0.00238

Table 4.11: Distances between synonyms in Semicolon Groups

counting the relationships for hypernyms and holonyms at various distances in Table 4.12.

Although closer semicolon groups do seem to be more likely to contain hypernyms and holonyms, it is not as clear as for synonyms. Adjacent semicolon groups do seem to have a greater chance of having a hypernym or holonym relationship, however for verb hypernyms and noun holonyms this does not extend to semicolons of distance 2 or more from each other. In no case does the number of relationships found in adjacent semicolon groups come close to the number of relationships found within the same semicolon group.

Although this does not necessarily indicate a particularly strict ordering of the semicolon groups, it does suggest that at least neighboring semicolon groups are more likely to contain related terms than more distant semicolon groups within the same paragraph. It is difficult to take advantage of this finding. Although neighboring semicolon groups seem to have more relationships between each other than semicolon groups that are farther apart this does not help infer what those relationships are.

4.3 Conclusions about Relationships in Roget's Thesaurus

This chapter has shown that that the bulk of relationships that the synonym and hypernym pairs frequently occur within the same Paragraph in Roget's Thesaurus. Other relationships like Antonym and Meronym/Holonym do appear in the Thesaurus, however they are likely to go between different Heads. To map relationships between terms in different Heads would require an accurate method of word sense disambiguation. Most hypernym, meronym and synonym relationships found within a given head can also be found in the same paragraph.

Synonyms and near Hypernyms can be found frequently within the Semicolon group. This may suggest that if a hypernym for one term in a semicolon group is found then same hypernym relationship can be applied to the other terms in the semicolon group. This will be further tested in Chapters 5 and 6.

SG Distance	Hypernym: Noun		Hypernym: Verb		Holonym: Noun	
	Total	*Ratio*	*Total*	*Ratio*	*Total*	*Ratio*
0	14973	0.0219	12065	0.0363	1431	0.0021
1	7192	0.0143	5127	0.0211	317	0.000629
2	4514	0.00997	3025	0.0141	178	0.000393
3	3472	0.0086	2592	0.0138	174	0.000431
4	2744	0.00774	2002	0.0124	127	0.000358
5	2308	0.00746	1676	0.0122	136	0.00044
6	1956	0.00729	1428	0.0122	72	0.000268
7	1647	0.00718	1088	0.011	65	0.000283
8	1265	0.00652	1041	0.0125	74	0.000382
9	1001	0.0061	838	0.0121	43	0.000262
10	897	0.00652	849	0.0147	26	0.000189
11	664	0.00581	601	0.0129	24	0.00021
12	544	0.00573	524	0.0136	26	0.000274
13	458	0.00587	455	0.0144	11	0.000141
14	385	0.00603	379	0.015	15	0.000235
15	269	0.00517	323	0.0163	12	0.000231
16	214	0.0051	205	0.0131	13	0.00031
17	179	0.00539	202	0.0164	6	0.000181
18	165	0.00604	156	0.0162	7	0.000256
19	110	0.00496	115	0.0152	6	0.00027
20	116	0.00665	103	0.0177	0	0

Table 4.12: Distances between hypernym pairs and holonym pairs in Semicolon Groups

Chapter 5

Building the Resources

In this chapter the details of how relationships are discovered and imported into Roget's Thesaurus is described. The chapter outlines which resources are used as well as the techniques used to extract the relationships from them.

5.1 What Relationships will be Included

As described in Chapter 3 there are many methods of discovering relationships from different kinds of text as well as methods of inferring new relationships through known synonyms. I will focus on discovering hyponym/hypernym relationships to import into Roget's Thesaurus.

As can be seen in the previous chapter on examining Roget's Thesaurus there is a clear tendency for known hypernym relationships to fall within the same Roget's Paragraph. That is why the hypernym relationships that will be imported will be limited to being within the same Roget's Paragraph. There are two primary advantages of doing this. First of all, this allows for filtering of potentially incorrect relationships. Since a term and its hypernym are usually quite closely semantically related it is less likely that they will fall in different paragraphs than the same one. This will also filter out distant hypernym pairs, for example a relationship like "eraser *is a* thing" is a distant relationship since the pair does not appear to be very semantically similar. However, if one has a relationship "subject *is a* thing" it is much better since both will likely appear to be more semantically similar.

The second advantage of allowing relationships to only appear within a paragraph is that it eliminates the need for word sense disambiguation since even very polysemous

words do not appear twice in the same paragraph very often. If a hypernym pair is discovered by one of several means (to be described in the next section), the relationship can be imported directly into Roget's Thesaurus without the need for determining which senses of the words were being used.

5.2 Sources of Relationships

Roget's Thesaurus is extremely large containing over 50,000 unique nouns and noun phrases, many of which are not found in any given lexical resources or corpora. This means a variety of methods and resources are needed, to collect new hypernym relationships to add to the Thesaurus. A total of four different methods are applied to discovering new hypernym/hyponym relationships in text. These methods are:

- Direct extraction from existing resources
  - WordNet
  - OpenCyc
- Using patterns to extract hypernym relationships from specialized texts such as dictionaries
  - Longman Dictionary of Contemporary English (LDOCE)
  - Wiktionary definitions
- Extracting relationships ships from general text using techniques similar to (Hearst, 1992)
  - The British National Corpus (BNC)
  - The Waterloo Multi-Text Corpus
- Inferring new hypernym relationships by replacing terms with their synonyms
  - Roget's Thesaurus Semicolon Groups
  - Wiktionary synonyms

In addition to these four methods an an attempt is made to develop a Machine Learning approach based on Snow et al. (2005) but with several significant modifications. Several variations on this method are attempted. The experiments using Machine

Learning and inferring new relationships from synonymy were not as successful as the other methods listed above.

For all methods listed above all possible morphological forms of the terms in the hypernym/hyponym pair are considered when adding these relationships to Roget's Thesaurus. This has very little affect on hypernyms extracted from WordNet and OpenCyc, and to a lesser extent LDOCE and Wikitionary, however it is quite useful for hypernyms extracted from the BNC and the Waterloo Multi-Text Corpus where words are not regularly in their base form.

These morphological forms are obtained from the Roget's Thesaurus ELKB (Jarmasz and Szpakowicz, 2001a). The ELKB can be used to find all possible morphological forms of a word. This way if a search is done for a non-root word that does not appear in the Thesaurus, the word's root will be derived and searched for.

5.2.1 Mining Hypernyms Relationships from Existing Ontologies

Using existing lexical resources is a free and easy way to collect lexical relationships. In this case WordNet (Fellbaum, 1998b) and OpenCyc (Lenat, 1995) are both used. WordNet is a commonly used lexical resource in NLP. Hypernym relationships are extracted from WordNet and then added to Roget's Thesaurus. The hypernyms extracted can be any distance from each other in the hypernym tree. The only requirement is that both hypernym and hyponym be contained in the same Paragraph in Roget's Thesaurus. A total of 53,404 relationships are imported from WordNet to Roget's Thesaurus.

When retrieving relationships from WordNet, first a paragraph of terms/phrases is taken from Roget's Thesaurus. Then each word is taken on its own, and lemmatized to appear in a form that exists in WordNet. If the word exists all possible senses of that word are found, and all hypernyms of each word sense are collected. For each of these hypernyms if any word in its WordNet synset is found to be in the same Roget's Paragraph then those two words are recorded as hypernyms.

OpenCyc is a freely distributed version of Cyc. Although it is not intended as a lexical ontology it does contain a hierarchy of classes and subclasses, called "genls". Phrases are also included in Cyc, although generally they appear put together as a single word. For example the phrase "platinic idea" is represented in OpenCyc as "PlatonicIdea". Although OpenCyc contains only a fraction of the relationships that the full version of Cyc does, 1,608 relationships are still extracted from this resource. When mining

relationships from Cyc I do not take all possible hypernyms. Cyc makes use of multiple inheritance which is quite frequently used. As a result, when searching for hypernyms many concepts that are only loosely related to the hyponym are retrieved. To compensate for this only hypernyms of 4 steps or less are retrieved.

The process of actually mining the relationships from Open Cyc is similar to that of mining relationships from WordNet, although there are some differences. Words are taken from Roget's Thesaurus, and the phrases are modified to appear as they do in Open Cyc. The spaces between the words in the phrases are removed and the first letter in every word is capitalized. Once this is done they can be searched for in OpenCyc. A function already developed for Roget's Thesaurus Jarmasz and Szpakowicz (2003b) can test to see if two words are in the same paragraph (although if a word is part of a phrase it is also counted as being in the paragraph). This was used initially when selecting hypernym pairs from Open Cyc. These hypernym pairs were later filtered by removing all pairs that do not appear as exact terms or phrases in the same paragraph.

5.2.2 Mining Hypernyms Relationships from Dictionaries

A second method for extracting hypernym/hyponym pairs is to use dictionaries. One such resource is the Longman Dictionary of Contemporary English (LDOCE). This resource has been used frequently in the past to mine relationships from text. Patterns have been used before to extract relationships from LDOCE (Nakamura and Nagao, 1988). These patterns are:

- {determiner} {adjective}* *key noun* {adjective phrase}*
- {determiner} {adjective}* *function noun* of *key noun* {adjective phrase}*

An adjective phrase is a phrase that describes a noun or pronoun. A function noun is any noun that indicates a type of semantic relationship between the word being defined and the key noun. For example "kind", "type", "set", or "feeling" can all be function nouns. A key noun is the noun that the word being defined is related to. In the first pattern, the key noun will almost always be a hypernym, in the second pattern, the key noun, may be a hypernym, or may be related by some other semantic relationship to the word being defined. The pattern ends with 0 or more adjective phrases.

Once hypernym pairs are extracted from the dictionaries these hypernyms are "filtered" using Roget's Thesaurus. Filtering basically means removing all hypernym pairs where the two terms/phrases do not co-exist in the same paragraph.

As an example the word "acknowledgements" is defined as : "a short piece of writing at the beginning or end of a book in which the writer thanks all the people who have helped him or her". In this case "piece" is the function noun, and "writing" is the key noun. There are no adjective phrases in this definition. "at the beginning or end of a book ..." would be a preposition phrase. The key noun is taken as the hypernym of the term being defined. Nakamura and Nagao (1988) claim that the first pattern is far more frequent than the second one. Although the function noun can be used to determine the kind of relationship, between the word being defined and the key noun, it is not necessarily useful to do so. For example "abhorrence" is a "feeling of hatred" in LDOCE, "feeling" being the function noun and "hatred" being the key noun. Although abhorrence is a kind of hatred, it is also a kind of feeling. It is not clear that separating the function noun from the key noun will be helpful as "abhorrence *is a* feeling of hatred" is already a valid hypernym. Also phrases of this sort (*function noun* of *key noun*) do exist in Roget's Thesaurus. Some examples of these phrases in Roget's Thesaurus are:

- feeling of obligation
- set of rules
- piece of information

It should be noted that 86% of the hypernyms extracted come from the first pattern (before filtering with Roget's Thesaurus). This means that even if the relationships discovered with the second pattern are not as accurate they only contribute a small number.

Extracting hypernyms from electronic dictionaries takes a few steps. First of all the format of the files must be examined to determine what words are being defined. In the case of LDOCE information including part of speech, pronunciation and examples of its use can be found along with a definition. Figure 5.1 shows an example of how the definitions appear in the LDOCE text files. The head word in the tags "<HWD>" and its definition in the "<DEF>" tags are extracted. Once the word and its definition is extracted the Brill Tagger (Brill, 1994) is used to tag the part of speech of each definition. Finally once the part of speech for every word is identified patterns can be used to extract the hypernyms from the definition.

There does not seem to be a simple way to identify an adjective phrase, and so the problems in Nakamura and Nagao (1988) are patterns are generalized to appear as follows, where the hypernym is the key noun from the previous example.

```
<Entry><Head><HWD>abandon</HWD>
<HOMNUM>2</HOMNUM>
<POS>n</POS>
<GRAM>U</GRAM></Head>
<Sense><LEXUNIT>with gay/wild abandon</LEXUNIT>
<ACTIV>CONTROL</ACTIV>
<DEF>in a careless or uncontrolled way without thinking or
caring about what you are doing</DEF>
<EXAMPLE>The kids hurled pieces of wood on the fire with gay
abandon.</EXAMPLE></Sense></Entry>
```

Figure 5.1: Example definition of the word "abandon" in LDOCE

- {determiner} {adjective}* *key noun*
- {determiner} {adjective}* *function noun* of *key noun*

These patterns have been shown to work well for LDOCE in Nakamura and Nagao (1988). I attempt to apply them to Wiktionary as well. This is somewhat more difficult since Wiktionary is created by members of the general public, where as LDOCE is built by professionals. As such these patterns, which are frequent in LDOCE, may not appear as frequently in Wiktionary.

Wiktionary is a free-for-use collaborative project. It allows everyone to create and edit definitions for words, as well as list synonyms, antonyms, pronunciations, translations, and lots of other information. The English version of this page has over 150,000 entries as of the writing of this thesis (Wiktionary, 2006).

The only real difference between mining relationships from LDOCE and mining those from Wiktionary is how the format of the definitions is handled. While LDOCE uses tags, Wiktionary uses both tags and a series of headers and definitions that are changed to html when viewed. Users of the system write these headers and definitions. This can make identification of some definitions difficult. An example of how Wiktionary formats their definitions can be seen in Figure 5.2. The word being defined appears in the "<title>" tag while the definitions appear preceded by a "#" under the "===Noun===" header inside the "<text xml:space="preserve">" tag. Examples can be distinguished from definitions because they are in quotes and preceded by "#:". Once the word and its definitions are extracted they are tagged with the Brill Tagger (Brill, 1994) and the previously described patterns are used to extract the hypernyms.

```
  <page>
    <title>wares</title>
    <id>183</id>
    <revision>
      <id>833665</id>
      <timestamp>2006-02-17T23:59:34Z</timestamp>
      <contributor>
        <username>RobotGMwikt</username>
        <id>3478</id>
      </contributor>
      <minor />
      <comment>robot  Adding: io</comment>
      <text xml:space="preserve">==English==
===Noun===
'''wares''' ''plural''

#Items that are for sale.
#: ''The square was filled with booths, with vendors offering their wares.''

====Synonyms====
*[[goods]], [[merchandise]], [[product]]s

===See Also===

*[[warez]]
[[fr:wares]]
[[io:Wares]]</text>
    </revision>
  </page>
```

Figure 5.2: Example definition of the word "merchandise" in Wiktionary

A total of 5153 definitions are extracted from LDOCE using this method. This is slightly more than Wiktionary where 4483 relationships are extracted.

5.2.3 Mining Hypernym Relationships from a Large Corpus

Three different attempts are made to extract relationships from a large corpus. Two attempts involve using patterns found in (Hearst, 1992) to extract relationships from text. Two different resources are used. The first is the British National Corpus (BNC) (Burnard, 2000) and the second being the Waterloo MultiText System (Clarke and Terra, 2003). The third attempt involves applying Machine Learning using WordNet and the BNC, to the task of hypernym extraction. It is not as successful as the other methods described here. Its presentation can be found in more detail in Section 5.2.6.

Mining the BNC

In (Hearst, 1992) six different patterns were used to extract hypernyms from text. These patterns are applied to the BNC. The BNC already labels each term/phrase with a part of speech tag, which is convenient for implementing the patterns. For example the phrase "such as" is already labeled as a preposition. The patterns are as follows:

- *such NP as* {*NP,* }* {*(and* | *or)*} *NP*
- *NP such as* {*NP,* }* {*(and* | *or)*} *NP*
- *NP* {*NP,* }* *or other NP*
- *NP* {*NP,* }* *and other NP*
- *NP* {,} *including* {*NP,* }* {*(and* | *or)*} *NP*
- *NP* {,} *especially* {*NP,* }* {*(and* | *or)*} *NP*

For my purposes a Noun Phrases (NP) is 0 or more adjectives followed by 1 or more nouns: *Adjective* Noun+*. The BNC contains approximately 100 million words and 6 million sentences (called S-units) (Burnard, 2000). Approximately 90% of the BNC is written English, while the other 10% is spoken English.

All the text in the BNC is used and a total of 1332 relationships are discovered using this method. This is much lower than I had expected, however it can be explained by the fact that most of the relationships extracted from this resource do not occur in the

same paragraph in Roget's Thesaurus. In fact almost 30,000 hypernym relationships were initially extracted from the BNC before filtering with the Thesaurus.

Mining the Waterloo MultiText System

The Waterloo MultiText System (Clarke and Terra, 2003) contains half a terabyte of web data and another half terabyte index. Its size will make it more likely to contain a large number of extractable hypernym pairs. Unlike the BNC the data in the Waterloo MultiText System does not come tagged for part of speech. It is also not designed in such a way that makes it easy to search for specific patterns across the whole corpus. This system requires a query as well as a maximum number of results to return entered into the system. Since at most 2000 queries can be returned safely I decided that searches should be done for specific terms in conjunction with the Hearst (1992) patterns. This method is supposed to be used where all other methods failed. The reason for this is that using Hearst's patterns is less accurate than taking relationships from either dictionaries or other lexical resources. Also unlike other methods this one can be used to search for specific words and phrases that may not appear in other lexical resources. It is also more time consuming to run than any of the other methods.

First a list of terms that has no assigned hypernyms from any of the methods described above, or in Section 5.2.4 gets compiled first. This list contains 26430 unique terms. Some terms are repeated because they appeared in several places in the Thesaurus. With this list the term/phrase X is placed into Hearst (1992)'s patterns in the following way:

- *such NP as X*
- *NP such as X*
- *X or other NP*
- *X and other NP*
- *NP* {,} *including X*
- *NP* {,} *especially X*

100 examples of each query are retrieved from the Waterloo MultiText System. Many of the queries do not give any results. Of the 26430 unique words searched for 15443 had at least one phrases retrieved using this method. 100 results for each query is chosen

because it allows for a large number of possible hypernyms for each word (600 if all 6 patterns return 100 results) while still running relatively quickly (about 4 days for all 26430 words/phrases). It is possible to search for more than 100 results, however assuming one has obtained one or two correct hypernyms from the first 100, there is not much benefit in retrieving another 2000 query results.

Once the terms are extracted they are tagged using the Brill Tagger (Brill, 1994). This may be the source of some errors since the text retrieved is not always complete sentences. Since the Waterloo MultiText System does not count punctuation in its patterns many of the extracted sentences had to be disregarded due to incorrect or irregular punctuation. After a noun phrase gets extracted the hypernym pair is filtered through Roget's Thesaurus to make sure that they both appeared in the same paragraph. In the end 11392 relationships are extracted using this method.

5.2.4 Inferring new Hypernyms using Synonymy

Another method of acquiring new hypernym pairs is to infer new relationships from old ones using synonymy. Since all possible hypernym pairs from WordNet are already retrieved there is no point in using WordNet synsets to infer new relationships. Instead two other resources are used. The first is synonym lists from Wiktionary, the second is the semicolon groups of Roget's Thesaurus itself.

For many terms in Wiktionary a list of synonyms is provided. In Wiktionary it is not clear what sense of the word from Wiktionary these synonyms apply to. This is not a problem though since relationships are only taken if the two words co-occur in the same Roget's Paragraph. The word senses in the original dictionary or lexical resource are not taken into account. Figure 5.3 shows the formatted definitions and synonyms of "music". To infer new relationships all hypernym pairs extracted from the following sources are taken. One can also see unformatted examples of synonyms for "wares" in Figure 5.2, which includes "goods", "merchandise" and "products".

- WordNet
- OpenCyc
- Longmans Dictionary of Contemporary English (LDOCE)
- Wiktionary definitions
- The British National Corpus (BNC)

music (mass noun)

1. a natural intuitive phenomenon operating in the three worlds of time, pitch, energy, and under the three distinct and interrelated organization structures of rhythm, harmony, and melody
2. sound organized in time in a melodious way coming from an instrument (or appearing to), as opposed to song.
3. a song accompanied by instruments, or appearing to.
4. any pleasing or interesting sounds
5. a guide to playing or singing a particular tune as opposed to just the lyrics-sheet music
6. (with capital M) the subject devoting to song and playing instruments
7. Something wonderful.

Synonyms

- melody
- vibe

Figure 5.3: Definition of "music" from Wiktionary

The previously described method using the Waterloo MultiText System does not get employed until after synonyms are used to infer new relationships. This is done because this method is supposed to be a last resort to find relationships that can not be obtained any other way.

From the list of hypernym pairs, any term/phrase that had a synonym in Wikitionary gets replaced by that synonym to create a new hypernym. This is applied to both the hypernym and hyponym terms in the relationship. For example: In WordNet the hypernym "cancer *is a* sickness" appears. In Wiktionary "sickness" has three synonyms "disease", "illness" and "infirmity". This is used to infer three new relationships

- "cancer *is a* disease"
- "cancer *is an* illness"
- "cancer *is an* infirmity"

If these newly inferred relationships appear in the same Roget's Paragraph, then they are kept. A total of 10,718 new relationships are inferred using this process.

When using Roget's Thesaurus to infer new relationships I apply what was discovered in Chapter 4. That is, that coordinate terms and synonyms are present in most semicolon groups. This means that given a pair from one of the above five resources, the hyponym should, in many cases, be interchangeable with the other terms in the same semicolon group. Unlike when I used Wiktionary synonyms the hypernym term in the relationship is not changed. This process is applied twice, once to discover which semicolon groups had no hypernyms assigned yet (those terms are then used when searching for relationships in the Waterloo MultiText System. The second time it is used on all mined hypernyms, including those from the Waterloo MultiText System. One problem with both this method, and using Wiktionary synonyms to infer new relationships is that it relies on the relationships taken from the other resources being accurate. The accuracy of these two methods cannot be greater than the average accuracy of all other methods combined.

In Table 5.1 it can be seen that the number of words with hypernyms is 41,242, and they are contained in 19,027 Semicolon groups. 12,104 semicolon groups contain only terms with no known hypernyms. The Waterloo MultiText System is used to search for the 34,983 terms with no hypernyms. From this 37,901 new relationships are inferred by using this method. Table 5.1 counts the number of relations before relationships from the Waterloo MultiText System were applied, and Table 5.2 contains the same numbers for after that systems results are added.

Semicolon Group With Hypernyms	19,027
Semicolon Group Without Hypernyms	12,104
Words With Hypernyms	41,242
Words Without Hypernyms	34,983
Words With Implied Hypernyms	37,901

Table 5.1: Count of terms and semicolon groups with hypernyms before using the Waterloo Multitext System

Semicolon Group With Hypernyms	20,711
Semicolon Group Without Hypernyms	10,420
Words With Hypernyms	45,403
Words Without Hypernyms	29,323
Words With Implied Hypernyms	39,400

Table 5.2: Count of terms and semicolon groups with hypernyms after using the Waterloo Multitext System

5.2.5 Decisions when Mining

When extracting hypernyms from the various resources I had to make decisions regarding how to extract relationships, and which relationships to extract. These decisions can be thought of as adjusting a threshold that affects the precision and recall of the relationships extracted from these resources. In most cases, different thresholds could be used, but since evaluation is done using human users it is difficult to examine each resource with more than one threshold. For this thesis precision is considered to be the accuracy of the relationships extracted from a resource. Recall is the proportion of the set of all relationships that are extracted from that resource. See Section 6.2.1 for a discussion of the precision and recall of each resource.

I decided that hypernym pairs separated by any number of hypernym links in WordNet would be imported into Roget's Thesaurus. I did this because these hypernyms are correct even if they are farther apart. Pairs of words in WordNet that are distant hypernyms of each other are less semanticly similarity than closely related hypernyms. Since the terms in these hypernyms are less closely related they are also less likely to appear in the same Roget's paragraph. These hypernyms are not incorrect they are simply not closely related. For this reason all hypernym relationships in WordNet, regardless of

distance, are kept as candidates for importing into Roget's Thesaurus.

When mining relationships from OpenCyc an upper limit was put on the distance betwee words/phrases in the hierarchy. The main reason for this is simply the run times required to extract the relationships, and the amount of manual supervision needed. When a maximum hypernym distance of 4 is used it takes about 4 hours to extract the relationships from OpenCyc on a computer with 512 MB RAM and a 3.4 GHz Intel Pentium 4 processor. One drawback of the OpenCyc resource is that its Java interface will inexplicably stall periodically. As such the longer the program runs for, the more manual supervision is required. I conducted a test where relationships for the words in the first 100 heads were mined using a depth of 4 and a depth of 5. The run time and number of relationships (before being incorporated into the Thesaurus) were counted. When using a depth of five, 547 unique relationships are mined in 1 hour and 36 minutes. When using a depth of four, 523 unique relationships are extracted in 37 minutes. This suggests that there is very little to be gained by extending the search depth from 4 to 5 in OpenCyc. The number of relationships went up by about 4% while the run time tripled. When a distance of 4 is used a total of 2823 unique relationships are extracted before 1608 were finally incorporated into Roget's Thesaurus.

My reasoning for limiting both the number of queries and the number of query results for the Waterloo MultiText system was influenced by the run time required to extract these relationships. It requires 4 days of constant access to the system to extract the relationships.

Mining Relationships from LDOCE, Wiktionary, the Waterloo MultiText System and the BNC was done using established patterns that have been shown to work in previous works by (Nakamura and Nagao, 1988) and (Hearst, 1992). It is possible to use other patterns discovered in other works, or to explore patterns of my own invention. It is also be possible to bootstrap for new patterns. The problem with adding new patterns by any method is that it is not clear how accurate these patterns would be. Common sense dictates that doing so would likely increase recall at the cost of precision. I did not attempt any such methods largely because the methods I used are already well established as effective methods of extracting hypernym relationships. It was not clear how well the patterns for mining LDOCE would transfer to mining Wiktionary.

When inferring new relationships using synonyms from Wiktionary I made the decision to swap both words in each hypernym relationship with their synonyms. Another option would have been to swap only hyponym terms with its synonyms. Since Wiktionary has not been used for this purpose so far it is not clear how well either of these

methods will work and so I decided to chose the method that was most likely to generate the larger number of hypernym relationships. Likewise when inferring new relationships from Roget's Thesaurus if any word in a semicolon group had a hypernym in the same paragraph then that hypernym was assigned to all words in the semicolon group. This method can perhaps be made more accurate if hypernyms are only assigned to all words in a semicolon group when more than one term/phrase in the semicolon group already has that hypernym relationship. I chose a very low threshold of 1 because I wanted to generate as many hypernyms as possible using this method. Both of these methods may benefit from only being used to infer new relationships from relationships that come from trusted resources (i.e. resources with known high accuracies).

5.2.6 Machine Learning for Hypernym Discovery

Another method that I attempted is using Machine Learning to learn patterns for hypernyms from text. I start with a method similar to that of Snow et al. (2005) and then make several significant modifications. Some differences include both the data in the training/testing corpus and its size as well as small differences in how the dependency paths are formatted. This is also my own implementation of the system, it is not modified from the original code in Snow et al. (2005) and so it is difficult to be sure what all of the differences between my implementation and theirs are. Other differences are included in the following description of my method.

For this method I use a corpus, specifically the BNC. In Snow et al. (2005) text from the Wall Street Journal, Associated Press and Los Angeles Times were used (about 6 million sentences in total). I parse approximately half of the BNC (about 3 million sentences) using Minipar (Lin, 1998a). Next I create a labeled data set of sentences from the parsed sentences in the BNC. Pairs of nouns are taken from each sentences and labeled as hypernyms or non-hypernyms using WordNet. Each pair contains a hypernym A and a hyponym B. A pair of terms is only labeled as having a hypernym relationship if both A and B have only one sense that is frequent in WordNet, and for those senses A is a hypernym of B. A sense is frequent if it has a frequency of greater than 0 and all other senses of the word have a frequency of 0. A hypernym can be any word in the hypernym chain, not just the first word. For a pair to be labeled as a non-hypernym pair, A must not be a hypernym for any sense of B. Any pair that does not fit these criteria is not labeled. As a result many correct hypernym pairs go unlabeled. To balance the corpus off under-sampling of the negative class is done. This is described in more detail later in

this section. Each pair A and B is labeled twice. Once for if A is a hypernym of B and again for if B is a hypernym of A.

The output from Minipar is a dependency tree. This dependency tree is a tree structure that contains nodes for each word, and links indicating relationships between the words. In some cases there are empty nodes in the tree and in other cases the same word may get more than one node. The next step is to extract dependency paths for all known hypernym and known non-hypernym pairs. A dependency path is simply the shortest path through the dependency tree between two terms. The dependency path includes the words, part of speech and relationships between words in the path. The dependency path can be at most 5 words in length. Extra satellite links, like in Snow et al. (2005) are included in the dependency path, although I do experiments both with and without satellite links. Satellite links are potential links to words on either side of the two nouns at the paths ends. If these words are directly linked to one of the nouns at the ends of the path then they are included. In Snow et al. (2005) satellite links were not included for words that are already found in the path. This will eliminate some redundant information but it is not necessary. Punctuation is not included as a satellite link. These links are included to allow for patterns such as "such X as Y" where the word "such" may not appear in the path. There can be up to 4 satellite links, two at each end of the dependency path. Another difference between my dependency paths and those of Snow et al. (2005) is that I take direction of the relationship into account. Nouns represented by "N" are found at each end of the dependency paths and satellite links are contained in brackets at either end of the path. The format of the path goes "*word#part of speech*:*relationship*:*word*", and repeats for the entire path. The *relationship* has arrows, either "<" or ">" on either side of it to indicate which direction the relationship link from Minipar is directed. An example of a dependency tree from Minipar can be seen in Figure 5.4. There are two nouns in the sentence in Figure 5.4 "dog" and "pet". An example of a dependency path going between "pet" and "dog" is:

$$N :< pcomp-n <: as\#Prep :< mod <: keep\#V :> s >: N(N :> punc >: and\#U)$$

This example of a satellite link is the extra link to the word "and" that appears next to "Cats" in the sentence. Since this method relies on counting dependency paths, the actual format of the dependency path will not affect the results of the system.

Some word pairs, and features are not counted. Pairs of words that appear with fewer than 5 different dependency paths are removed. After this is done features that appear fewer than 5 times are removed. This does have the effect that some pairs of words

```
> (
E0 (() fin C * )
1 (Cats cat N 6 s (gov keep))
2 (and   U 1 punc (gov cat))
3 (dogs dog N 1 conj (gov cat) (additional 6 s))
4 (are be be 6 be (gov keep))
5 (often   A 6 amod (gov keep))
6 (kept keep V E0 i (gov fin))
E2 (() cat N 6 obj (gov keep) (antecedent 1))
7 (as   Prep 6 mod (gov keep))
8 (pets pet N 7 pcomp-n (gov as))
9 (.   U * punc)
)
```

Figure 5.4: Minipar output for the sentence "Cats and dogs are often kept as pets."

will have fewer than 5 features remaining. Alternatively, if features with fewer than 5 appearances in the corpus are removed, then pairs of words with fewer than 5 features are removed, some features would appear fewer than 5 times. It is not completely clear from Snow et al. (2005) in what order this was implemented.

The next step is under-sampling the negative class of the testing and training data. Since many positive examples are not included the number of negative examples are disproportionately large. In Snow et al. (2005) human evaluators are used to determine how frequently hypernym pairs appear in text. They found that there is approximately a 50:1 ratio of non-hypernym noun pairs to hypernym noun pairs in text. This was discovered on a test set of 5387 noun pairs. 134 noun pairs were found to be hypernyms, 131 were found to be coordinate terms, and the other 5122 were found to be unrelated. This ratio of 50:1 seems somewhat lower than I had expected. It was my intuition that fewer than 1 in 50 noun pairs found in random English text were hypernyms so I do two small experiments to attempt to verify these results.

In the first experiment I used WordNet to label pairs of nouns from the BNC. A pair of words was labeled as a hypernym pair if for any noun sense of the two words, they are found to be hypernyms in WordNet. A pair is labeled negative if for no senses of the words are they hypernyms in WordNet. This will not give a completely accurate picture of what the ratio should be, however it will give a ratio that can be thought

of as a lower bound. I found that there was approximately a 57:1 ratio of negative to positive examples in a few thousand sentences from the BNC. This experiment is still not particularly accurate since it is not guaranteed that the hypernym/hyponym pair will be of the correct word senses. I used Roget's Thesaurus to filter out hypernym pairs that do not exist in the same paragraph and found that the ratio drops to approximately 17:1.

A second experiment is carried out using the Semantic Concordance Package (Semcor) (Miller et al., 1993) with WordNet; Semcor is a series of sentences where each word is labeled with its correct WordNet sense. Semcor was designed to work with senses from WordNet 1.6. That is, SemCor was annotated using senses from WordNet 1.6, although the senses can be mapped to newer versions of WordNet. Using Semcor and WordNet 1.6 I found that there was a 165:1 ratio of negative to positive pairs of terms. Once again, I used Roget's Thesaurus to filter the hypernym pairs. This resulted in a ratio of 54:1.

This suggests that an imbalance of 50:1 in regular English text is a bit too low. Another issue with the Snow et al. (2005) under-sampling method is that it under-samples by selecting the most frequent negative examples. As a result the under-sampling is not completely random. I try two different variations of under-sampling, one by using the Snow et al. (2005) method of under-sampling by selecting the most frequent examples to a 50:1 ratio. The second is to randomly under-sample the negative class to a 165:1 ratio. In addition to this, I experiment with satellite links, I try both with them and without them. I also experiment with filtering the training and testing sets with Roget's Thesaurus.

In Snow et al. (2005) three different machine learning algorithms were tried: Multinomial Naïve Bayes, Complement Naïve Bayes and Logistic Regression. While using these algorithms I found that Logistic Regression was extremely slow and required enormous amounts of RAM to run some experiments, and so could not always be used with my variations on the method. Instead I use the two variants of Naïve Bayes[1] and Support Vector Machines (SVM)[2].

10-fold cross validation was done on all these tests in order to determine how well the different variations worked. However since under-sampling is used, it is not clear that the results obtained from 10-fold cross validation will accurately reflect the true accuracy of the classifier. As such human testers are used to evaluate the quality of the

[1]For Multinomial and Complement Naïve Bayes the Weka implementations of these algorithms were used (Witten and Frank, 2005).

[2]SVM Light was the implementation of Support Vector Machines used (Joachims, 1999).

most successful variation on the proposed method. Results for both of those evaluations are shown in Chapter 6.

5.3 Building a Usable Hypernym Network

As a final step in adding hypernym relationships to Roget's Thesaurus I need an interface for retrieving these relationships. Each hypernym relationship is taken and inserted into all the paragraphs in which the word pair is found. An interface was then developed where a word, its head number and paragraph number within the head can be given to retrieve that words hypernyms and hyponyms.

To determine which hypernyms to include, an evaluation is done in Section 6.2. Hypernyms from only the most accurate resources are added to the Thesaurus. There are 68,717 unique hypernyms that appear 92,675 times in the Thesaurus. This happens because some hypernyms will appear in two or more paragraphs.

5.3.1 Removing Redundant Hypernyms, and Breaking Cycles

Cycles and redundant links should be removed. The first step is to remove cycles from the hypernym structure. A cycle is a series of hypernym links where a term can eventually become its own hypernym. For example "A *is a* B *is a* ... *is a* C *is a* A" is a cycle. One way to fix a cycle is to remove one of its links. This could be accomplished by removing the link that is least likely to be correct. In the next chapter I discuss how human evaluators are used to determine the accuracy of hypernyms taken from each resource and the results of this evaluation. A probability can be computed for the hypernyms from each resource using the human evaluators. A probability can be associated with each hypernym pair by combining the probabilities for each resource from which it is taken. This process is described in more detail in Section 6.2.2. Using this technique a probability of error can be computed for all hypernym pairs collected. In a cycle the hypernym link most likely to be in error can be dropped to fix that cycle. In the case where there are two or hypernyms with the lowest probability, the first one found in the cycle is removed. There were 3,756 cycles found and fixed in the thesaurus. Some examples of cycles that were fixed can be seen in Figure 5.5.

A second source of error is redundant hypernym links. This happens in cases when a series of relationships: "A *is a* B *is a* ... *is a* C" exists as well as the relationship: "A *is a* C". The relationship "A *is a* C" is not incorrect, however it is unnecessary since C is

cycle: place *is a* rank *is a* position *is a* place
probabilities:
place *is a* rank : 0.735
rank *is a* position : 0.94223
position *is a* place : 0.782

place is a rank is removed

cycle: construction *is a* work *is a* construction
probabilities:
construction *is a* work : 0.843632
work *is a* construction : 0.735

work *is a* construction is removed

Figure 5.5: Examples of cycles removed from the hypernym hierarchy.

homogeneity *is a* uniformity *is a* sameness
Redundant hypernym: homogeneity *is a* sameness

thickness *is a* width *is a* dimension
Redundant hypernym: thickness *is a* dimension

Figure 5.6: Examples of redundant hypernym links in the hypernym hierarchy.

already a hypernym of A, just not an immediate hypernym. These redundant hypernym links can be simply dropped. A total of 30,068 redundant hypernym links were found and removed. Examples of redundant links removed can be seen in Figure 5.6.

After these two fixes were done 58,851 hypernym relationships were left. The remaining relationships were incorporated into the ELKB, so that it is possible to query a word's hypernyms and hyponyms by providing the word/phrase and the paragraph in which it is located.

Chapter 6

Evaluating the Resource

Evaluation is carried out in several different ways. For the system that makes use of Machine Learning 10-fold cross validation can be used as one method of evaluation. Human evaluators are also used to evaluate all nine methods of extracting hypernyms, including the Machine Learning one. Finally hypernyms mined from the most successful systems are incorporated into the thesaurus and the newly enhanced thesaurus is tested on three applications. The first application is that of finding semantic similarity between terms, tested on Miller and Charles (1991) types data sets. The second method is to use this same similarity function to solve TOEFL style synonym identification problems. The third is for solving SAT analogy problems.

The human evaluation shows mixed results. Many evaluators did not agree on which word pairs actually are hypernyms. The results for testing on applications are mostly positive as they show an improvement on six out of seven data sets, and no change on the remaining one.

6.1 Evaluating the Machine Learning Hypernym Classifier

I test several variations on the method Machine Learning method based on Snow et al. (2005) and described in the previous chapter. These variations include using Roget's Thesaurus to filter the training and testing data, two methods of under-sampling, and either including or not including satellite links. Three machine learning algorithms are compared as described in Chapter 5.

In Tables 6.1, 6.2 and 6.3 Multinomial Naïve Bayes, Complement Naïve Bayes and

Support Vector Machines are used. In the first column, labeled *Filter*, "None" indicates that no filtering is used, and "Yes" indicates that Roget's Thesaurus is used to filter both the positive and negative classes in both the training and testing data. As mentioned before removing word pairs that do not appear in the same paragraph anywhere in the thesaurus filters the data.

From these values it can be seen that most of the results do not give a precision of over 0.5. It is hard to tell if satellite links are useful or not. In 8 cases the results improve when the satellite links are removed, however 4 times they are better when satellite links are included.

Unexpectedly the results for Multinomial Naïve Bayes improved when using random under-sampling with a 165:1 ratio of negative to positive examples than when under-sampling 50:1 by picking the most frequent negative examples as seen in Table 6.1. In all other systems the results were much better when under-sampling with the most frequent examples with a ratio of 50:1. Whether a system performs better or worse with these different methods of under-sampling does not mean that one under-sampling method is superior to the other, simply that they affect the results differently. The point of under-sampling is not to get higher precision and recall, but rather to represent the correct ratio of positive to negative examples of hypernyms in text.

One thing that is always helpful is filtering the training data and testing data with Roget's Thesaurus. Only pairs of nouns that exist in the same Roget Paragraph were kept. In all cases this increased the F-Measure.

In the end both Multinomial and Complement Naïve Bayes do not produce results good enough to use in my system. There is no sense in including results with a precision of less than 50% as most of the relationships will be incorrect. One variation on SVM is fairly successful though as seen in Table 6.3. This variation used Roget's Thesaurus as a filter, it did not use satellite links, and it used the 50:1 under-sampling ratio method of under-sampling. Precision, recall and F-measure of about 0.63 is achieved. This system is selected and the entire data set is used to train a classifier and it is then run on previously unseen data from the BNC. The results from this data are then manually evaluated by a set of judges. This is the topic of Section 6.2.

6.2 Manual Evaluation of Hypernyms

Human raters can also be used to evaluate a subset of the results for each system. I use human evaluators because only they can tell if a given pair of words is related

Filter	Under-sample	Satellite	Precision	Recall	F-Measure
None	Top 50	With	0	0	0
		Without	0.003	0.001	0.001
	Random 165	With	0.011	0.001	0.002
		Without	0.037	0.007	0.011
Yes	Top 50	With	0.048	0.012	0.019
		Without	0.042	0.009	0.014
	Random 165	With	0.406	0.094	0.153
		Without	0.421	0.036	0.066

Table 6.1: Results for 10-fold cross validation on Multinomial Naïve Bayes

Filter	Under-sample	Satellite	Precision	Recall	F-Measure
None	Top 50	With	0.039	0.211	0.065
		Without	0.105	0.112	0.108
	Random 165	With	0.009	0.295	0.018
		Without	0.016	0.622	0.032
Yes	Top 50	With	0.079	0.207	0.115
		Without	0.283	0.603	0.385
	Random 165	With	0.107	0.754	0.187
		Without	0.108	0.798	0.191

Table 6.2: Results for 10-fold cross validation on Complement Naïve Bayes

Filter	Under-sample	Satellite	Precision	Recall	F-Measure
None	Top 50	With	0	0	0
		Without	0.917	0.023	0.045
	Random 165	With	0.323	0.05	0.086
		Without	0	0	0
Yes	Top 50	With	0.571	0.048	0.088
		Without	**0.631**	**0.635**	**0.633**
	Random 165	With	0.608	0.104	0.178
		Without	0.5	0.025	0.047

Table 6.3: Results for 10-fold cross validation on Support Vector Machines

by hypernymy or not. WordNet could be used but there may be hypernyms missing from WordNet. Because of this one could never be sure that a hypernym pair labeled as incorrect by WordNet was actually incorrect and not just missing. A set of five raters (myself and four colleagues) who are fluent in English were selected and given sample sets of 200 pairs from each of the nine resources. The sample size of 200 from each resource is not extremely high, however the total number of samples is 1800. One reason for not using more samples is that this can be quite time consuming for the evaluators. Also it gives a reasonable confidence interval. A sample of 200 gives a confidence interval of +/- 7% with a 95% confidence level for each individual evaluator. For interested readers, the samples along with their ratings by each rater can be found at `http://www.site.uottawa.ca/~akennedy/mastersThesis/supplement_B.pdf`.

One of the five evaluators did not finish annotating the entire data set. The scores by each evaluator for each resource can be seen in Table 6.5. The averages of each evaluator and kappa scores for each resource, as well as for all samples can be seen in Table 6.6. An interval at the 95% confidence level is found in the table too. A few interesting results can be seen in these tables. First of all despite having an F-measure and precision of 63% my best Machine Learning system does not receive a high rating from the raters, getting an average of about 25%. This can partly be attributed to the under-sampling used on the negative class. The results found when using 10-fold cross validation may be artificially high because necessary negative examples had been removed. Although the method itself may work in some cases, my attempted implementation of it was unsuccessful.

In Table 6.6 Kappa scores and their confidence interval are presented. I use Fleiss (1981) to determine inter-rater agreement. The standard error at a confidence interval of 95% is computed and shown in the table too.

In addition to Fleiss' kappa, Cohen's kappa (Cohen, 1960) is used to determine inter-rater agreement between pairs of evaluators. The highest and lowest value for each system are shown in Table 6.6. Kappa rating systems proposed by Fleiss (1981) and Landis and Koch (1977), are shown in Table 6.4.

The judges were given the set of 1800 samples. They were not informed ahead of time from which resource the samples were taken. They were then asked to assign each sample as either a true hypernym, or a false hypernym.

From the human evaluation of hypernyms it is clear that some systems are not good enough to be used as a source of hypernyms. The worst one is my Machine Learning system for hypernym extraction. The average accuracy for this system was only 25% and so does not make it a good candidate for using in this thesis. To help determine

Fleiss' Kappa	(Fleiss, 1981)
< 0.40	Poor
0.40 - 0.75	Fair to Good
> 0.75	Excellent
Cohen's Kappa	**(Landis and Koch, 1977)**
< 0.00	Poor
0.00 - 0.20	Slight
0.21 - 0.40	Fair
0.41 - 0.60	Moderate
0.61 - 0.80	Substantial
0.81 - 1.00	Almost Perfect

Table 6.4: Evaluation measures for Fleiss' and Cohen's kappa.

Resource	Rater 1	Rater 2	Rater 3	Rater 4	Rater 5
BNC	0.675	0.605	0.75	0.665	0.62
CYC	0.95	0.775	0.86	0.855	0.885
LDOCE	0.85	0.59	0.82	0.725	0.925
Machine Learning	0.31	0.03	0.33	0.17	0.405
Waterloo MultiText	0.715	0.385	0.505	0.51	0.565
Roget's Thesaurus	0.315	0.14	0.42	0.325	0.46
Wiktionary Synonyms	0.47	0.06	0.56	0.455	0.54
Wiktionary Hypernyms	0.73	0.555	0.75	-	0.87
WordNet	0.855	0.515	0.8	-	0.77

Table 6.5: Results for each resource from the 5 evaluators (Raters).

Resource	Average	Fleiss' Kappa	+/-	Max Kappa	Min Kappa
BNC	0.663	0.436	0.044	0.576	0.309
CYC	0.865	0.379	0.044	0.573	0.25
LDOCE	0.782	0.267	0.044	0.445	0.067
Machine Learning	0.249	0.257	0.044	0.386	0.087
Waterloo MultiText	0.536	0.371	0.044	0.472	0.339
Roget's Thesaurus	0.332	0.206	0.044	0.38	0.139
Wiktionary Synonyms	0.417	0.122	0.044	0.326	-0.013
Wiktionary Hypernyms	0.726	0.168	0.057	0.273	0.096
WordNet	0.735	0.106	0.057	0.197	-0.036
Total	-	0.389	0.015	-	-

Table 6.6: Average results and kappa for each resource.

Resource	Average	Fleiss' Kappa	+/-	Max Kappa	Min Kappa
BNC	0.678	0.432	0.057	0.576	0.309
CYC	0.888	0.332	0.057	0.448	0.25
LDOCE	0.83	0.285	0.057	0.42	0.067
Machine Learning	0.304	0.336	0.057	0.386	0.087
Waterloo MultiText	0.574	0.354	0.057	0.388	0.339
Roget's Thesaurus	0.38	0.225	0.057	0.38	0.139
Wiktionary Synonyms	0.506	0.205	0.057	0.326	0.066
Wiktionary Hypernyms	0.783	0.214	0.08	0.273	0.096
WordNet	0.808	0.075	0.08	0.197	-0.036
Total	-	0.405	0.02	-	-

Table 6.7: Average results and kappa for each resource when Rater 2 is not included.

Positive	5 pos	4 pos	3 pos	2 pos	1 pos	0 pos
BNC	80	30	24	22	27	17
CYC	136	30	12	9	11	2
LDOCE	94	44	32	14	12	4
Machine Learning	6	11	18	40	41	84
Waterloo MultiText	40	42	25	32	29	32
Roget's Thesaurus	10	16	23	47	55	49
Wiktionary Synonyms	4	28	51	46	40	31
Wiktionary Hypernyms	-	72	69	34	18	7
WordNet	-	73	63	46	15	3

Table 6.8: Agreement among the 5 raters.

why the 10-fold cross validation results were so much better than the results found by the evaluators I do another test. I found that if one selects all samples where just one feature was found then one gets .673 precision, .555 recall and .608 F-measure for the positive class. This is because positive examples tend to have far fewer features than negative examples because they appear less often. This could have been a factor in creating misleading results. In Chapter 5.2.6 I explain that I removed all samples with less than 5 features, and then removed all features that appear less than 5 times. This has the affect that some samples will still have less than 5 features. If, alternatively, I had removed all features that appear less than 5 times and then remove all samples with less than 5 features, then perhaps this would have worked better. This would have the affect that many features would appear fewer than 5 times. Further experimentation with this system is left for future work.

Unfortunately using synonyms from Wiktionary and using Roget's Thesaurus semi-colon groups to infer new relationships does not work much better, as can be seen in Tables 6.5, 6.6, 6.7 and 6.8. The relationships extracted from the Waterloo MultiText system are not as good as I had hoped, however the system does give an accuracy of greater than 50%. Given that the methods used to extract relationships from the Waterloo MultiText system are similar to those used on the BNC (which had an average of 66%) one may expect similar results, however there are a few reasons why this did not happen. First of all the BNC has part-of-speech tags already provided, while the Brill Tagger used on sentences (or sentence fragments) from Waterloo MultiText will contain some errors. Another, perhaps more important factor, is that the Waterloo MultiText

system was used on terms for which no hypernyms could be found using other systems. The words that are being assigned hypernyms may be less frequently occurring in text, or may represent more abstract concepts and so be more difficult to assign hypernyms to. It may also be that some terms do not have any hypernyms co-occurring in the same Roget's Paragraph and so any assigned hypernyms would be incorrect.

From Table 6.5 and 6.6 one can see that Cyc is consistently one of the two top-rated systems averaging at 87% accuracy, however relatively few, only 1,608 pairs were extracted from this resource. LDOCE, WordNet and Wiktionary Hypernyms all averaged over 70% which is quite good. The BNC and the Waterloo MultiText system gave averages of 66% and 54% respectively. Hypernyms mined from resources such as BNC, CYC, LDOCE and Wiktionary Hypernyms and WordNet all provided much better results averaging over 65% accuracy each. All the hypernym pairs extracted from these resources can be included in the thesaurus. Quite a few relationships are found to co-occur in two or more of the resources. The count of co-occurring hypernym pairs found in the resources can be seen in Table 6.9. Relationships taken from the Waterloo MultiText system can also be included however they are less reliable. Although most of the hypernym pairs found with in the Waterloo MultiText system are new, some of them were already found. The numbers are shown in Table 6.10. No hypernyms from a system with an average of less than 50% accuracy is included in the enhanced Roget's Thesaurus.

Looking at Table 6.6 one can see that the kappa scores are not very high. If one uses the evaluation criteria proposed by Fleiss (1981) from Table 6.4 it can be seen that with the exception of the BNC all the kappa scores can be considered poor. This is particularly true for the hypernyms mined from Wiktionary and WordNet. The maximum and minimum Cohen's Kappa scores show similar results. These kappa result are somewhat lower than the score of 0.51 shown in Rydin (2002) on a similar problem, though in a different language.

In Table 6.5 it can be clearly seen that Rater 2 gives consistently lower ratings than all the other raters. I remove his ratings and then find the averages and kappa scores for all systems in Table 6.7. These results, not surprisingly show an improvement in the average accuracy for all systems. Fleiss' Kappa does not improve greatly from system to system, although it does improve when taken across all systems.

Instead of showing the average rating for each system, I show the amount of agreement in Table 6.8. This table shows how many times 5, 4, 3, 2, 1 or 0 raters agreed that any sample was a positive example. There are 5 raters for most of the systems except for Wiktionary Hypernyms and WordNet where there are only 4 raters. From the evaluators

one can see that at least 1 rater stated that each sample was indeed a hypernym pair in at least 96% of the examples in CYC, LDOCE, Wiktionary Hypernyms and WordNet. This may not be the most accurate method of evaluating the samples since it stands to reason that the more raters one has, the more likely one of them is to give a positive rating to any one sample. Even though many of those samples may be incorrectly labeled as hypernyms this suggests that their identification as a hypernym is at least acceptable to some. BNC and the Waterloo MultiText system had 92% and 84% respectively, of their samples with at least one positive rating from the 5 raters. Wiktionary Synonyms had a similar percentage to Waterloo MultiText, however it had far fewer samples with 5 or 4 evaluators agreeing the examples were positive.

The fact that the Kappa scores are so low combined with the fact that WordNet scored relatively poorly shows that people are not always good judges of hypernymy. Clearly identifying hypernyms is difficult for both people and computers. It may be a bit of a surprise that WordNet did not get higher scores. There are several possible reasons for this. First of all WordNet will have many infrequent senses of words and so it is quite possible that some evaluators may not realize that two words can in fact be hypernyms for some sense. It is also possible that some hypernym senses in WordNet appear to be closer to synonymy than actual hypernymy. The hypernyms from WordNet which were unanimously labeled as incorrect are:

- cup *is a* beverage
- round *is a* line
- round *is a* whole

It is easy to see how a person may not believe these to be correct hypernyms. In the case of "round *is a* whole", a "round" is a unit of ammunition, while a "whole" is a whole object or a unit. This given this knowledge this relationship would seem to be correct, but it would not be obvious unless the words were put into context, or a definition is given with them.

6.2.1 Evaluating the Usefulness of each Resource

The value of a resource for hypernym mining is a combination of two parameters: the number of relationships extracted, and the accuracy of those relationships. Each resource is assigned scores for precision, recall and F-measure. The human assigned accuracy seen

1 Resource	54188
2 Resources	4575
3 Resources	778
4 Resources	72
5 Resources	4

Table 6.9: Co-occurrences of hypernym pairs in BNC, CYC, LDOCE, Wiktionary Hypernyms and WordNet.

1 Resource	61581
2 Resources	5839
3 Resources	1102
4 Resources	171
5 Resources	21
6 Resources	3

Table 6.10: Co-occurrences of hypernym pairs in BNC, CYC, LDOCE, Wiktionary Hypernyms, WordNet and Waterloo MultiText.

Resource	Accuracy	Number	Precision	Recall	F-measure
BNC	0.663	1,332	0.663	0.004	0.008
CYC	0.865	1,608	0.865	0.005	0.010
LDOCE	0.782	5,153	0.782	0.016	0.031
Machine Learning	0.249	51,129	0.249	0.192	0.217
Waterloo MultiText	0.536	11,392	0.536	0.036	0.067
Roget's Thesaurus	0.332	176,689	0.332	0.560	0.417
Wiktionary Synonyms	0.417	10,597	0.417	0.034	0.063
Wiktionary Hypernyms	0.726	4,483	0.726	0.014	0.027
WordNet	0.735	53,404	0.735	0.169	0.275

Table 6.11: Evaluating the Resources.

in Table 6.6 is the precision. The proportion of hypernyms extracted from each resource is recall. The precision recall and F-measure are all shown in Table 6.11.

It can be seen in Table 6.11 that inferring new hypernym relationships using Roget's Thesaurus has the highest F-measure, followed by WordNet and the Machine Learning method. The number of relationships mined using the Machine Learning algorithm is going to be largely proportional to the amount of text used. Likewise, the number of relationships mined using the Waterloo MultiText system will be a function of the number of queries, and the number of results allowed per query. The number of relationships inferred using Roget's semicolon groups and synonyms from Wiktionary depends on the number of relationships mined from other resources. If very few relationships are found in other resources, then very few new relationships can be inferred using these methods.

For some of the resources used, high recall is not possible. For LDOCE and Wiktionary Hypernyms the number of definitions in each dictionary limits the number of relationships that can be extracted. This is also true for the BNC, OpenCyc and WordNet.

This method of ranking resources does show that the BNC and OpenCyc are not extremely good resources for mining hypernyms from, simply because it is difficult to extract a large number of hypernyms from them.

Another factor to consider when interpreting the results in Table 6.11 is that all the relationships extracted from these resources must be found in the paragraphs of Roget's Thesaurus. That is why the accuracy of these resources is not the accuracy for all hypernyms extracted but only for a subset that appears in Roget's Thesaurus.

6.2.2 Combining the Hypernyms from the Resources

These sets of hypernyms need to be combined in such a way as to try to promise high accuracy. The average accuracies have been generated for each resource in Table 6.6. These accuracies indicate how likely the hypernyms mined from each resource are to be correct. These results can be used to determine new accuracies for hypernyms that are mined from two or more resources. Let the probability of error in a resource A be written as *P(A)* (Devore, 1999). In the case where a particular hypernym pair x appears in just one resource the probability of error is:

$$P(x) = P(A)$$

The probability P(x) when it is found in more than one resource can be written as:

$$P(x) = P(A \cap B) = P(A|B) * P(B)$$

However if two events are independent then:

$$P(A|B) = P(A)$$

Independence between two events, means that the occurrence of one event does not affect the probability of the other event occurring. In this can, given that a hypernym x was misclassified as a positive example in resource A does not increase or decrease the likelihood that that same hypernym will be misclassified as positive in resource B. Since the two events A and B come from different resources mined with different techniques these events are independent. I compute the probability of error between two resources as:

$$P(x) = P(A \cap B) = P(A) * P(B)$$

If x appears in several resources then the probability of error can be expanded to be:

$$P(x) = P(A \cap B \cap ... \cap Z) = P(A) * P(B) * ... * P(Z)$$

Now each hypernym pair has a probability of error $P(x)$. The probability that the hypernym pair is correct is: $1 - P(x)$.

Once the accuracy for each hypernym pair has been assigned I can compute the average accuracy across all hypernym pairs. Two variations of this are examined. In the first one I took BNC, CYC, LDOCE, Wiktionary Hypernyms and WordNet and found there is a total average accuracy of 75.6% over 59617 hypernym pairs. In the second variation I take those same resources but include Waterloo MultiText and found there is a total average accuracy of 73.1% over 68717 hypernym pairs. For the first variation the top ten ranked hypernyms can be see in Table 6.12 and for variation two in Table 6.13.

Hypernym Pair	Accuracy
mosquito *is a(n)* insect	0.999
crow *is a(n)* bird	0.999
cactus *is a(n)* plant	0.999
drill *is a(n)* tool	0.999
legend *is a(n)* story	0.998
tutu *is a(n)* skirt	0.998
padlock *is a(n)* lock	0.998
marlin *is a(n)* fish	0.998
teaspoon *is a(n)* spoon	0.998
herbivore *is a(n)* animal	0.998

Table 6.12: 10 most probably hypernym pairs from combining BNC, CYC, LDOCE, Wiktionary Hypernyms and WordNet.

Hypernym Pair	Accuracy
drill *is a(n)* tool	1.000
crow *is a(n)* bird	1.000
cactus *is a(n)* plant	1.000
mosquito *is a(n)* insect	0.999
stork *is a(n)* bird	0.999
legend *is a(n)* story	0.999
sign language *is a(n)* language	0.999
hammer *is a(n)* tool	0.999
parrot *is a(n)* bird	0.999
ant *is a(n)* insect	0.999

Table 6.13: 10 most probable hypernym pairs from combining BNC, CYC, LDOCE, Wiktionary Hypernyms, WordNet and Waterloo MultiText.

6.3 Evaluation Through Applications

The last method of evaluating the hypernyms imported to Roget's Thesaurus is to test these relationships for some task. The new hypernyms are used to enhance Roget's Thesaurus capacity for determining the semantic similarity between two terms. This is tested on several data sets including Miller and Charles (1991) sets and is also used to improve synonym identification questions, like those in Test Of English as a Foreign Language (TOEFL)(Landauer and Dumais, 1997). Another task that can benefit from the use of hypernyms is that of analogy identification. A currently unpublished method of analogy identification using Roget's Thesaurus is improved using the imported hypernyms.

6.3.1 Semantic Distance and Correlation with Human Annotators

I create a new function for measuring semantic similarity using Roget's Thesaurus. This method builds on the one used by Jarmasz and Szpakowicz (2003b). Initially pairs of words are given a score of between 0 and 16 depending on how closely related these words are, 16 being the closest, and 0 being the most distant. I take this score and then use the hypernym hierarchy to modify the scores. If two words are direct hypernyms/hyponyms of each-other their score is increased by 4. If the two words are 2 hypernym/hyponym links apart the score is increased by 3. If the two words are 3 hypernym/hyponym links apart the score is increased by 2. If the two words are 4 hypernym/hyponym links apart the score is increased by 1. If a word is compared to itself the similarity score is also increased by 4. In addition to this, for each of the two words, if that word has no hypernyms or hyponyms the similarity function decreases the score by 1. This is done because sometimes the relationship between a word and the other words in its head, paragraph and even semicolon group are not clear. If no hypernym for that term exists in its head then it becomes more likely that the paragraph that contains the word does not really represent a true sense of the word. For example: "griffin" appears in a paragraph in Head 547 that contains words related to coat of arms. This is not an actual sense of the word griffin, but rather a common use of its image. Not surprisingly "griffin" does not have any hypernyms/hyponyms in this paragraph. This gives a possible range of scores from between -2 and 20. So that there are no negative scores 2 is added to each score giving it a range from 0 to 22.

Three data sets are used for comparing the new and old semantic similarity measures.

Those data sets are Miller and Charles (1991), Rubenstein and Goodenough (1965) and Finkelstein et al. (2001). The Pearson product-moment correlation coefficient is measured between the numbers given by human judges and those achieved by the two systems. The results can be seen in Table 6.15 where the original and enhanced systems are compared. Only nouns were considered for this task.

All three data sets show an improvement. One of the reasons for such an improvement over the old semantic distance function from Jarmasz and Szpakowicz (2003b) is that a disproportionally large number of pairs (almost half of the pairs, 14 out of 30, in the Miller and Charles (1991) data set) were assigned similarity scores of 14 or 16. This new method of finding semantic similarity helps to distinguish between closely related words, specifically those who's distance using the old function were 14 or 16. The old and new semantic distance functions results for Miller and Charles (1991) can be seen in Table 6.14.

I test the results of all three data sets for statistical significance. To do this I divide each data set up into 5 subsets, by placing every fifth sample into the same subset. Once this is done the Pearson product-moment correlation coefficient is calculated for each subset, for both the original system from Jarmasz and Szpakowicz (2003b) and the enhanced one I propose. I then performed a paired t-test on the data, using Microsoft Excel. I found that Rubenstein and Goodenough (1965) and Finkelstein et al. (2001) had P-values of 0.028 and 0.042 respectively. Since both p-values are less than 0.05 these results can be considered statistically significant. I did not find the improvement on Miller and Charles (1991) to be statistically significant, however this is most likely due to the small size of the data set.

6.3.2 Synonym Identification Problems

The same semantic similarity function can also be put towards the problem of identifying a correct synonym of a word from group of candidates. A method similar to that of Jarmasz and Szpakowicz (2003b) is used. This method works by taking a word and four candidates for synonymy and picking the candidate that appears to be most similar using the semantic distance function. If two candidates are found to be equally close to the word with the higher frequency of shortest paths in Roget's Thesaurus is picked. Frequency is based on how many paths there are within a given range of lengths. These different ranges can be called categories, and are shown in Table 6.16. This means, if both candidates have a shortest path of score 14, but one candidate had 6 shortest paths

Word Pair	Human Score	Old System	New System
car,automobile	3.92	16	22
gem,jewel	3.84	16	21
journey,voyage	3.84	16	22
boy,lad	3.76	16	22
coast,shore	3.7	16	22
asylum,madhouse	3.61	14	15
magician,wizard	3.5	14	20
midday,noon	3.42	16	16
furnace,stove	3.11	14	15
food,fruit	3.08	12	14
bird,cock	3.05	12	14
bird,crane	2.97	14	19
tool,implement	2.95	16	22
brother,monk	2.82	14	20
lad,brother	1.66	14	16
crane,implement	1.68	0	2
journey,car	1.16	12	14
monk,oracle	1.1	12	14
cemetery,woodland	0.95	6	8
food,rooster	0.89	12	13
coast,hill	0.87	4	6
forest,graveyard	0.84	6	8
shore,woodland	0.63	6	7
monk,slave	0.55	6	7
coast,forest	0.42	16	16
lad,wizard	0.42	4	4
chord,smile	0.13	0	1
glass,magician	0.11	4	5
rooster,voyage	0.08	2	3
noon,string	0.08	6	6
Correlation		0.773	0.836

Table 6.14: The old and new semantic distance functions on Miller and Charles (1991).

Data Set	Original System	Enhanced System
Miller and Charles (1991)	0.773	0.836
Rubenstein and Goodenough (1965)	0.781	0.838
Finkelstein et al. (2001)	0.411	0.435

Table 6.15: Pearson product-moment correlation coefficient for the original and improved semantic distance functions.

Category Number	Lengths	Granularity in Roget's
1	16, 14, 12	Semicolon Group, Paragraph, POS
2	10, 8	Head, Head Group
3	4, 6	Section, Subsection
4	2	Class
5	0	Thesaurus

Table 6.16: Categories used for calculating frequency of different path lengths.

in Category 1, and another had 3 shortest paths in Category 1 then the one with 6 paths gets picked. Sometimes candidates are phrases, not words. In those cases if the phrase is not found then each word in the phrase is taken separately and the shortest distance to any of the words from the phrase is taken. Exceptions are made for the words "to", "and", and "be", where the semantic distance is not calculated.

Three data sets are used for this application. These data sets are: Test Of English as a Foreign Language (TOEFL)(Landauer and Dumais, 1997), English as a Second Language (ESL) (Turney, 2001) and Reader's Digest Word Power Game (RDWP) (Lewis, 2001). The results for the original and improved systems are show in Table 6.17. The numbers of correct and incorrect answers are shown. Tied results are considered incorrect; as such ties are counted twice, once as a tie, and once as incorrect. Often words were not found in the Thesaurus. The number of question words not found, correct answer words not found and incorrect answer words (other words) not found are also counted.

The results from this experiment show small improvements for ESL and RDWP. No improvement is found for TOEFL, however it did not do any worse. ESL improves from 76% to 82%. TOEFL was unchanged at 72.5%. RDWP improves from 67% to 68.3%.

	ELS		TOEFL		RDWP	
	Original	Enhanced	Original	Enhanced	Original	Enhanced
Correct	38	42	58	58	201	205
Incorrect	12	8	22	22	99	95
Ties	3	0	5	5	23	13
Questions Not Found	0	0	4	4	21	21
Answers Not Found	1	1	5	5	5	5
Other Not Found	1	1	17	17	12	12

Table 6.17: Results for choosing the correct synonym from a set of candidates

6.3.3 Analogy Identification Problems

Analogy Identification problems are those where a pair of words (A and B) is given, and from several other pairs of words one must pick the pair (C and D) that has the same relationship between them as A and B. This can be written A:B::C:D, which is read as: A is to B as C is to D (Turney et al., 2003). This is a difficult problem, university bound high school seniors have an average score of only 57%. One fairly successful system found in Turney (2006a) had an accuracy of 56%. My proposed system does not achieve this level of success, but rather demonstrates how Roget's Thesaurus can be used to help solve analogy problems, and how additional hypernyms can be used to improve these results.

For this task 374 SAT analogy questions from Turney et al. (2003) were used. A pair of words is given and one pair from a set of another 5 pairs of words one pair is selected as the analogy. A simple formula that uses semantic distance is computed to pick the correct pair for the analogy (Nastase, private communication). Each word in the data set had previously been labeled with its part of speech. This information was used when determining semantic distances. The words in the original pair are A and B and the pair that make up the potential analogy pair are C and D. The distance formula is as follows:

$$dist = |semDist(A, B) - semDist(C, D)| + 1/(semDist(A, C) + semDist(B, D) + 1)$$

SemDist is the results of the semantic distance function. The pair that gets the lowest distance score is chosen as the correct analogy. Either the original or the enhanced semantic distance function can be used with this function. It is also possible to modify the function by checking for hypernym analogies. If both the original word pair and one

of the possible analogy candidates have a hypernym relationship between its terms then that can be used to help identify the correct pair. To do this, "hypernym matching", I take into account the difference in the number of hypernym links between two terms in the original pair and the potential analogy pair. *HypernymDist* is simply a count of how many hypernym links there are between two words. The distance calculated above is altered by the following formula:

$$dist = dist - (k - |hypernymDist(A, B) - hypernymDist(C, D)|)$$

k is a constant. Ideally k should be a number that is suitably high so that pairs of words that are both related by hypernymy are favored more than pairs that are not both related by hypernymy. I found that $k = 8$ gives good results largely through trial and error. Four different variations on this algorithm are tested. The variations are: the original distance function without hypernym matching, the original distance function with hypernym matching, the enhanced distance function without hypernym matching, and the enhance distance function with hypernym matching. The results are shown in Table 6.18. Omitted are problems that could not be solved because one or more of the words in the original analogy were not present in the Thesaurus and Ties are not counted as Incorrect, as they were in Table 6.17. The original semantic distance function with hypernym matching had the best results. Using the enhanced semantic distance function with hypernym matching gave the same number of correct answers, but it had more incorrect answers as well. The accuracy of the system is improved from 33.1% to 34.8%. All four systems are significantly above the baseline of 20% that would be achieved by random guessing.

The best system would appear to be where hypernym matching and the original semantic distance function is used. There are only 24 cases where the original word pair and at least one analogy candidate were both related by hypernymy. Table 6.19 shows the results on only those 24 word pairs for all four systems. All problems where the original two terms are not hypernyms are omitted for this experiment. The results from this experiment are interesting. It can be clearly seen that both using the enhanced semantic distance function, or using the original distance function, with hypernym matching perform quite similarly, while using the enhanced semantic distance function in conjunction with hypernym matching yields only a small improvement. All three of these options show considerable improvement over simply using the original semantic distance function without any sort of hypernym matching.

System	Correct	Incorrect	Ties	Omitted
Original	124	226	14	10
Original with Hypernyms	130	220	14	10
Enhanced	129	231	4	10
Enhnaced with Hypernyms	130	230	4	10

Table 6.18: Results for choosing the correct analogy from a set of candidates.

System	Correct	Incorrect	Ties
Original	7	15	2
Original with Hypernyms	13	9	2
Enhanced	13	10	1
Enhnaced with Hypernyms	14	9	1

Table 6.19: Results for choosing the correct analogy from a set of candidates, where the original pair of words are related by hypernymy.

6.4 Conclusion

Several interesting points can be taken from this chapter. First of all, it is quite difficult for people to agree on what is and is not a hypernym, as shown by the tests on human evaluators. There are several reasons why people may not agree. It may be that they are not aware of some more obscure word senses for which two terms are hypernyms. Having proper definitions for the words, or having the word used in context may help evaluators to agree as to whether two words are hypernyms or not.

Dispite poor human agreement, I have shown that these hypernyms that were imported into Roget's Thesaurus can be quite useful for some applications, particularly for determining semantic relatedness between two words. This can be seen in Table 6.15 where every data set performed better with my new semantic distance function that takes advantage of the hypernyms. The hypernyms were also useful for solving analogy questions. Using hypernyms 6 more questions were answered correctly. This may not seem like a large number, however there are only 24 questions where both the original word pair and one of the potential analogy candidates had hypernym relationships. With WordNet there are several different functions for semantic relatedness. It is quite possible that other functions for semantic relatedness could be devised for Roget's Thesaurus that

would take advantage of the new hypernym relationships. For the TOEFL style synonym questions, although the new semantic distance function is only a small improvement for 2 of the 3 data sets, though at no point are the results worse.

Chapter 7

Conclusion

7.1 Evaluation Results

Two main points can be taken from this thesis. One is that hypernym classification is difficult for people as well as machines. The second is that adding hypernym links, even those that humans do not always agree on, can useful for helping lexical resources like Roget's Thesaurus to solve some NLP applications.

7.1.1 Human Evaluations

The results from my experiments are mixed. My variations on the Machine Learning method based on Snow et al. (2005) did not work very well. It should be noted that the Snow et al. (2005) implementation had a f-score of 30% to 35%, while my implementation had a maximum f-score of 63% using 10-fold cross validation. Human evaluators on average found the system to be about 25% accurate (precision). Although it should be noted that 58% of the sample hypernyms from this system were labeled correct by at least 1 evaluator seen in Table 6.8. It is impossible for human to effectively measure recall and so it is difficult to conclude exactly what F-measure score the system would actually get. The frequency with which different pairs of words, and dependency paths appear in the training corpus will depend on the size of the training corpus. To actually use this system on unseen data, a comparable amount of data may be needed to ensure that correct and incorrect hypernym pairs can be identified by the dependency paths found. It is possible that finding a few new dependency paths for a new pair of terms could change their classification from positive to negative, or vice versa. There are many differences between my implementation of the system and that of Snow et al. (2005),

as described in Chapter 5. This makes any direct comparison of my results with their results difficult as well.

My attempts to infer new hypernyms using synonyms from Wiktionary and Roget's Thesaurus Semicolon Groups were not very successful. It would appear that while most terms in a semicolon group do share a common hypernym, it is not always clear what hypernym that is. Since multiple hypernyms are possible, and allowed for in my work a method of selecting the most appropriate hypernym may be useful. These synonym-based methods are also likely to increase error. If an incorrect hypernym is used to infer new hypernyms the new hypernyms will almost certainly be incorrect too.

Mining relationships from text in techniques similar to Hearst (1992) can be done with mixed accuracy as was seen by using the patterns on the BNC and the Waterloo MultiText System where accuracies of 66% and 54% respectively were found. Mining hypernym relationships from dictionaries like LDOCE and Wiktionary proved to be fairly successful, and was comparable in accuracy to mining relationship from other lexical resources and ontologies like WordNet and Cyc.

It was difficult to get good agreement between raters. With kappa scores ranging between .11 and .44 the rater agreement was not very strong at all. WordNet itself did not get an extremely good rating. Much of this may be because more obscure senses of words may exist in WordNet and so some words may be mislabeled because a rater may not think of the correct sense of the word. Also some hypernym pairs in WordNet are very close to being synonyms and so may be labeled as non-hypernyms. All this shows that hypernym identification is difficult for people as well as for machines.

Ultimately hypernyms were taken from 6 resources. WordNet, Open Cyc, LDOCE, Wiktionary, the BNC and the Waterloo MultiText System. The last two were done using Hearst (1992)'s patterns and LDOCE and Wiktionary were mined using patterns similar to those used by Nakamura and Nagao (1988). When the accuracy for each of the hypernyms mined from each resource (as determined by averaging the results from the human annotators) is averaged, the hypernyms imported to Roget's Thesaurus are about 73% accurate. The estimated accuracy of the hypernyms mined ranges from nearly 100% to as low as 54%.

7.1.2 Evaluation through applications

Despite some mixed results from human annotators, importing these hypernym relationships into Roget's Thesaurus has proven to be quite beneficial. The new semantic

similarity function that I created, which takes advantage of the new hypernyms imported to the thesaurus, has shown itself to improve Roget's Thesaurus' ability to determine semantic similarity. On the Miller and Charles (1991) and Rubenstein and Goodenough (1965) data sets the correlation with human annotators improved by 0.063 and 0.057. These improvements are fairly substantial given that the scores using the un-enhanced Roget's Thesaurus were quite good to start with. A smaller but still significant improvement can be seen on the Finkelstein et al. (2001) data set.

For identifying the correct synonym three data sets were tried: TOEFL (Landauer and Dumais, 1997), ESL (Turney, 2001) and RDWP (Lewis, 2001). Small improvements were measured for ESL and RDWP, while TOEFL showed no change between the old and new semantic similarity scores. The improvement seen for these three data sets was not as great as on the Miller and Charles (1991) and Rubenstein and Goodenough (1965) however the new semantic distance function will only help in cases where two words are both good candidates for synonymy and it is hard to tell which one is better.

Another application that showed some improvement using hypernyms was that of solving SAT Analogy questions (Turney et al., 2003). Using the improved semantic distance function did improve the results for answering SAT questions, however the best improvements came from hypernym matching, where analogies that are related by hypernymy are identified. Given that there were relatively few cases where the original word pair, and one of the possible analogies were both hypernyms a noticeable improvement can be seen. This system would likely be much more effective if more hypernym relationships as well as other relationships could be found. My best system had 34.8% accuracy. The problem of solving analogy questions is not an easy one for people or machines. The most successful system that I am aware of is 56% accurate on the same SAT data set (Turney, 2006a), while the average college bound high school student gets about 57% accuracy.

The semantic similarity function proposed in this thesis has shown to improve the results on 6 out of 7 data sets it was tested on. It performed equally well on the other data set. Like with WordNet it is possible that many different semantic similarity functions could be designed to work with the enhanced Roget's Thesaurus. The function I developed is not necessarily the best function, but merely shows the potential that Roget's Thesaurus has when new relationships are integrated into the Thesaurus.

7.2 Human Confirmation of the Hypernym Structure

As was stated before, the hypernyms added to this resource are the first step in a greater project to make building lexical resources easier for humans. By providing a human annotator with a set of potential hypernyms and a probability assigned to each hypernym a great deal of manual effort can be removed from the process. The job of the human would now be to simply assign correct/incorrect values to each hypernym pair, rather than build an entire hypernym hierarchy from scratch.

To make this process go smoothly for Roget's Thesaurus a system will need to be developed which displays all necessary information for assigning correct/incorrect labels to the hypernyms. This information will include the hypernym pair, an estimate of accuracy and the context in which it occurs. For Roget's Thesaurus the context will be the paragraph or semicolon group that the hypernym pair appears in.

7.3 Other resources for Hypernym Extraction

Two resources that are discussed in the literature review, but are not mined for relationships are FACTOTUM and UMLS. The primary reason for not using UMLS is that it focuses on medical terminology, which is less likely to appear in a general use thesaurus like Roget's Thesaurus.

FACTOTUM contains many relationships, though it sometimes combines relationships in ways that are not always clear. None the less, if this resource contains relationships not already found it may be worth considering in the future.

OpenCyc proved to be an accurate source of hypernyms although its relatively small size limits its usefulness. A new resource called ResearchCyc has been released which is similar to OpenCyc, but is much larger in size. ResearchCyc may be worth examining as a resource for hypernym data.

More hypernyms can also be mined from other corpora using methods similar to those in this thesis. Other dictionaries could also be used. To find relationships for words pertaining to a specific subject a corpus focusing on that subject could be used.

7.4 New Kinds of Relationships and New Words

Relationships other than hypernyms/hyponyms can also be mined from the resource. This could include such relationships as meronymy/holonymy, antonymy and synonymy. Roget's Thesaurus contains Nouns, Verbs, Adjectives and Adverbs all in the same Head. It could be quite useful to identify what relationships exist between words from different parts of speech within the same head. Identifying the actions and attributes of particular objects could lead to valuable relationships.

Relationships for different parts of speech can also be included. For verbs, hypernymy as well as troponymy and entailment could be included. Causal relationships between verbs could also be useful. There are fewer obvious relationships to go between adjectives and adverbs, however their relationships to nouns and verbs can be included.

It may also be useful to find methods of incorporating new words into Roget's Thesaurus. Since the lexicon used is from 1987 many new inventions in technology and significant cultural events will not be included. An automatic method of updating the thesaurus would be quite useful in keeping it relevant. This would be useful for other resources too, particularly the 1911 version of Roget's Thesaurus since, unlike the 1987 version, it is freely available.

7.5 Further Evaluation

One useful evaluation would be to compare the enhanced Roget's Thesaurus against other established resources like WordNet. Also comparing the enhanced version of Roget's Thesaurus against other versions of the same resource will be useful in demonstrating which tasks this resource is best suited to. The 1911 version of Roget's Thesaurus is freely available and can be compared against both this enhanced version of the 1987 Roget's Thesaurus, and the original 1987 Roget's Thesaurus. As I do in Chapter 6.3 applications can be selected for which each of these resources can be used. If the two resources are interchangeable but all other parameters remain the same they can be compared.

There are several applications that can be used to evaluate the enhanced Roget's Thesaurus. Semantic similarity scores have been used for word sense disambiguation problems (Patwardhan et al., 2003). Another application is determining similarity between texts (Corley and Mihalcea, 2005). Document clustering is also an application that could be tested. In Hotho et al. (2003) several methods determining similarity between documents in the Reuters corpus using WordNet are tested. All of these applications

could be used lexical resources.

There is still much work that can be done to improve the usefulness of Roget's Thesaurus. Although automatic methods of enhancing existing lexical resources will never be as accurate as having humans manually do it, I have shown that it can be done, and that it is useful to do so.

Bibliography

ACE (2004). The automatic content extraction (ACE) projects. http://www.ldc.upenn.edu/Projects/ACE/.

Achour, S., Dojat, M., Brethon, J.-M., Blain, G., and Lepage, E. (1999). The use of the UMLS knowledge sources for the design of a domain specific ontology: A practical experience in blood transfusion. In *AIMDM '99: Proceedings of the Joint European Conference on Artificial Intelligence in Medicine and Medical Decision Making*, pages 249–253, London, UK. Springer-Verlag.

Alshawi, H. (1987). Processing dictionary definitions with phrasal pattern hierarchies. *Computational Linguistics*, 13(3-4):195–202.

Berland, M. and Charniak, E. (1999). Finding parts in very large corpora. In *Proceedings of the 37th Annual Meeting of the Association for Computational Linguistics on Computational Linguistics*, pages 57–64, Morristown, NJ, USA. Association for Computational Linguistics.

Brill, E. (1994). Some advances in transformation-based part of speech tagging. In *AAAI '94: Proceedings of the Twelfth National Conference on Artificial intelligence (vol. 1)*, pages 722–727, Menlo Park, CA, USA. American Association for Artificial Intelligence.

Budanitsky, A. and Hirst, G. (2006). Evaluating wordnet-based measures of lexical semantic relatedness. *Computational Linguistics*, 32(1):13–47.

Bunescu, R. and Pasça, M. (2006). Using encyclopedia knowledge for name entity disambiguation. In *Proceedings of EACL-2006*, pages 9–16, Trento, Italy.

Burnard, L. (2000). Reference guide for the british national corpus (world edition).

Caraballo, S. A. (1999). Automatic construction of a hypernym-labeled noun hierarchy from text. In *Proceedings of the 37th Annual Meeting of the Association for Computational Linguistics on Computational Linguistics*, pages 120–126.

Caraballo, S. A. and Charniak, E. (1999). Determining the specificity of nouns from text. In *Proceedings the joint SIGDAT Conference on Empirical Methods in Natural Language Processing (EMNLP) and Very Large Corpora (VLC)*, pages 63–70.

Cassidy, P. J. (2000). An investigation of the semantic relations in the roget's thesaurus: Preliminary results. In *Proceedings of CICLing-2000, International Conference on Intelligent Text Processing and Computational Linguistics*, pages 181–204.

Cederberg, S. and Widdows, D. (2003). Using LSA and noun coordination information to improve the precision and recall of automatic hyponymy extraction. In *Proceedings of the Seventh Conference on Natural Language Learning at HLT-NAACL 2003*, pages 111–118.

Chen, J., Ji, D., Tan, C. L., and Niu, Z. (2006). Relation extraction using label propagation based semi-supervised learning. In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics*, pages 129–136, Sydney, Australia. Association for Computational Linguistics.

Clarke, C. L. A. and Terra, E. L. (2003). Passage retrieval vs. document retrieval for factoid question answering. In *SIGIR '03: Proceedings of the 26th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 427–428, New York, NY, USA. ACM Press.

Cohen, J. (1960). A coefficient of agreement for nominal scales. *Educational and Psychological Measurement*, 20(1):37–46.

Corley, C. and Mihalcea, R. (2005). Measures of text semantic similarity. In *Proceedings of the ACL Workshop on Empirical Modeling of Semantic Equivalence and Entailment*, pages 13–18.

Culotta, A., McCallum, A., and Betz, J. (2006). Integrating probabilistic extraction models and data mining to discover relations and patterns in text. In *Proceedings of the Human Language Technology Conference of the NAACL, Main Conference*, pages 296–303, New York City, USA. Association for Computational Linguistics.

Deerwester, S., Dumais, S. T., Furnas, G. W., Landauer, T. K., and Harshman, R. (1990). Indexing by latent semantic analysis. *American Society for Information Science*, 41(6):391–407.

Devore, J. L. (1999). *Probability and Statistics for Engineering and the Sciences (5th ed.)*. Brooks/Cole Publishing Company, Pacific Grove, CA.

Dunning, T. (1993). Accurate methods for the statistics of surprise and coincidence. *Computational Linguistics*, 19(1):61–74.

Fellbaum, C. (1998a). A semantic network of english verbs. In Fellbaum, C., editor, *WordNet: An Electronic Lexical Database*, pages 69–104. MIT Press, Cambridge, MA.

Fellbaum, C., editor (1998b). *WordNet – An Electronic Lexical Database*. MIT Press, Cambridge, Massachusetts and London, England.

Finkelstein, L., Gabrilovich, E., Matias, Y., Rivlin, E., Solan, Z., Wolfman, G., and Ruppin, E. (2001). Placing search in context: the concept revisited. In *WWW '01: Proceedings of the 10th International Conference on World Wide Web*, pages 406–414, New York, NY, USA. ACM Press.

Fleiss, J. L. (1981). *Statistical Methods for Rates and Proportions (2nd edn)*. John Wiley & Sons, New York.

Geleijnse, G. and Korst, J. (2006). Learning effective surface text patterns for information extraction. In *Proceedings of the EACL-2006 Workshop on Adaptive Text Extraction and Mining (ATEM 2006)*, pages 1–8, Trento, Italy.

Girju, R., Badulescu, A., and Moldovan, D. (2003). Learning semantic constraints for the automatic discovery of part-whole relations. In *NAACL '03: Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology*, pages 1–8, Morristown, NJ, USA. Association for Computational Linguistics.

Girju, R., Badulescu, A., and Moldovan, D. (2006). Automatic discovery of part-whole relations. *Computational Linguistics*, 32(1):83–136.

Grolier, editor (1990). *Academic American Encyclopedia*. Grolier Electronic Publishing, Danbury, Connecticut.

Guthrie, L., Slator, B. M., Wilks, Y., and Bruce, R. (1990). Is there content in empty heads? In *Proceedings of the 13th Conference on Computational Linguistics*, pages 138–143, Morristown, NJ, USA. Association for Computational Linguistics.

Hearst, M. A. (1992). Automatic acquisition of hyponyms from large text corpora. In *Proceedings of the 14th Conference on Computational Linguistics*, pages 539–545.

Hirst, G. and St-Onge, D. (1998). Lexical chains as representation of context for the detection and correction malapropisms. In Fellbaum, C., editor, *WordNet: An Electronic Lexical Database*, pages 305–322. MIT Press, Cambridge, MA.

Hotho, A., Staab, S., and Stumme, G. (2003). Wordnet improves text document clustering. In *Proceedings of the Semantic Web Workshop at SIGIR-2003, 26th Annual International ACM SIGIT Conference*.

Humphreys, B. L. and Lindberg, D. B. A. (1993). The UMLS project: Making the conceptual connection between users and the information they need. *Bulletin of the Medical Library Association*, 81(2):170–177.

Ide, N. and Véronis, J. (1993). Refining taxonomies extracted from machine-readable dictionaries. *Research in Humanities Computing II*, pages 145–159.

Jarmasz, M. (2003). Roget's thesaurus as a lexical resource for natural language processing. Master's thesis, University of Ottawa.

Jarmasz, M. and Szpakowicz, S. (2001a). The design and implementation of an electronic lexical knowledge base. In *Proceedings of the 14th Biennial Conference of the Canadian Society for Computational Studies of Intelligence (AI 2001)*, pages 325–334.

Jarmasz, M. and Szpakowicz, S. (2001b). Roget's thesaurus: a lexical resource to treasure. In *Proceedings of the NAACL WordNet and Other Lexical Resources Workshop*, page 186 188.

Jarmasz, M. and Szpakowicz, S. (2003a). Not as easy as it seems: Automating the construction of lexical chains using roget's thesaurus. In *Proceedings of the 16th Canadian Conference on Artificial Intelligence (AI 2003)*, pages 544–549.

Jarmasz, M. and Szpakowicz, S. (2003b). Roget's thesaurus and semantic similarity. In *Proceedings of Conference on Recent Advances in Natural Language Processing (RANLP 2003)*, pages 212–219.

Jiang, J. J. and Conrath, D. W. (1997). Semantic similarity based on corpus statistics and lexical taxonomy. In *Proceedings of 10th International Conference on Research on Computational Linguistics (ROCLING X)*, pages 19–33.

Joachims, T. (1999). Making large-scale support vector machine learning practical. In B. Schölkopf, C. Burges, A. S., editor, *Advances in Kernel Methods: Support Vector Learning*, pages 169–184. MIT Press, Cambridge, MA, USA.

Kirkpatrick, B., editor (1987). *Roget's Thesaurus of English Words and Phrases*. Penguin, Harmondsworth, Middlesex, England.

Kwong, O. Y. (1998a). Aligning wordnet with additional lexical resources. In *Proceedings of the COLING/ACL Workshop on Usage of WordNet in Natural Language Processing Systems*, pages 73–79.

Kwong, O. Y. (1998b). Bridging the gap between dictionary and thesaurus. In *Proceedings of the 36th Annual Meeting on Association for Computational Linguistics*, pages 1487–1489, Morristown, NJ, USA. Association for Computational Linguistics.

Kwong, O. Y. (2001). Word sense disambiguation with an integrated lexical resource. In *Proceedings of the NAACL WordNet and Other Lexical Resources workshop*, pages 11–16.

Landauer, T. and Dumais, S. (1997). A solution to Plato's problem: The latent semantic analysis theory of acquisition, induction, and representation of knowledge. *Psychological Review*, 104:211–240.

Landis, R. J. and Koch, G. G. (1977). The Measurement of Observer Agreement for Categorical Data. *Biometrics*, 33:159–174.

Leacock, C. and Chodorow, M. (1998). Combining local context and wordnet sense similarity for word sense disambiguation. In Fellbaum, C., editor, *WordNet: An Electronic Lexical Database*, pages 265–284. MIT Press, Cambridge, MA.

Lenat, D. B. (1995). Cyc: A large-scale investment in knowledge infrastructure. *Communications of the ACM*, 38(11).

Lewis, M., editor (2000-2001). *Readers Digest, 158(932, 934, 935, 936, 937, 938, 939, 940), 159(944, 948).* Readers Digest Magazines Canada Limited.

Lin, D. (1998a). Dependency-based evaluation of MINIPAR. In *Proceedings of the Workshop on the Evaluation of Parsing Systems, First International Conference on Language Resources and Evaluation*.

Lin, D. (1998b). An information-theoretic definition of similarity. In *ICML '98: Proceedings of the Fifteenth International Conference on Machine Learning*, pages 296–304, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

Lin, D. and Pantel, P. (2001). Discovery of inference rules for question-answering. *Natural Language Engineering*, 7(4):343–360.

Matuszek, C., Cabral, J., Witbrock, M., and DeOliveira, J. (2006). An introduction to the syntax and content of cyc. In *Proceedings of the 2006 AAAI Spring Symposium on Formalizing and Compiling Background Knowledge and Its Applications to Knowledge Representation and Question Answering*.

McLernon, B. and Kushmerich, N. (2006). Transductive pattern learning for information extraction. In *Proceedings of the EACL-2006 Workshop on Adaptive Text Extraction and Mining (ATEM 2006)*, pages 35–41, Trento, Italy.

Miller, G. (1998a). Nouns in wordnet. In Fellbaum, C., editor, *WordNet: An Electronic Lexical Database*, pages 23–46. MIT Press, Cambridge, MA.

Miller, G. A. and Charles, W. G. (1991). Contextual correlates of semantic similarity. *Language and Cognitive Process*, 6(1):1–28.

Miller, G. A., Leacock, C., Tengi, R., and Bunker, R. T. (1993). A semantic concordance. In *HLT '93: Proceedings of the workshop on Human Language Technology*, pages 303–308, Morristown, NJ, USA. Association for Computational Linguistics.

Miller, K. (1998b). Modifiers in wordnet. In Fellbaum, C., editor, *WordNet: An Electronic Lexical Database*, pages 47–67. MIT Press, Cambridge, MA.

Morin, E. and Jacquemin, C. (1999). Projecting corpus-based semantic links on a thesaurus. In *Proceedings of the 37th Annual Meeting of the Association for Computational Linguistics on Computational Linguistics*, pages 389–396.

Morris, J. and Hirst, G. (1991). Lexical cohesion computed by thesaural relations as an indicator of the structure of text. *Computational Linguistics*, 17(1):21–48.

Nakamura, J.-i. and Nagao, M. (1988). Extraction of semantic information from an ordinary english dictionary and its evaluation. In *Proceedings of the 12th Conference on Computational linguistics*, pages 459–464, Morristown, NJ, USA. Association for Computational Linguistics.

Nastase, V. and Szpakowicz, S. (2001). Word sense disambiguation in roget's thesaurus using wordnet. In *Proceedings of the NAACL WordNet and Other Lexical Resources workshop*, pages 12–22.

O'Hara, T. P. and Wiebe, J. (2003). Classifying functional relations in factotum via wordnet hypernym associations. In *Fourth International Conference on Intelligent Text Processing and Computational Linguistics (CICLing-2003)*, pages 347–359.

Pantel, P. and Pennacchiotti, M. (2006). Espresso: Leveraging generic patterns for automatically harvesting semantic relations. In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics*, pages 113–120, Sydney, Australia. Association for Computational Linguistics.

Pantel, P. and Ravichandran, D. (2004). Automatically labeling semantic classes. In *Proceedings of the 2004 Human Language Technology Conference (HLT-NAACL-04)*, pages 321–328.

Patwardhan, S., Banerjee, S., and Pedersen, T. (2003). Using measures of semantic relatedness for word sense disambiguation. In *Proceedings of the Fourth International Conference on Intelligent Text Processing and Computational Linguistics*, pages 241–257.

Pennacchiotti, M. and Pantel, P. (2006). Ontologizing semantic relations. In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics*, pages 793–800, Sydney, Australia. Association for Computational Linguistics.

Procter, P. (1978). *Longman Dictionary of Contemporary English*. Longman Group Ltd.

Quinlan, J. R. (1993). *C4.5: Programs for Machine Learning*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA.

Resnik, P. (1995). Using information content to evaluate semantic similarity. In *Proceedings of the 14th International Joint Conference on Artificial Intelligence*, pages 448–453.

Rubenstein, H. and Goodenough, J. B. (1965). Contextual correlates of synonymy. *Communication of the ACM*, 8(10):627–633.

Ruiz-Casado, M., Alfonseca, E., and Castells, P. (2005). Using context-window overlapping in synonym discovery and ontology extension. In *Proceedings of the International Conference Recent Advances in Natural Language Processing, RANLP-2005*.

Rydin, S. (2002). Building a hyponymy lexicon with hierarchical structure. In *Proceedings of the SIGLEX Workshop on Unsupervised Lexical Acquisition, ACL'02*, pages 26–33.

Senellart, P. and Blondel, V. D. (2003). Automatic discovery of similar words. In Michael W. Berry, editor, *Survey of Text Mining*. Springer-Verlag.

Shinzato, K. and Torisawa, K. (2004). Acquiring hyponymy relations from web documents. In *Proceedings of the 2004 Human Language Technology Conference (HLT-NAACL-04)*, pages 73–80.

Shinzato, K. and Torisawa, K. (2005). A simple WWW-based method for semantic word class acquisition. In *Proceedings of the Recent Advances in Natural Language Processing (RANLP05)*, pages 493–500.

Siniakov, P. (2006). Recognition of synonyms by a lexical graph. In *Proceedings of the EACL-2006 Workshop on Adaptive Text Extraction and Mining (ATEM 2006)*, pages 32–39, Trento, Italy.

Snow, R., Jurafsky, D., and Ng, A. Y. (2005). Learning syntactic patterns for automatic hypernym discovery. In Saul, L. K., Weiss, Y., and Bottou, L., editors, *Advances in Neural Information Processing Systems 17*, pages 1297–1304. MIT Press, Cambridge, MA.

Snow, R., Jurafsky, D., and Ng, A. Y. (2006). Semantic taxonomy induction from heterogenous evidence. In *Proceedings of COLING/ACL 2006*.

Sombatsrisomboon, R., Matsuo, Y., and Ishizuka, M. (2003). Acquisition of hypernyms and hyponyms from the www. In *Procedings of the 2nd International Workshop*

on Active Mining (AM2003) (In Conjunction with the International Symposium on Methodologies for Intelligent Systems), pages 7–13.

Suchanek, F. M., Ifrim, G., and Weikum, G. (2006). LEILA: Learning to extract information by linguistic analysis. In *Proceedings of the 2nd Workshop on Ontology Learning and Population: Bridging the Gap between Text and Knowledge*, pages 18–25, Sydney, Australia. Association for Computational Linguistics.

Surdeanu, M., Turmo, J., and Ageno, A. (2006). A hybrid approach for the acquisition of information extraction patterns. In *Proceedings of the EACL-2006 Workshop on Adaptive Text Extraction and Mining (ATEM 2006)*, pages 48–55, Trento, Italy.

Tanev, H. T. and Magnini, B. (2006). Weakly supervised approaches for ontology population. In *Proceedings of EACL-2006*, pages 17–24, Trento, Italy.

Tomita, J., Soderland, S., and Etzioni, O. (2006). Expanding the recall of relation extraction by bootstrapping. In *Proceedings of the EACL-2006 Workshop on Adaptive Text Extraction and Mining (ATEM 2006)*, pages 56–63, Trento, Italy.

Turney, P. (2001). Mining the web for synonyms: PMI-IR versus LSA on TOEFL. In *Proceedings of the Twelfth European Conference on Machine Learning (ECML-2001)*, pages 491–502.

Turney, P. (2006a). Similarity of semantic relations. *Computational Linguistics*, 32(3):379–416.

Turney, P., Littman, M., Bigham, J., and Shnayder, V. (2003). Combining independent modules to solve multiple-choice synonym and analogy problems. In *Proceedings International Conference on Recent Advances in Natural Language Processing (RANLP-03)*, pages 482–489.

Turney, P. D. (2006b). Expressing implicit semantic relations without supervision. In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics*, pages 313–320, Sydney, Australia. Association for Computational Linguistics.

United States National Library of Medicine: National Institute of Health (2004). About the UMLS resources. http://www.nlm.nih.gov/research/umls/about_umls.html.

Voorhees, E. M. and Harman, D. K., editors (2000). *The Ninth Text Retrieval Conference (TREC-9)*. Department of Commerce, National Institute of Standards and Technology.

Wiktionary (2006). Main page - wiktionary. http://en.wiktionary.org/wiki/Main_Page/.

Witten, I. H. and Frank, E., editors (2005). *Data Mining: Practical Machine Learning Tools and Techniques 2nd Edition*. Morgan Kaufmann, San Francisco.

Zhang, M., Zhang, J., and Su, J. (2006). Exploring syntactic features for relation extraction using a convolution tree kernel. In *Proceedings of the Human Language Technology Conference of the NAACL, Main Conference*, pages 288–295, New York City, USA. Association for Computational Linguistics.